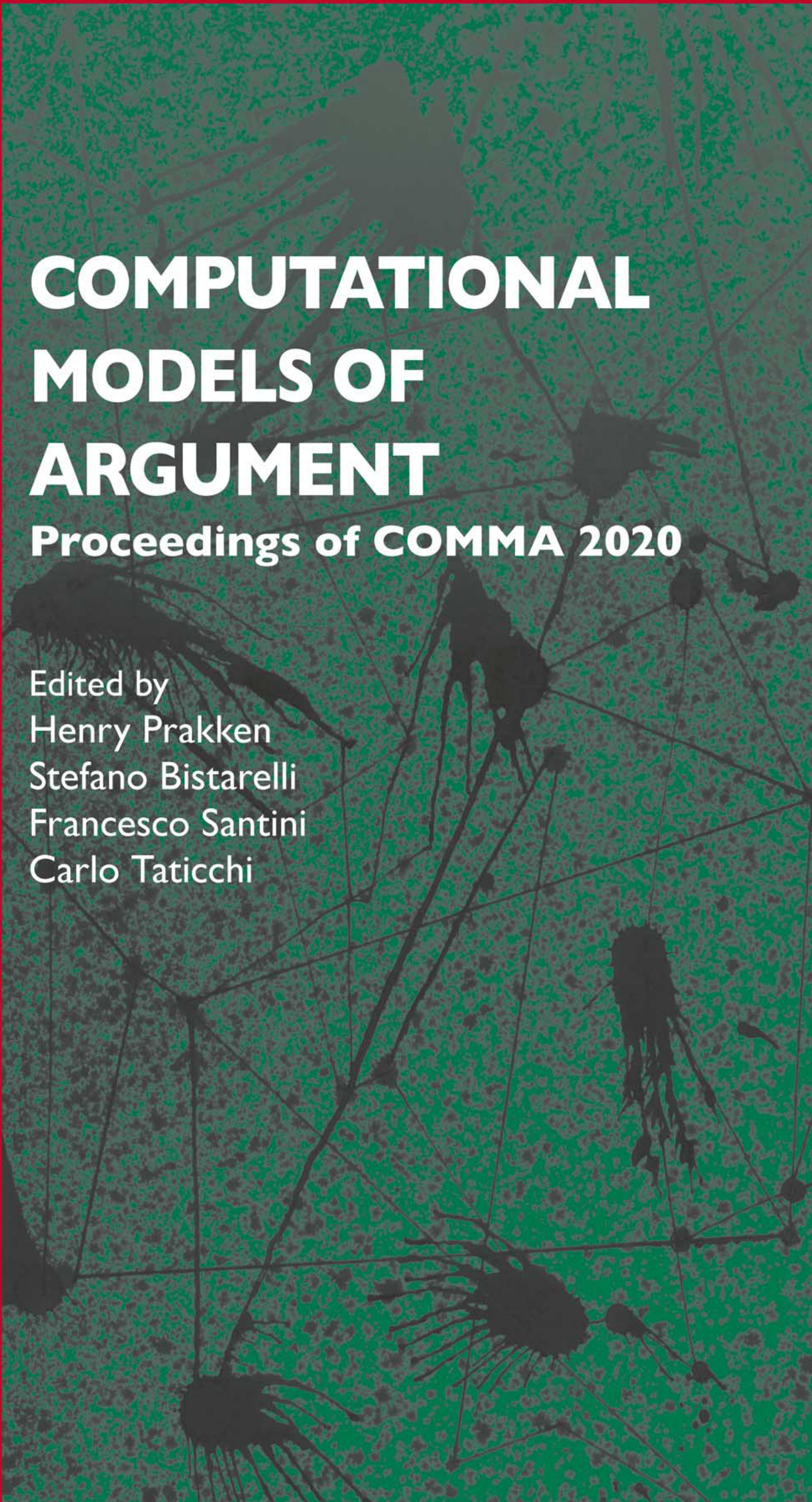
Frontiers
in
Artificial
Intelligence
and
Applications



COMPUTATIONAL MODELS OF ARGUMENT

Proceedings of COMMA 2020

Edited by
Henry Prakken
Stefano Bistarelli
Francesco Santini
Carlo Taticchi

IOS
Press

COMPUTATIONAL MODELS OF ARGUMENT

The investigation of computational models of argument is a rich and fascinating interdisciplinary research field with two ultimate aims: the theoretical goal of understanding argumentation as a cognitive phenomenon by modeling it in computer programs, and the practical goal of supporting the development of computer-based systems able to engage in argumentation-related activities with human users or among themselves.

The biennial International Conferences on Computational Models of Argument (COMMA) provide a dedicated forum for the presentation and discussion of the latest advancements in the field, and cover both basic research and innovative applications. This book presents the proceedings of COMMA 2020. Due to the Covid-19 pandemic, COMMA 2020 was held as an online event on the originally scheduled dates of 8 -11 September 2020, organised by the University of Perugia, Italy. The book includes 28 full papers and 13 short papers selected from a total of 78 submissions, the abstracts of 3 invited talks and 13 demonstration abstracts. The interdisciplinary nature of the field is reflected, and contributions cover both theory and practice. Theoretical contributions include new formal models, the study of formal or computational properties of models, designs for implemented systems and experimental research. Practical papers include applications to medicine, law and criminal investigation, chatbots and online product reviews. The argument-mining trend from previous COMMA's is continued, while an emerging trend this year is the use of argumentation for explainable AI.

The book provided an overview of the latest work on computational models of argument, and will be of interest to all those working in the field.

9 781643 681061

ISBN 978-1-64368-106-1 (print)
ISBN 978-1-64368-107-8 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

COMPUTATIONAL MODELS OF ARGUMENT

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 326

Recently published in this series

Vol. 325. G. De Giacomo, A. Catala, B. Dilkina, M. Milano, S. Barro, A. Bugarín and J. Lang (Eds.), ECAI 2020 – 24th European Conference on Artificial Intelligence – 29 August–8 September 2020, Santiago de Compostela, Spain

Vol. 324. C. Troussas and C. Sgouropoulou, Innovative Trends in Personalized Software Engineering and Information Systems – The Case of Intelligent and Adaptive E-learning Systems

Vol. 323. L.C. Jain, X. Zhao, V.E. Balas and F. Shi (Eds.), Information Technology and Intelligent Transportation Systems

Vol. 322. M. Araszkiewicz and V. Rodríguez-Doncel (Eds.), Legal Knowledge and Information Systems – JURIX 2019: The Thirty-second Annual Conference

Vol. 321. A. Dahanayake, J. Huiskonen, Y. Kiyoki, B. Thalheim, H. Jaakkola and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXXI

Vol. 320. A.J. Tallón-Ballesteros (Ed.), Fuzzy Systems and Data Mining V – Proceedings of FSDM 2019

Vol. 319. J. Sabater-Mir, V. Torra, I. Aguiló and M. González-Hidalgo (Eds.), Artificial Intelligence Research and Development – Proceedings of the 22nd International Conference of the Catalan Association for Artificial Intelligence

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Computational Models of Argument

Proceedings of COMMA 2020

Edited by

Henry Prakken

Department of Information and Computing Sciences, Utrecht University & Faculty of Law, University of Groningen, The Netherlands

Stefano Bistarelli

Department of Mathematics and Computer Science, University of Perugia, Italy

Francesco Santini

Department of Mathematics and Computer Science, University of Perugia, Italy

and

Carlo Taticchi

Department of Computer Science, Gran Sasso Science Institute, L'Aquila, Italy

IOS Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-106-1 (print)
ISBN 978-1-64368-107-8 (online)
Library of Congress Control Number: 2020944781
doi: 10.3233/FAIA326

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

The investigation of computational models of argument is a rich, interdisciplinary, and fascinating research field with two ultimate aims. A theoretical goal is to understand argumentation as a cognitive phenomenon by modelling it in computer programmes, while a practical goal is to support the development of computer-based systems able to engage in argumentation-related activities with human users or among themselves. These ambitious research goals involve the study of natural, artificial, and theoretical argumentation and, as such, requires openness to interactions with a variety of disciplines, such as philosophy, cognitive science, linguistics, communication studies, formal logic, game theory and mathematical graph theory.

The computational study of argumentation has two main historic origins. In 1987 John Pollock published his seminal paper *Defeasible reasoning*, in which he stressed the importance of reasons in the construction of arguments and gave the first systematic formal account of the evaluation of arguments given their internal structure and their relation with counterarguments. And in 1995 Phan Minh Dung's paper *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games* initiated the study of so-called abstract argumentation frameworks, which leave the nature of arguments and their relations unspecified but still allow for a rich theory of argument evaluation. This paper was in 2018 awarded the AI Journal Classic Paper Award, to recognise its role in making argumentation a mainstream research topic in artificial intelligence.

Since 2006 the biennial International Conference on Computational Models of Argument (COMMA) has provided a dedicated forum for presentation and discussion of the latest advancements in this interdisciplinary field, covering both basic research and innovative applications. The first COMMA was supported by the EU 6th Framework Programme project ASPIC and was hosted by the University of Liverpool in 2006. After the event, a steering committee promoting the continuation of the conference was established and, since then, the steady growth of interest in computational argumentation research worldwide has gone hand in hand with the development of the conference itself and of related activities by its community. Since the second edition, organized by IRIT in Toulouse in 2008, plenary invited talks by world-leading researchers and a software demonstration session became an integral part of the conference programme. The third edition, organized in 2010 by the University of Brescia in Desenzano del Garda, saw the addition of a best student paper award. The same year, the new journal *Argument and Computation*, closely related to the COMMA community, was started. Since the fourth edition, organized by the Vienna University of Technology in 2012, an Innovative Application Track and a section for Demonstration Abstracts were included in the proceedings. At the fifth edition, co-organized in 2014 by the Universities of Aberdeen and Dundee in Pitlochry, the main conference was preceded by the first *Summer School on Argumentation: Computational and Linguistic Perspectives*. The same year also saw the launch of the first *International Competition on Computational Models of Argumentation* (ICCMA). Since COMMA 2016, hosted by the University of Potsdam, the COMMA proceedings are Open Access. This COMMA was also the first that included additional satellite workshops in the programme. COMMA 2018 was hosted by the Institute of Philosophy and Sociology of the Polish National Academy of Sciences in Warsaw, Poland. It included an industry afternoon bringing together businesses, NGOs, academics and students interested in practical applications of argument technologies in industry.

This year COMMA is in Italy for the second time, now hosted by the University of Perugia. It is preceded by the 4th *Summer School on Argumentation: Computational and Linguistic Perspectives* (SSA 2020), and features a demonstrations session and three satellite workshops. The *International Workshops on Systems and Algorithms for Formal Argumentation* (SAFA), initiated at COMMA 2016, has its third edition, while there is a new *Workshop on Argument Visualization.* Finally, the well-known *Workshop on Computational Models of Natural Argument*, established in 2001, has its 20th edition at COMMA 2020.

Despite these continuing traditions, COMMA 2020 is different from all preceding COMMAs in one respect: because of the coronavirus pandemic that hit the world early 2020, the entire conference and its preceding summer school have to take place online. This is, of course, a huge disappointment for the local organisers and for all participants, who had been looking forward to a great conference in the beautiful city of Perugia. Nevertheless, going online secures the continuation of the COMMA conference series, allowing the presentation and discussion of the latest research results and their publication in these proceedings.

The COMMA 2020 programme reflects the interdisciplinary nature of the field, and its contributions range from theoretical to practical (although most are theoretical). Theoretical contributions include new formal models, the study of formal or computational properties of models, designs for implemented systems and experimental research. Practical papers include applications to medicine, law, crime investigation, chatbots and online product reviews. The conference respects its historic origins by providing both abstract and structured accounts of argumentation. Some papers propose formal argument schemes for specific forms of argument. Many papers focus on the evaluation of arguments or their conclusions given a body of arguments, with a continuation of a recent trend to study gradual (e.g. probabilistic) notions of evaluation. Other papers focus on the dialogical processes by which argumentation proceeds, sometimes from a game-theoretical point of view. The focus on argument mining, which first appeared at COMMA 2016, is continued while an emerging trend this year is the use of argumentation for explainable AI.

The three invited talks also reflect the diverse nature of the field. Professor Catarina Dutilh Novaes from the Free University Amsterdam discusses the role of adversariality in argumentation from a social-epistemology perspective. Professor John Horty of the University of Maryland gives a logical analysis of defeasible reasoning about open-textured predicates in natural language and legal theory. Finally, Professor Chris Reed of the University of Dundee covers a broad spectrum from philosophical foundations via algorithmic research to technological applications.

Finally, we acknowledge the work of all those who have contributed in making the conference and its satellite events a success. We would like to thank IOS Press for publishing these proceedings and continuing to make them Open Access. As local and international sponsors of the conference, we would like to thank in random order *Fondazione Cassa di Risparmio di Perugia*, *Gruppo Nazionale per il Calcolo Scientifico* (GNCS-INdAM), the Artificial Intelligence Journal (funding scheme for promoting AI research), the Department of Mathematics and Computer Science (University of Perugia), the University of Perugia, *Confcommercio Umbria*, *Associazione Nazionale Imprese ICT* (Assintel), and *iter innovazione terziario*. We also thank the Italian Association for Artificial Intelligence (AIxIA), which supported the best student paper award. We thank the invited speakers Catarina Dutilh Novaes, John Horty and Chris Reed for their insightful and inspiring talks. We acknowledge steady support and encouragement by the COMMA Steering Committee, and are very grateful to the Programme Committee and additional reviewers, whose invaluable expertise and efforts have led to the selection of 28 full papers and 13 short papers out of a record total of 78 submissions, and 13 demonstration abstracts. The submission and reviewing process has been managed through the Easychair conference system, which we acknowledge for supporting COMMA since the first edition. Our thanks also to the COMMA 2020 workshop organisers Jodi Schneider, Matthias Thimm, Fabian Sperrle and their co-organisers, and to the summer school programme chair Massimiliano Giacomin. Last but not least, we thank all the authors and participants for contributing to the success of the conference with their hard work and commitment.

Henry Prakken (Programme chair)
Stefano Bistarelli (Conference chair)
Francesco Santini (Conference co-chair and demo chair)
Carlo Taticchi (Publicity chair)

Utrecht/Perugia, July 2020

This page intentionally left blank

Programme Committee

Additional reviewers

Gianvincenzo Alfano
Carl Corea
Martin Diller
Stefano Ferilli
Andrea Galassi
Maximilian Heinrich
Nadin Kokciyan
Jieting Luo
Tobias Mayer
Anna Rapberger
Federico Ruggeri
Kenneth Skiba
Carlo Taticchi
Markus Ulbricht
Zhe Yu

Contents

Invited Talks

Innovative Applications

Regular Papers

Demonstrations

Invited Talks

This page intentionally left blank

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200483

Conflict, Adversariality, and Cooperation in Argumentation

Catarina DUTILH NOVAES

Department of Philosophy, Vrije Universiteit Amsterdam, The Netherlands and Arché, University of St. Andrews, Scotland

Since at least the 1980s, the role of adversariality in argumentation has been extensively discussed. Some authors criticize adversarial conceptions and practices of argumentation and instead defend more cooperative approaches, both on moral and on epistemic grounds. Others retort that argumentation is inherently adversarial, and that the problem lies not with adversariality per se but with overly aggressive manifestations therof. In this paper, I defend the view that specific instances of argumentation are (and should be) adversarial or cooperative *proportionally* to pre-existing conflict. What determines whether an argumentative situation should be primarily adversarial or primarily cooperative are contextual features and background conditions, in particular the extent to which the parties involved have prior conflicting or convergent interests and goals. I articulate a notion of adversariality in terms of the relevant parties pursuing conflicting interests, and argue that, while cooperative argumentation is to be encouraged whenever possible, conflict as such is an inevitable aspect of human sociality and thus cannot be completely eliminated.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200484

Open Texture and Defeasible Semantic Constraint

John HORTY
Philosophy Department and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA

I will discuss some of the problems presented by open textured predicates for the semantics of natural language, as well as in legal theory. I will then (i) sketch an account of constraint in common law, (ii) suggest that this account can be adapted to help us understand open textured predicates as well, (iii) talk a bit about the reasoning involved in reaching decisions that satisfy this account of constraint, and (iv) show how this reasoning can be modeled in a simple defeasible logic.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200485

Argument Technology from Philosophy to Phone

Chris REED
Centre for Argument Technology, University of Dundee, UK

Computational models of argument have vast potential to transform human reasoning and decision-making wherever it occurs taking theories rooted in philosophy, developing algorithms in data science, natural language processing and AI, and engineering solutions that could end up on a phone in everyone's pocket. Fulfilling that potential, however, is enormously challenging. Sometimes, what's required is overhauling our most fundamental theories to accommodate real world phenomena: arguments in the real world, for example, most typically occur in multi-party contexts, so new theories have had to be developed to account for and handle dialogical, dialectical and interactional aspects of argumentation, whilst still supporting formally well-understood phenomena such as abstraction and acceptability, audiences and values, lexical semantics and argument structure.

At other times, though, what's required is forging ahead with a pragmatic compromise at the theoretical level that sacrifices a complete computational account of all facets of argumentation, but which nonetheless helps tackle some specific problem. Applications for supporting argumentation in domains as diverse as law, science and intelligence analysis have adopted this tack, delivering prototypes that demonstrate the potential of argument technology in different sectors.

At yet other times the problem is more a practical one: how on Earth do we assemble datasets of argumentation large enough for training supervised machine learning algorithms (let alone large enough for sheer statistical learning)? Or how can we develop, ab initio, linguistic annotation methods that can keep up with live debate? Right across its broad range of competence, the field of computational models of argument has had to pull itself up by its bootstraps, developing its own working methods, requirements, data standards, software tooling, research challenges and vocabulary.

Then again, sometimes what's required is hard academic slog to drive forward performance: the new field of argument mining is an excellent example where progress is being made in leaps and bounds, even as the challenges are being broadened from domain specific to domain independent, monolingual to multilingual, monological to dialogical. It is the determined inspiration of those working in argument mining that is responsible for results starting to come through that represent acceptable performance on realistic tasks.

But perhaps the greatest challenge, though, is what in commercial terms is known as route to market. How do we get the fruits of our labours into the hands of the hundreds of millions of people who could benefit from it? Whether contributing to the quality of national and international debate, helping the general public identify fake news, improving counterterrorism threat analysis, or enhancing democratic processes or whether nudging arguments in a pub to be a bit more accurate, helping separating couples reach more acceptable agreements, or offering an elderly parent some advice on the latest Covid rumour: wherever argument plays a role, argument technology has the potential to improve matters. Neither developing new philosophical theory nor building new phone apps (nor anything in between) is enough on its own, but with a clearer game plan for the community as a whole there is an opportunity for us to start to fulfil the potential we have collectively for making a significant difference in the world.

Innovative Applications

This page intentionally left blank

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200487

A Persuasive Chatbot Using a Crowd-Sourced Argument Graph and Concerns

Lisa A. CHALAGUINE [a] and Anthony HUNTER [a]
[a] *Department of Computer Science, University College London, London, UK*

Abstract. Chatbots are versatile tools that have the potential of being used for computational persuasion where the chatbot acts as the persuader and the human agent as the persuadee. To allow the user to type his or her arguments, as opposed to selecting them from a menu, the chatbot needs a sufficiently large knowledge base of arguments and counterarguments. And in order to make the user change their current stance on a subject, the chatbot needs a method to select persuasive counterarguments. To address this, we present a chatbot that is equipped with an argument graph and the ability to identify the concerns of the user argument in order to select appropriate counterarguments. We evaluate the bot in a study with participants and show how using our method can make the chatbot more persuasive.

Keywords. chatbots, argumentative persuasion systems, computational persuasion, natural language argumentation, concerns, argument graphs

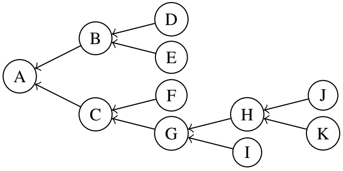
1. Introduction

Chatbots have the potential of being used as agents in argumentative persuasion systems that can engage in argumentative dialogues with users where the chatbot acts as the persuader and the user as the persuadee. Argument graphs as proposed by Dung [11] can be used as a knowledge base for the chatbot. Graphs are a useful representation to study attack and support relationships of a given set of arguments. Different kinds of semantics can be applied in order to identify the “winning” and “losing” arguments in a graph. This, however, assumes that all the possible arguments and their relationships are present in the graph.

Acquiring an argument graph raises several issues: most importantly *where* to obtain the relevant arguments for the argument graph, but also, which arguments to include in the knowledge base and how to justify the inclusion of some and exclusion of others (e.g. noise and repetition of arguments), and how to establish relations between arguments (the arcs of the graph). In our previous work [7] we presented a method and its evaluation for the acquisition of a large argument graph with over 1200 arguments via crowd-sourcing.

An argumentative chatbot could use such a graph in order to persuade a human agent to accept the bot’s stance by presenting arguments from the graph that support its stance and counter user arguments that do not. One way to utilise such a graph is by using a *menu-based* approach where the chatbot, after presenting an argument, gives the user a choice of counterarguments that the user can select from a menu [13]. Taking the argument graph shown in Figure 1 as an example, the chatbot would give argument A and then give the user arguments B and C to choose from. Suppose the user prefers

Figure 1. Argument graph where child nodes are attacking parent nodes.



argument C and selects that one. The chatbot selects a counterargument based on some criteria (or randomly) and replies with argument G and gives the user arguments H and I as countering choices, and so on. This way, the chatbot and the user would follow the arcs of the graph until (depending on the type of graph) all the arguments are used, or the user chooses an argument that has no counterarguments in the graph.

The drawback of the menu-based approach is, of course, that the user is limited to the choice of possible counterarguments presented by the chatbot, which might not include the user's preferred choice. This might limit the persuasive effect of the argumentative dialogue, as well as deny the chatbot the opportunity to acquire novel arguments on that topic which were not collected during the acquisition phase of the graph. The user arguments from the chats could then be used to extend the existing argument graph.

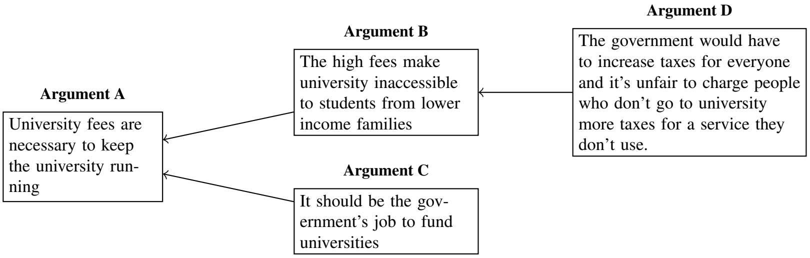
An alternative to the menu-based approach would be a *free-text* approach that allows using a similarity measure to find an argument in the graph that is similar to the user argument. If an argument similar to the one used by the user is present in the graph, the chatbot could simply reply with a counterargument from the graph. Taking the graph from Figure 1 again as an example, the chatbot would present argument A and allow the user to reply via free-text input. The user would counter with an argument that is similar to H. Suppose the chatbot counters it with K and the user replies with an argument that is similar to B. The chatbot could counter it with D or E and so on. In this case, the chatbot can jump around the graph rather than just following a single branch.

However, this poses two questions for the free-text approach: firstly, how to deal with a user argument that is not present in the graph. Not finding a match to the user's argument can be expected to be a common phenomenon given that it cannot be assumed that all arguments on that topic are contained in the graph. The versatility of natural language with its seemingly infinite number of ways to rephrase something, is also likely to limit the ability of the chatbot to find a similar argument in the graph. A second problem is that, even if the user's argument is present in the graph, a counterargument must be chosen so as to increase the persuasive effect of the dialogue.

A potential answer to address the first question is for the chatbot to present an argument that is not necessarily a counterargument to the user's argument. This way the dialogue would resemble argumentation as it would happen in real life between two people: if two human agents engage in an argumentative dialogue, just because one presents an argument the other cannot counter, the dialogue is not necessarily ended prematurely. The other agent might switch topics and present a new argument he or she believes in, without referencing and directly countering the previous argument. Another example would be product reviews where reviewers present a range of pro and con arguments. The judgement is not about whether all counterarguments were answered or not, but whether the pro arguments outweigh the con arguments.

An answer to the second question could come from taking the *concerns* of the user into account [8,14,13], a concern being a matter of interest or importance to the user.

Figure 2. Simple argument graph with arguments *B* and *C* attacking argument *A* and argument *D* attacking argument *B*.


Argument A
University fees are necessary to keep the university running
Argument B
The high fees make university inaccessible to students from lower income families
Argument C
It should be the government's job to fund universities
Argument D
The government would have to increase taxes for everyone and it's unfair to charge people who don't go to university more taxes for a service they don't use.

The notion of a concern seems to be similar to the notion of a value. Values have been used in a version of abstract argumentation called value-based argumentation frameworks (VAFs) [3]. For this, when selecting the counterargument, the chatbot could select the counterargument that addresses the more important concerns of the user. In our previous works, however, the concerns of the user were either known in advance [8] or the chatbot did not allow free-text input and the concerns that were addressed by each argument in the graph were known [14,13]. During the chat with a chatbot that allows free-text input, however, the concerns that are addressed by the user arguments need to be classified during the chat in order to choose a suitable counterargument accordingly.

In this paper, we present a free-text chatbot that can engage in an argumentative dialogue in order to persuade the user to accept the bots stance. The chatbot is equipped with a crowd-sourced argument graph with automatically assigned concerns to each argument and a concern classifier that can assign concerns to the user arguments during the chat. With the help of this chatbot, we show that it is not necessary to follow the arcs of a graph during each dialogue move in order to create reasonable and relevant dialogues, and that concerns can be automatically detected and used in order to choose appropriate counterarguments to increase the persuasiveness of the dialogue.

The rest of the paper is structured as follows: Section 2 presents our previous work that the current study builds upon; Section 3 gives the aim of the paper and the hypotheses; Section 4 describes the chatbot architecture that was used for the experiments; Section 5 describes the experiments that were conducted with the chatbot including their results, and in Section 6 we discuss and conclude our findings.

2. Previous work

2.1. *Crowd Sourced Argument Graph*

The purpose of argumentation is to exchange different viewpoints or opinions, handle conflicting information and make informed decisions. A situation involving argumentation can be represented by a directed graph, as proposed by Dung [11]. Each node represents an argument, and each arc denotes an attack by one argument on another. Such a graph can then be analysed to determine which arguments are acceptable according to some general criteria [4,2]. Figure 2 shows such an argument graph and the attack relationships between the arguments.

Argument graphs are extensively studied in the computational argumentation literature. Their acquisition, however, tends to be neglected. In [7] we present a method of automatically acquiring a large argument graph via crowd-sourcing[1]. We evaluated that method in a case study on the topic of UK university fees. The graph contains 5 levels of depth, starting with the root statement *“University fees in the UK should be kept at 9k pounds”* (Depth 0). The next level of depth (Depth 1) contains arguments that counter the root statement. The arguments in Depth 2 counter the arguments in Depth 1 and so on. Our graph contains 1288 arguments with each argument on average having 3 counterarguments, apart from the last level of depth. Depths 1-4 of the acquired argument graph are used as the knowledge base of the chatbot presented in this paper.

2.2. *Concerns*

We have confirmed the long-held view that taking the concerns of the user into consideration increases the persuasiveness of the dialogue in our previous works [8,14,13]. Arguments can raise or address various concerns for the persuadee that need to be accounted for. A persuader might present a perfectly valid argument to a university student (persuadee), e.g. *“If someone decides to go into higher education, the general public should not be expected to pay for it via taxes.”*. The persuadee might not even disagree with this argument, however, she is very likely to be concerned about her finances due to her personal debt and therefore this argument may have no impact on her stance. If, however, the persuader presents an argument that addresses her concern like *“If you have a student loan in the UK, it will not appear on your credit report. So, when you are applying for a credit card, loan or mortgage your student loan will not make an appearance.”* it is more likely to change her stance. This is not surprising, however, concerns are often ignored when judging the effectiveness of arguments or choosing a strategy. Some studies that make use of different personality traits of the user attributes in order to evaluate what sort of argument might be more effective for this particular person (for examples see [16,10,21,18]). However, computational argumentation largely focuses on sentimental [9], rhetorical [12] and structural [5] attributes of the argument, rather than attributes about the user.

In the following sections, we outline our hypotheses and describe how we utilise the argument graph and the notion of concerns in order to build a chatbot that can engage in persuasive dialogues, and the experiments conducted with the chatbot.

3. Hypotheses

In this paper, we chose UK university Fees as a case study. We have developed a chatbot that utilises a crowd-sourced argument graph, described in [7], as the knowledge base. The chatbot uses concerns to make strategic moves in order to engage in argumentative dialogues with users to persuade them to accept that chatbot’s stance (that university fees should be kept).

Given this setting, we want to test two questions: Firstly, whether the crowd-sourced argument graph can be used as a chatbot knowledge base that allows free-text input. This means that the graph contains at least some common arguments that the user might use,

[1] https://github.com/lisanka93/Argument_Graph_Corpus

and the resulting dialogues are therefore of an appropriate length and quality, and that the users perceive the chatbot arguments as relevant. And secondly, whether the chatbot can automatically identify the concerns addressed by the user argument and whether replying with counterarguments that address the same concern, increases the persuasiveness of the chat. We summarise these points in the following two hypotheses:

H1 A crowd-sourced argument graph can be used as a knowledge base for a persuasive chatbot allowing free text input by the users. The resulting chats are of appropriate length and quality, and the chatbot arguments perceived as relevant by the users.

H2 A concern raised or addressed by a given user argument can be automatically identified in order to give appropriate counterarguments that address the same concern and thereby increase the persuasiveness of the dialogue.

In the remainder of this paper we describe the design of our chatbot that was used for the argumentative dialogues and explain the experiments conducted with the chatbot in order to test our hypotheses.

4. Chatbot Design

We developed two versions of our chatbot, one that classifies the concern of the user argument and takes it into account when presenting counterarguments (strategic), and one that did not (baseline).

4.1. Argument Graph

The argument graph described in Section 2.1 is used as the chatbot's knowledge base. We only use the depths 1-4, since depth 5 does not have any counterarguments. Depths 1 and 3 contain arguments against keeping university fees, while depth 2 (attacking depth 1 arguments) and 4 (attacking depth 3 arguments) contain arguments that support the stance of keeping university fees.

When the user types in an argument (source argument), the chatbot uses a similarity measure in order to find the closest match of the user argument in the graph (target argument). We used cosine-similarity as a similarity measure [19]. Cosine similarity is a metric used to measure how similar the vector representation of two texts are. It measures the cosine of the angle between two vectors. The smaller the angle, the higher the cosine similarity. We used a threshold of 0.9 for measuring the similarity of two arguments. If the chatbot finds an argument in the graph that has a similarity of 0.9 or above compared with the source argument, the chatbot chooses one of the counterarguments that attack the target argument in the graph as a response. This happens at every dialogue turn, meaning that the target argument can be either in depth 1 or depth 3 of the graph.

4.2. Default Arguments

In case no target argument is found, we also acquired arguments for keeping university fees, where the root statement is the opposite to our main argument graph *"University fees in the UK should be abolished"*. It is therefore a very shallow graph with only one level of depth where the arguments that attack the root argument are for keeping the

Table 1. Types of concern for the topic of charging university tuition fees

Concern	**Description of what concern deals with**
Student Finance	Finances of students, including tuition fees, student debts, life costs etc.
Government Finance	Government finances, including general taxation, government spending etc.
Employment	Careers and employability of students and the general job market.
Free Education	Whether higher education is a human right and should be free or not.
Fairness	Whether something is fair or not (using a general understanding of fairness), including equal and just treatment of individuals.

fees. We also used crowd-sourcing for the acquisition and voting in order to select the best arguments. The best 7 arguments were used as *default* arguments, which the chatbot can use if no match is found. These arguments are therefore not counterarguments in the traditional sense, as they do not refer to or address the source argument but instead "change topic" and present a new issue in the debate. We also added phrases like *"Ok but", "I still think"* and *"Don't you think that"* to the beginning of the default arguments to indicate a deviation from the topic occurs.

4.3. Concern Labelling and Classification

The baseline chatbot uses the argument graph and default arguments during the chat with the user and does not make use of concerns. The strategic chatbot, however, classifies the concern of the source argument and chooses one of the attackers of the target argument that addresses the same concern.

During the acquisition of the argument graph described in [7], only arguments were included in the graph that contained *topic words*. These are words that we considered meaningful in the given context. The choice of suitable topic words depends entirely on the domain and their choice is left to the researchers' discretion and their knowledge of the domain. The topic words in the argument graph were: *loan, debt, job, tax, free, accessible, affordable, government, scholarship, interest, career* and *background*. We grouped topic words that address the same or similar issues into 5 concerns: **Student Finance** (loan, debt, scholarship, interest), **Government Finance** (government, tax), **Employment** (job, career), **Free Education** (free) and **Fairness** (affordable, accessible, background). Apart from the concern *free*, the concerns were taken from [14]. The definitions are given in Table 1.

We took the arguments from the argument graph, as well as the user arguments from the chats with the baseline chatbot that contained any of the topic words, to train a concern classifier using the Python Scikit-learn library[2]. The classifier uses logistic regression and a tf-idf feature representation in order to predict the concern of the incoming user argument. We extract the top two concern predictions. If the top prediction is over 0.7 the argument is labeled with one concern, otherwise with two. If a target argument in the graph is found, the chatbot chooses one of the attackers of the target argument that addresses the same concern as counterargument. If a user argument is labeled with two concerns, an attacker is chosen that addresses one of the concerns, with priority given to the concern with the higher predicted value.

[2] https://scikit-learn.org

It could be argued that since the arguments in the graph are labelled with concerns, the source argument addresses the same concerns as the target argument in the graph and hence no classifier is needed as one could take the concerns of the target argument. However, the concerns of the target argument are not necessarily the same as the user's free-text argument, despite being similar. For example, the target argument in the graph *"Universities should be accessible to all, not just those that can afford it, or are not scared away from the high debt after their studies"* would be labeled with both concerns *fairness* and *student finance*. A similar source argument *"Universities should be accessible to everyone who wants a higher education, not just those that can afford it"* does not address the concern *student finance* and would be labeled with *fairness* only by the classifier.

If no match in the graph is found or none of the counterarguments of the target argument address the same concern, the chatbot replies with a default argument.

5. Evaluation of the Chatbot

The purpose of the chatbots was to test both of our hypotheses. The chatbots were deployed on Facebook via the Messenger Send/Receive API. For more on the implementation of such chatbots see [6]. For each chatbot we recruited 50 participants via Prolific[3], which is an online recruiting platform for scientific research studies. Before the chat the users were directed to a Google Form and asked whether they *strongly disagreed, disagreed, neutral, agreed* or *strongly agreed* that university fees should be kept[4].

After submitting their answers they were redirected to the Facebook page where they could begin the chat. The chatbot started the chat by asking why the user believed that university fees should be abolished. The user, therefore, presented their first argument. The chatbot then replied with either a counterargument from the argument graph or a default counterargument, depending on whether a similar argument was found in the graph or not. If a similar match was found, the baseline chatbot replied with a randomly selected counterargument from the direct attackers of the target argument in the graph. The strategic chatbot, however, selected an attacker from the graph that addressed the same concern as the user argument (if such an argument exists). If no match was found, both chatbots replied with a default argument.

If the user response was shorter than 6 words, the chatbot queried the user to expand on their answer. However, if the user agreed with an argument the chatbot gave, for example by sending *"I agree"*, the chatbot would not ask to expand despite the message being shorter than 6 words, and instead replied with a default argument.

The chatbot would eventually end the chat as soon as all default arguments were used up and no match in the graph was found. The users were, however, advised that they could end the chat anytime by sending the word *"stop"*. At the end of the chat the chatbot presented the user with a link that redirected them to a second Google Form where they were asked a series of questions[5]:

[3] `https://prolific.co`

[4] For the baseline chatbot only 2 people selected *agree* and none for the strategic one. 98% of participants therefore did not share the chatbots stance before the chat.

[5] Further questions were asked but analysis of the answers is left to future work

Table 2. Answers to first three questions for baseline and strategic groups

Chatbot	**Understood (Q1)**			**Relevant Args (Q2)**			**Points addresses (Q3)**		
	Yes	No	Sometimes	Yes	No	Some	Yes	No	Some
Baseline	16	4	30	21	3	26	13	15	22
Strategic	15	6	29	31	1	18	10	14	26

1. Did you feel understood by the chatbot? (Yes/No/Sometimes)
2. Did you feel that the chatbot's arguments were relevant? (Yes/No/Sometimes)
3. Do you feel like all your points were addressed? (Yes/No/Some of them)
4. How much do you agree that fees in the UK should be kept as they are? (Strongly disagree - strongly agree)

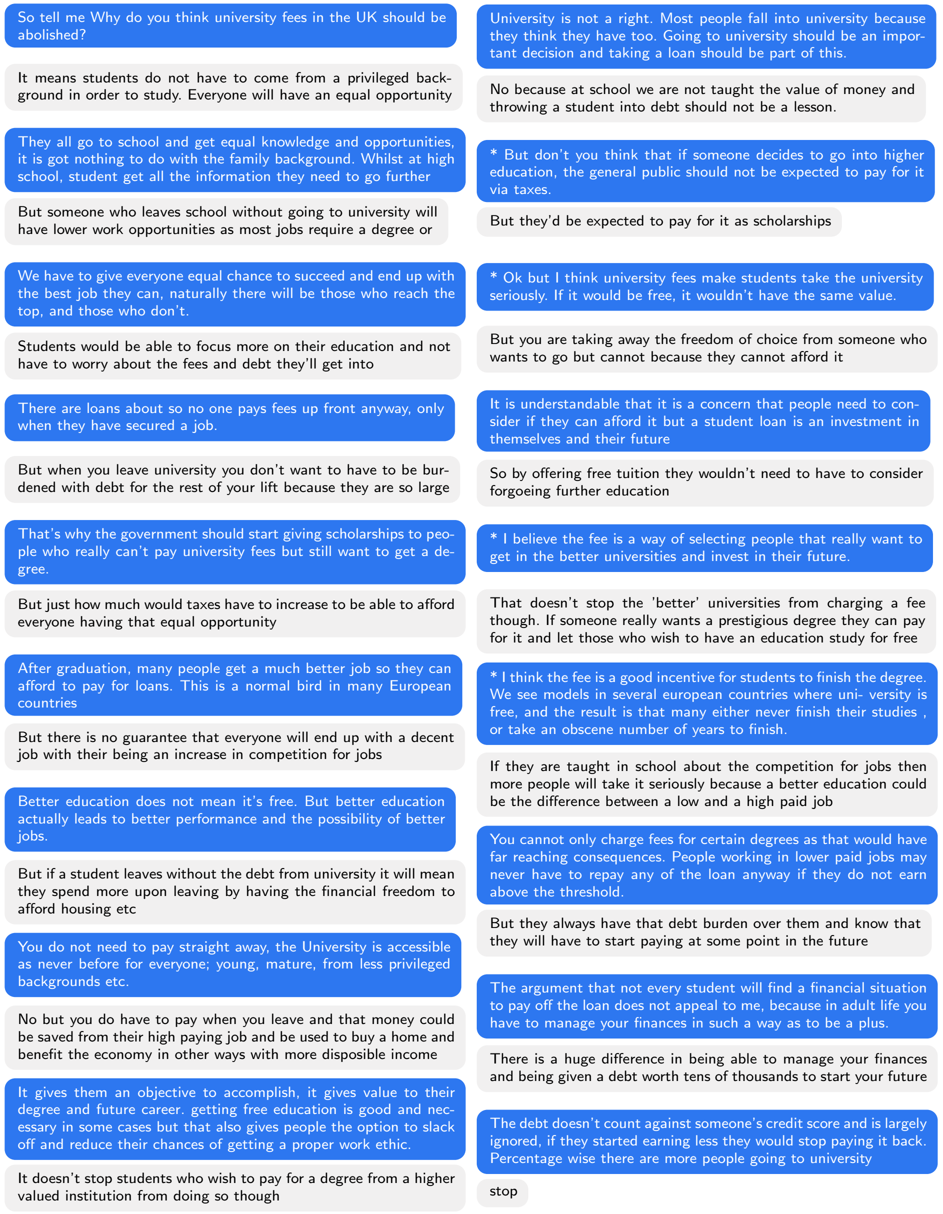
Questions 1-3 were used to test our first hypothesis and judge the relevance, length and quality of the chats, and question 4 was to test our second hypothesis and compare the persuasiveness of the baseline chatbot to the strategic chatbot. Table 2 shows the results for the first three questions for the baseline and the strategic groups. One can see that the majority of the participants considered the chatbot's arguments as relevant in most cases, and answered the first three questions with either *yes* or *sometimes*. Interestingly there is a 50% increase in the perception of relevance for the strategic chatbot, while the numbers for questions 1 and 3 remained almost the same. This is a statistically significant difference with a p-value of 0.045 using Chi-Square. Using concerns, therefore, makes the arguments more relevant.

Regarding questions 1 and 3, given that the chatbot does not use natural language generation and only relies on finding matches in an argument graph, it is not able to address novel arguments or expand on existing ones by giving more information. It is therefore not surprising that the numbers for these questions remained approximately the same. The given results, however, are promising given that the chatbot solely relies on a similarity measure in order to pick counterarguments from a crowd-sourced graph or to pick default arguments in case no match is found.

Regarding the length of the chats, chats with the chatbots on average lasted 24 turns (disregarding the chatbot querying the user to expand). This means that the chatbot, on average, gave 12 arguments, 7 of which were default arguments and 5 from the graph. This supports our first hypothesis that a crowd-sourced argument graph can be used as a chatbot knowledge base and that the resulting argumentation dialogues are of satisfactory length and quality, with perceived relevance of the arguments being 50% higher during chats with the strategic chatbot. An example of a chat can be seen in Figure 3. All chatlogs and the code for the chatbot can be found in [1].

Using the replies for question 4 we calculated the change in stance for both groups in order to evaluate the persuasiveness of the strategic chatbot. The persuasion goal of the chatbot was that UK university fees should be kept. Since participants were given the choice of 5 stances ("strongly disagree" to "strongly agree") before and after the chat, they could either change their stance to the better, to the worse or not at all. The change in stance is the final stance minus the original stance. We call the units of this measure *change in stance* (CS) points. For example, "strongly disagree" being 1 and "strongly agree" being 5, if a participant changed her stance from "disagree" (2) to "neutral" (3) after chatting with the bot this counts as 1 CS point, whereas changing from "neutral" to "disagree" counts as -1 CS point. Persuasion means that a person who was against

Figure 3. Example chat between a participant and the strategic chatbot. Chatbot arguments are in the dark boxes and user arguments in the light boxes. Default arguments are indicated with an *. The chat begins in the left column and continues in the right one.



So tell me Why do you think university fees in the UK should be abolished?

It means students do not have to come from a privileged background in order to study. Everyone will have an equal opportunity

They all go to school and get equal knowledge and opportunities, it is got nothing to do with the family background. Whilst at high school, student get all the information they need to go further

But someone who leaves school without going to university will have lower work opportunities as most jobs require a degree or

We have to give everyone equal chance to succeed and end up with the best job they can, naturally there will be those who reach the top, and those who don't.

Students would be able to focus more on their education and not have to worry about the fees and debt they'll get into

There are loans about so no one pays fees up front anyway, only when they have secured a job.

But when you leave university you don't want to have to be burdened with debt for the rest of your lift because they are so large

That's why the government should start giving scholarships to people who really can't pay university fees but still want to get a degree.

But just how much would taxes have to increase to be able to afford everyone having that equal opportunity

After graduation, many people get a much better job so they can afford to pay for loans. This is a normal bird in many European countries

But there is no guarantee that everyone will end up with a decent job with their being an increase in competition for jobs

Better education does not mean it's free. But better education actually leads to better performance and the possibility of better jobs.

But if a student leaves without the debt from university it will mean they spend more upon leaving by having the financial freedom to afford housing etc

You do not need to pay straight away, the University is accessible as never before for everyone; young, mature, from less privileged backgrounds etc.

No but you do have to pay when you leave and that money could be saved from their high paying job and be used to buy a home and benefit the economy in other ways with more disposible income

It gives them an objective to accomplish, it gives value to their degree and future career. getting free education is good and necessary in some cases but that also gives people the option to slack off and reduce their chances of getting a proper work ethic.

It doesn't stop students who wish to pay for a degree from a higher valued institution from doing so though

University is not a right. Most people fall into university because they think they have too. Going to university should be an important decision and taking a loan should be part of this.

No because at school we are not taught the value of money and throwing a student into debt should not be a lesson.

* But don't you think that if someone decides to go into higher education, the general public should not be expected to pay for it via taxes.

But they'd be expected to pay for it as scholarships

* Ok but I think university fees make students take the university seriously. If it would be free, it wouldn't have the same value.

But you are taking away the freedom of choice from someone who wants to go but cannot because they cannot afford it

It is understandable that it is a concern that people need to consider if they can afford it but a student loan is an investment in themselves and their future

So by offering free tuition they wouldn't need to have to consider forgoeing further education

* I believe the fee is a way of selecting people that really want to get in the better universities and invest in their future.

That doesn't stop the 'better' universities from charging a fee though. If someone really wants a prestigious degree they can pay for it and let those who wish to have an education study for free

* I think the fee is a good incentive for students to finish the degree. We see models in several european countries where uni- versity is free, and the result is that many either never finish their studies , or take an obscene number of years to finish.

If they are taught in school about the competition for jobs then more people will take it seriously because a better education could be the difference between a low and a high paid job

You cannot only charge fees for certain degrees as that would have far reaching consequences. People working in lower paid jobs may never have to repay any of the loan anyway if they do not earn above the threshold.

But they always have that debt burden over them and know that they will have to start paying at some point in the future

The argument that not every student will find a financial situation to pay off the loan does not appeal to me, because in adult life you have to manage your finances in such a way as to be a plus.

There is a huge difference in being able to manage your finances and being given a debt worth tens of thousands to start your future

The debt doesn't count against someone's credit score and is largely ignored, if they started earning less they would stop paying it back. Percentage wise there are more people going to university

stop

Table 3. Change of stance measured by number of participants and CS points

Chatbot	Baseline			Strategic		
Change in stance	Negative	No Change	Positive	Negative	No Change	Positive
No. of participants	5	41	4	1	26	23
Change in CS points	-5	0	5	-1	0	32

keeping university fees before the chat changed her stance to the positive and that her CS points score is positive.

Table 3 shows the number of participants who changed their stance to the worse (negative), to the better (positive), and that did not change their stance at all (no change) for both chatbots, as well as the number of total CS points. We can see that 23 people changed their stance to the better when chatting with the strategic chatbot with a total of 32 CS points, meaning that some participants changed their stance by more than 1 CS point (e.g. from *disagree* to *agree*). If counting the total number of CS points, also including the participants that changed their stance to the worse, the strategic chatbot achieved a total change of 31 CS points whereas for the baseline the total number of CS points is 0.

It could be argued that a change from *strongly disagree* to *disagree* is not a remarkable change in stance despite resulting in the change of 1 CS point, whereas changing someone's stance from *disagree* to *neutral* or even better, *agree* is a much stronger shift in stance. However, for the strategic chatbot, only 2 participants changed their stance from *strongly disagree* to *disagree*, while the remaining 21 participants changed their stance from disagreement (strongly or not) to neutral (16 participants), from neutral to agreement (3 participants) and from disagreement to agreement (2 participants).

We used the number of participants who changed their stance to the positive in order to calculate the statistical significance of the difference between the control group that chatted with the baseline chatbot and the group that chatted with the strategic chatbot using the Chi-Square test. All results were statistically significant with a p-value of 0.00017. The results support our second hypothesis, that concerns can be automatically classified based on the use of topic key words which can be seen as a good indicator of the concerns being addressed or raised by the arguments. Presenting arguments that address the user's concern is more likely to have a positive impact on their stance, than presenting arguments that ignore the user's concern.

6. Discussion

Our contribution in this paper is twofold. Firstly, we have shown that a crowd-sourced argument graph can be utilised as a knowledge base for a chatbot that engages in argumentative dialogues. The resulting chats are of good length and quality and are perceived as relevant by the users. And secondly, we have shown that concerns can be automatically identified in order to give suitable counterarguments that address the same concern and thereby significantly increase the persuasiveness of the dialogue. Additionally, we have shown that the chatbot can jump around in the graph, without systematically following each arc and only use arguments that are connected via an attack relationship.

To date, at least two arguing chatbots have been presented in the literature: a chatbot Debbie, that uses a similarity algorithm to retrieve counterarguments [17] and Dave that used retrieval- and generative-based models [15]. Debbie's knowledge base consists of a subset of the qualitatively best arguments from the corpus created by Swanson et al [20] which is a combination of online political debates, Internet Argument Corpus (IAC), [22] and dialogues from online debate forums. Dave's knowledge base consists only of the IAC. Our chatbot, however, is different in several ways: firstly, our knowledge base consists of a previously crowd-sourced argument graph. And secondly, the aim of Dave/Debbie was to keep the conversation going, whereas we were interested in persuading the user to accept our chatbot's stance.

This study can be seen as a partial extension of the work in [14] where a chatbot was used to persuade the user to accept the chatbot's stance on the topic of university fees in the UK. The argument graphs that were used as the chatbot's knowledge base were hand-crafted and manually labeled. The chatbot also did not allow free-text input and was strictly following the arcs of the argument graph. The chatbot presented in this paper allows free-text input and uses a similarity measure to extract similar arguments from the graph and does not restrict the selection of arguments to a single path in the graph. If a match is not found, the chatbot replies with an argument that is not contained in the original graph. Our evaluation showed this approach performed well and shows that it is not necessary to and, in fact, often impossible to establish all possible relationships in a big argument graph. Therefore, instead of following a single path through the graph and only allowing the user to choose arguments that are present in the graph, one can search for a similar argument at each dialogue step without relying on a connecting arc between the new user argument and the previously given chatbot argument. And to avoid ending the chat prematurely if no similar user argument is found, default arguments can be introduced to keep the chat going.

We faced the additional challenge of having to automatically identify the concern of the user arguments during the chat. We showed that by grouping the most common meaningful words of the argument graph (topic words) into concerns, one can train a concern classifier on the graph arguments that can be used by the chatbot in order to improve its persuasive effect.

The advantage of using a crowd-sourced argument graph as a knowledge base is that it does not require professional research but solely relies on the input of participants and can be acquired quickly. This method also scales easily which allows obtaining many arguments from different people, and thereby create large and comprehensive argument graphs. There are, however, also potential risks to consider. For example, the spread of invalid arguments which, despite being popular, might contain wrong information. Therefore, in the future, we want to investigate methods on how to utilise the argument graph to improve the quality and persuasive effect of the chats even further. The chatbot could, for example, identify invalid or unpopular arguments and delete them from the graph. The bot could also learn which are the more persuasive arguments and use those more often in the future.

Acknowledgements

The first author is funded by a PhD studentship from the EPSRC. The authors would like to thank Sylwia Polberg for valuable feedback on earlier versions of this paper.

References

[1] https://github.com/lisanka93/university_fees_bot.

[2] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. In *Knowledge Engineering Review 26(4)*, pages 365–410, 2011.

[3] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[4] P. Besnard, A. J. García, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni. Introduction to structured argumentation. *Argument and Computation*, 5(1):1–4, 2014.

[5] F. Boltuzic and J. Snajder. Back up your stance: Recognizing arguments in online discussions. In *Proc. of the First Workshop on Argumentation Mining*, pages 49–58, 2014.

[6] L. A. Chalaguine and A. Hunter. Chatbot design for argument harvesting. In *Proc. of the Workshop on Systems and Algorithms for Formal Argumentation COMMA'18*, pages 457–458, 2018.

[7] L. A. Chalaguine and A. Hunter. Knowledge acquisition and corpus for argumentation-based chatbots. In *Proc. of the 3rd Workshop on Advances In Argumentation In Artificial Intelligence*, pages 1–14, 2019.

[8] L. A. Chalaguine, A. Hunter, F. L. Hamilton, and H. W. W. Potts. Impact of argument type and concerns in argumentation with a chatbot. In *Proc. of the 31st International Conference on Tools with Artificial Intelligence*, pages 1557–1562, 2019.

[9] Y. Choi and C. Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proc. of the Association of Computational Linguistics*, pages 793–801, 2008.

[10] T. Ding and S. Pan. Personalized emphasis framing for persuasive message generation. In *Proc. of the European Chapter of Computational Linguistics*, pages 1432–1441, 2016.

[11] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[12] I. Habernal and I. Gurevych. Which argument is more convincing? Analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proc. of the Association of Computational Linguistics*, pages 1589–1599, 2016.

[13] E. Hadoux and A. Hunter. Comfort or safety? Gathering and using the concerns of a participant for better persuasion. *Argument & Computation*, pages 1–35, 2019.

[14] A. Hunter, S. Polberg, and E. Hadoux. Strategic argumentation dialogues for persuasion: Framework and experiments based on modelling the beliefs and concerns of the persuadee. Technical report, University College London, 2020.

[15] D. T. Le, C. Nguyen, and K. A. Nguyen. Dave the debater: a retrieval-based and generativeargumentative dialogue agent. In *Proc. of the 5th Workshop on Argument Mining*, pages 121–130, 2018.

[16] S. Lukin, P. Anand, M. Walker, and S. Whittaker. Argument strength is in the eye of the beholder: Audience effect in persuasion. In *Proc. of the European Chapter of Computational Linguistics*, pages 742–753, 2017.

[17] G. Rakshit, K. K. Bowden, L. Reed, A. Misra, and M. Walker. Debbie,the debate bot of the future. In *Advanced Social Interaction with Agents - 8th International Workshop on Spoken Dialog Systems*, pages 45–52, 2017.

[18] R. Santos, G. Marreiros, C. Ramos, J. Neves, and J. Bulas-Cruz. Using personality types to support argumentation. In *Proc. of the Sixth International Workshop on Argumentation in Multi-Agent Systems*, pages 292–304, 2009.

[19] A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24:35–43, 2001.

[20] R. Swanson, B-Ecker, and M. Walker. Argument mining: Extracting arguments from online dialogue. In *Proc. of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 217–226, 2015.

[21] S. Villata, E. Cabrio, I. Jraidi, S. Benlamine, M. Chaouachi, C. Frasson, and F. Gandon. Emotions and personality traits in argumentation: An empirical evaluation. *Argument and Computation*, 8(1):61–87, 2017.

[22] M. A. Walker, P. Anand, J. F. Tree, R. Abbot, and J. King. A corpus for research on deliberation and debate. In *Proc. of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 812–817, 2012.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200488

An Explainable Approach to Deducing Outcomes in European Court of Human Rights Cases Using ADFs

Joe COLLENETTE, Katie ATKINSON and Trevor BENCH-CAPON
Department of Computer Science, University of Liverpool, UK

Abstract. In this paper we present an argumentation-based approach to representing and reasoning about a domain of law that has previously been addressed through a machine learning approach. The domain concerns cases that all fall within the remit of a specific Article within the European Court of Human Rights. We perform a comparison between the approaches, based on two criteria: ability of the model to accurately replicate the decision that was made in the real life legal cases within the particular domain, and the quality of the explanation provided by the models. Our initial results show that the system based on the argumentation approach improves on the machine learning results in terms of accuracy, and can explain its outcomes in terms of the issue on which the case turned, and the factors that were crucial in arriving at the conclusion.

Keywords. Legal case-based reasoning, Abstract Dialectical Frameworks, Explanation

1. Introduction

Reasoning with legal cases has always been a central topic of AI and Law. The basic feature of this sort of reasoning is that there is a body of case law - previous decisions - and a new case must be decided in the light of these precedents. A hearing of a case takes the form of each side presenting arguments, typically based on these precedent decisions, as to why they should win[1].

The adversarial nature of law naturally led to such reasoning being seen in terms of argumentation, with arguments being presented for both sides and the user being expected to choose between them. Examples of such systems are McCarty's TAXMAN [3], HYPO [4] and the many systems influenced by HYPO [5]. Advantages of this approach were that it was able to offer a model of legal reasoning, and provide a full explanation of the reasoning. Early systems only presented the arguments without offering a decision, but it is also possible to use such systems for prediction by attempting to assess the com-

[1] Although precedent cases are more especially associated with the common law traditions of UK and USA, precedents also play a role in European civil law, guiding the permissible interpretations. The role of precedents in Civil Law is discussed in [1], who quotes [2]: "precedent now plays a significant part in legal decision making and the development of law in all the countries and legal traditions that we have reviewed," including nine civil law jurisdictions (e.g., Germany, France, Italy, and Spain) and two common law jurisdictions (the United Kingdom and New York State)".

peting arguments. An example from the rule based tradition is [6] and an example from the case based tradition is [7]. The argumentation approach continues to be popular: recent work includes [8], [9] and [10] from ICAIL 2019 and a commercial implementation [11].

Although this approach has proven successful in modelling aspects of legal reasoning and providing outcomes backed with full explanations that are readily understandable in legal terms, it can require a good deal of expertise to analyse the precedents and construct a model of the case law, and so encounters the classic *knowledge engineering bottleneck*. Given that the problem is essentially classifying new examples on the basis of a large amount of previous cases, it would seem that the use of machine learning techniques might enable this bottleneck to be avoided. Therefore even several decades ago efforts were made to apply machine learning techniques to case law, such as [12], and [13]. This did produce some apparently promising results, while attracting some theoretical criticism [14]. But the practical difficulties were perhaps even more decisive: at that time the machine learning techniques could not be applied to natural language, and so analysis was needed to identify the relevant features of the cases and to ascribe them to the training examples, and the amount of case data available in suitable form in those pre-internet days was limited. For these reasons these approaches did not become mainstream.

Recently, however, because of the improvement in machine learning techniques and the vastly increased availability of data, there has been a marked revival of interest in these techniques. There is an annual Competition on Legal Information Extraction/Entailment (COLIEE)[2], the sixth edition of which was held in 2019. Some projects have attracted considerable attention, perhaps most notably [15], which attracted a great deal of media interest[3].

We may take [15] as representative of this new trend in using learning from large data sets to predict case outcomes. It was designed to classify cases heard before the European Court of Human Rights into violations and non-violations, according to three Articles of the European Convention on Human Rights. It was claimed that “our models can predict the court’s decisions with a strong accuracy (79% on average)”. The study used a dataset comprising 250 cases related to Article 3, 80 to Article 6 and 254 to Article 8, balanced between violation and non-violation cases. Textual information was represented using contiguous word sequences, i.e., N-grams, and topics, and used to train Support Vector Machine (SVM) classifiers [16]. The same domain was also examined in [17] where broadly similar results were obtained using more cases, with a slightly lower success rate (75% on average). That paper, however, reported another experiment, designed to test the usefulness for predicting future cases on the basis of data about the past. In this experiment they trained on data up to 2013 and tested on data from 2014-5 and from 2016-7. This showed a decrease in performance. For example, consider Article 6: whereas using all the data without regard to time produced a success rate of 80%, this fell to 64% for the 2014-5 data and to 63% and to only 59% for 2016-7 cases when the training set is all before 2005.

[2] https://sites.ualberta.ca/ rabelo/COLIEE2019/

[3] See for example, Artificial intelligence ’judge’ developed by UCL computer scientists, *Guardian*, October, 2016. https://www.theguardian.com/technology/2016/oct/24/artificial-intelligence-judge-university-college-london-computer-scientists.

This experiment bears out an important criticism made of the machine learning for prediction approach in works such as [18], that because case law evolves over time, changes may not be picked up by a system trained on historic data. Another key criticism is the success rate: although [15] describes 79% as "strong accuracy", having over a fifth of cases decided wrongly would not be acceptable in a legal system. Finally there is the lack of explanation: in law the parties to the suit have a *right* to explanation [19]. Moreover this explanation should be couched in legally relevant terms. No explanation is provided in [15], beyond a list of 20 most frequent words, listed in order of their SVM weight. These lists do not really provide a satisfying explanation, typically containing names of months and common words such as "court, applicant, article, judgment, case, law, proceeding, application, government", which head the list for topic 23 of Article 6.

For these reasons we decided to tackle the domain of [15] using argumentation based methods. Our aim is to show that, using proportionate effort, a system can be built with these methods which is more accurate, provides adequate explanation facilities and is structured so as to be amenable to changes in the case law [20]. Unlike machine learning approaches, however, this system will require users to have sufficient legal expertise to answer questions about the case. Our chosen approach is to use the ANGELIC methodology [21], of which we will give a brief overview in the next section.

2. Background: the ANGELIC methodology

The ANGELIC methodology was designed to encapsulate the knowledge of a body of case law in a way which would support argumentation and facilitate future modification as that case law evolved. It is described in [21] and was used to support the development of applications in [11] and [22]. Over the years, representing cases in terms of factors had become the *de facto* standard approach to reasoning with legal cases. These factors can be usefully organised into a factor hierarchy in the manner of [23] to show how they relate to issues in the domain. In a factor hierarchy the children of a node are reasons for the presence or absence of their parent. In ANGELIC this factor hierarchy is interpreted as an Abstract Dialectical Framework (ADF) [24]. In an ADF each node has acceptance conditions which specify the conditions under which a node will be accepted or rejected in terms of its children. The factor hierarchy and the ADF have the same structure, nodes with positive and negative children, but the ADF enables the relation between a parent and its children to be specified precisely. In ANGELIC the acceptance conditions take the form of a prioritised set of conditions for the acceptance or rejection of the node, each individually sufficient, and collectively necessary. The effect is that once the values of the children have been established, we can use the acceptance conditions to provide arguments to accept or reject the parent. In turn, as we descend the tree, we get arguments to accept and reject the children. The leaves of the tree take the form of questions, to be answered from the facts of the case. This produces the kind of argument-subargument structure found in ASPIC+ [25]. The relation between the ADFs of ANGELIC and ASPIC+ is discussed in [26]. Because the acceptability of the parent depends only on the acceptability of its children, we get a highly modular structure to support maintenance and to accommodate changes [20].

The ADF produced using ANGELIC is always a tree and the nature of the nodes changes as we descend the tree, as described in [27]. The root of the tree is the *verdict*, the overall question to be decided, such as whether there is a violation. The next layers represent *issues*, the various broad ways in which an answer to the question must be considered, such as the various ways in which an Article may be violated. Often, the issues can be found in the legislation. Below these are the *abstract factors*, the various considerations found relevant to the issues in previous cases. Below these are the *base level factors*, which are the legal facts as accepted by the court. These in turn unfold into the plain facts of the case, so that the base level factors can be resolved by posing questions to the user.

Thus the ADF produced by ANGELIC provides a systematic and modular representation of the case law, from which an argument from case facts to issues and verdict can readily be recovered.

3. Domain application: Cases on Article 6 in the European Court of Human Rights

In this section we describe the ADF produced for Article 6, in addition to providing a brief summary of Article 6. The previous work by Aletras et al. [15] used 80 cases that relate to Article 6: throughout this paper we will use some of these cases as examples to highlight how we have developed and implemented our ADF representation of the domain.

Article 6 of the European Convention on Human Rights guarantees the right to a fair trial. The aim of the article is to guarantee the procedural rights of parties in civil proceedings and the rights of defendants in criminal proceedings. In essence the article is concerned with whether an applicant had ample opportunity to present their side of the case and contest any aspects they consider to be false, rather than ensuring that the courts have come to the correct decision. Provided the procedures followed were acceptable, the decision is acceptable with respect to Article 6.

When developing the ADF which represents Article 6, the *verdict* is whether there has been a violation of Article 6. The *issues* which inform whether there has been violation, these were determined from the legislation. There are three substantive issues that come from the legislation, and there are two additional procedural issues that need to be considered before examining the substantive issues. The three substantive issues, which have been heavily summarised, are:

- The case was fair and public
- The applicant was presumed innocent until proven guilty
- The applicant had their minimum rights respected

The two additional procedural issues are:

- The applicant bringing the case is the victim who was the discussed case
- The case is admissible

These five main issues are decided by considering *abstract factors*. These abstract factors may in turn have further abstract factors which decide their validity. In total our developed ADF has thirty-five abstract factors. For example when considering the third main issue, that the applicant had the minimum rights, there are six abstract factors which have been determined from the legislation and which decide that issue for the given case. These presence of these abstract factors themselves is decided by further abstract factors.

We then come to the bottom of the ADF where we reach the *base level factors*: these are answers to questions given by the user on the basis of the facts of the case.

A full example, where we follow a particular branch to its base level factor, is as follows:

Verdict There has been a violation of Article 6

This verdict relates, as stated above, to five issues, one of which is:

Issue The case is admissible

This issue is determined by two abstract factors:

Abstract Factor The case is well-founded
Abstract Factor There was no significant disadvantage

If either of these are not acceptable, the case will fail on the issue. That *there was no significant disadvantage* descends into further abstract factors, but that *the case is well founded* can be accepted on the basis of three base level factors, relating to the following three questions:

Base Level Factor Question Has the case been trivially answered previously?
Base Level Factor Question Does the applicant have evidence which supports the breach of Article 6?
Base Level Factor Question Is the case nonsensical?

If the answer is *yes* to the first two, and *no* to the third, then the case can be accepted as admissible and the substantive issues considered.

From an examination of the application document lodged by the claimant it is possible for a person familiar with such documents to determine that there is some evidence to back the claim, and that the claim is not nonsensical. The answers to these questions do require some familiarity with existing case law, but this can be expected from a lawyer using the system.

Table 1 contains a small subset of all the issues and factors that we have developed for the ADF. The complete ADF contains 51 nodes: 1 verdict, 5 issues, 10 abstract factors and 35 base level factors, which can be ascribed on the basis of an answer to a corresponding question. The questions relevant to the nodes in Table 1 are given in Table 2.

3.1. Implementation in Prolog

To implement the above ADF as an executable computational program, we followed a similar path to Al-Abdulkarim et al. [27] where each case is represented by a list of base level factors in a Prolog program. The program will print the status of each factor as it is determined, by resolving the ADF structure in the program.

The Prolog program follows the European Court by firstly checking if the case is admissible. Only once the case had been declared admissible, the program will traverse the full ADF and report the findings it produces. The code snippet below shows how node ID 1 from Table 1 has been developed in the Prolog code. The program resolves what the values of X, Y and Z are before checking the conditions as described in the table before printing human-readable output. Note that the identity tests are required to ensure that every node is visited and so can be included in the explanation.

Table 1. Subset of issues and factors in the ADF describing Article 6

ID & Factor	Children	Conditions
1 - Violation of Article 6	[2,3,8,20,21]	REJECT IF NOT is a victim OR NOT case is admissible ACCEPT IF the case was not fair or public OR victim was presumed guilty OR the victim did not have the minimum rights REJECT otherwise
2 - Is a victim		REJECT IF Q2 ACCEPT otherwise
3 - The case is admissible	[4,5]	REJECT IF NOT the case is well-founded OR there was no significant disadvantage ACCEPT otherwise
4 - The case is well founded		ACCEPT IF Q4a OR Q4b OR NOT Q4c REJECT otherwise
5 - No significant disadvantage	[6,7]	REJECT IF there is no fundamental reason why the case should be looked at OR there are domestic tribunals that have not looked at the case ACCEPT otherwise

Table 2. Subset of base level factors which answer the leaf node abstract base factors (LN)

Q	Base level factor question
2	Was the person bringing the case the victim?
4a	Has the case been trivially answered previously?
4b	Does the applicant have evidence which supports the breach of Article 6?
4c	Is the case nonsensical?

```
violationOfArticle6(CASE) :- (isVictim(CASE, X),
    isAdmissible(CASE, Y)), X == valid, Y == valid,
    fullcheck(CASE), !.
violationOfArticle6(_) :-
    write("The case is therefore inadmissible"), nl.

fullcheck(CASE) :- (isFairAndPublic(CASE, X),
    isPresumedInnocent(CASE, Y), hadMinimumRights(CASE, Z)),
    (X == valid, Y == valid, Z == valid),
    write("Therefore there is no violation of Article 6")
    , nl, !.
fullcheck(_) :-
    write("Therefore there is a violation of Article 6"), nl.
```

To resolve what the value of X is in the example, the program will continue checking conditions and printing output in order to give the value to X that has been requested.

For example *isFairAndPublic* returns valid when the conditions of the ADF have been met and returns invalid otherwise, which is shown clearly in the code snippet below.

```
isFairAndPublic(case(_,L), valid) :-
    (isConductedInAReasonableTime(case(_,L),X),
    isIndependantAndImpartial(case(_,L), Y),
    isConductedPublicly(case(_,L), Z),
    isEqualityOfArms(case(_,L), A),
    givenAccessToCourt(case(_,L), B)),
    (A == valid, B == valid, X == valid,
    Y == valid, Z == valid),
    write("The case was fair and public"),
    nl, !.
isFairAndPublic(case(_,_), invalid) :-
    write("The case was was not fair and public"), nl.
```

The program will continue traversing the ADF until it reaches factors that can be resolved by checking the answers to a base level factor question. For example Q2, answers whether the applicant is the victim in question. If the question should be answered positively in the case, then it is included in the list that is provided at the start of the program (*L* is the code fragments), as shown in the code snippet below.

```
isVictim(case(_,L), valid) :- member(Q2, L),
    write("The applicant is the victim"), nl.
isVictim(case(_,L), invalid) :- not(member(Q2,L)),
    write("The applicant is not the victim"), nl.
```

These three aspects (*fullCheck* for the issues, a procedure for each abstract factor, and the test against the list of base level factors) make up the entire Prolog program and when run will produce the list of factors that determine the decision of the program.

4. Experiments

For our implemented Prolog program, we have manually ascribed base level factors to 10 different cases to test different parts of the Prolog program. The cases chosen are all from the European Court of Human Rights and require resolution regarding whether there was a violation of Article 6.

Table 3. Cases used to test the Prolog program, along with highlights of the output produced by the Prolog program

Case	Actual outcome	Highlights of program output
MARGUŠ v. CROATIA	No violation	Therefore there is no violation of Article 6.
CARDOT v. FRANCE	Inadmissable	Not all domestic courts have been exhausted The applicant suffered a disadvantage The case is therefore inadmissible
ABDULLAYEV v. RUSSIA	Violation	Not given appropriate access to a court The case was not fair and public The applicant has not waived right to defend themselves Not prevented from accessing lawyers The applicant is defending themselves in person Therefore there is a violation of Article 6
ZARKOV v. SERBIA	Violation	The Government caused unreasonable delays The case was not conducted in a reasonable time Therefore there is a violation of Article 6
MOSER v. AUSTRIA	Violation	The case was not pronounced publicly The case was not conducted publicly The case was required to be conducted publicly, and was not There was not an equality of arms Therefore there is a violation of Article 6
CHAPMAN v. THE UNITED KINGDOM	No violation	Therefore there is no violation of Article 6
KHANUSTARANOV v. RUSSIA	Violation	Not given appropriate access to a court Therefore there is a violation of Article 6
STOILKOVSKA v. THE FORMER YUGOSLAV REPUBLIC OF MACEDONIA	Violation	The Government caused unreasonable delays The case was not conducted in a reasonable time Not given appropriate access to a court The case was not fair and public Therefore there is a violation of Article 6
UŽKURĖLIENĖ AND OTHERS v. LITHUANIA	No violation	Therefore there is no violation of Article 6
T.P. AND K.M. v. THE UNITED KINGDOM	No violation	Therefore there is no violation of Article 6

To illustrate how the cases have been used to test our Prolog program, we will walk through Moser v. Austria[4]. We have analysed each case and manually identified the factors that will be input into the program, answering each of the base level questions. For example, question 2 (Q2) holds as Moser was the victim in the case that is being debated. Q19 and Q20 do not hold as the verdict was not pronounced or conducted publicly. Q21 also does not hold as the victim was unable to comment on reports that were used by the Austrian courts. While a number of the questions are easily answered, such as Q2, answers to other questions, such as Q25, need to be derived from assumptions. For example for Q25 ("Was the victim informed of the crime in a language they understand?"), there is no discussion of this in the case facts, therefore we assume that the victim was told in a language they understand and Q25 should be considered true. The program, as with the European Court, assumes there was no violation unless there is specific discussion stating the reasoning why there is a violation. We believe that had the victim not been informed in an appropriate language, that would have appeared in the application.

Table 3 shows the results for all ten cases that we have chosen, each case has been chosen to test a specific aspect of the program. We have chosen nine cases that were used as part of the Article 6 dataset in Aletras et al. [15]. In addition we have added Cardot v. France[5] in order to test a case that is inadmissible. Note that Aletras et al. [15] used post trial documents and so did not include any inadmissible cases. We believe that it is important for an implementation to assess whether the case is admissible and so demonstrate this through our test set.

From the results we can see that the implementation of the ADF achieves correct results for all 10 test cases, and with the ability to explain why the violation was reported, or was not reported. Whilst our sample size for this initial experiment is small, the cases have been carefully selected to exercise different branches of the program and hence are extremely encouraging. We believe that when compared to the Aletras et al. [15] approach, there are a number of benefits to our approach which will help with finding acceptance among lawyers interested to use these programs to provide decision-support in case management, as has been shown through [11]. Next follows the output the program produces from Moser v. Austria. Issues are given in italics and the factors that led to the violation are indicated by "***". It is these factors which provide the highlights column in Table 3.

```
The applicant is the victim
   The case is well founded
      The case does examine a fundamental part of human rights act
      Domestic courts have been exhausted
   The applicant suffered no disadvantage
The case is therefore admissible
      The Government did not cause unreasonable delays
   The case was conducted in a reasonable time
      The Government was subjectively impartial
      The Government was objectively impartial
   The Government was independent and impartial
      Public hearing  would not prejudice outcome of case
```

[4] MOSER v. AUSTRIA JUDGMENT, 2006, European Court of Human Rights, (Application no. 12643/02)

[5] CARDOT v. FRANCE JUDGMENT, 1991, European Court of Human Rights, (Application no. 11069/84)

```
     Safety of the public would not be impacted by the case being-
     publicly pronounced
     Privacy is not required to deliver justice
   The public hearing would not hinder delivery of justice
     ***The case was not pronounced publicly
     ***The case was not conducted publicly
   ***The case was required to be conducted publicly, and was not
     **There was not an equality of arms
     Given appropriate access to a court
***The case was not fair and public
   The Government bore the burden of proof
   Any doubt benefited the applicant
The applicant was presumed innocent until guilty
     Was informed in the correct language
     Was promptly detailed circumstances to mount a reasonable-
     defence
     Applicant was told what crime they had committed
  The applicant was informed promptly in a language they understand
     Did have time or facilities to mound a reasonable defence
     The applicant has not attempted to escape trial
     The applicant has not waived right to defend themselves
     Not prevented from accessing lawyers
     The applicant is defending themselves in person
     Free access to legal assistance was available
  The applicant therefore had access to legal assistance
     Any witnesses were examined under the same different-
     conditions when compared to the Government
     Any witnesses that were not present had valid reasoning
  The applicant therefore was able to examine witnesses
  Had access to interpreter as required
The applicant had the minimum rights required
***Therefore there is a violation of Article 6
```

The biggest strength of the ADF is that it is able to explain through a series of statements the line(s) of reasoning which produces the outcome for a case, and so give a complete explanation in terms used in the domain. The output could be post-processed as in [21] to improve the quality of the presentation.

Although the series of test cases does suggest that the ADF can deduce case outcomes with an accuracy better than than the 79% produced by [15], future experiments on a larger test set are needed to further confirm this. The reason why our approach leads to better results is that it is based on an understanding of the domain, rather than the machine learning approach which creates learned probabilities on word groupings lacking full context, which is not how court cases are decided in practice.

Even if a program produced by machine learning was a perfect predictor of Article 6 cases, the program will over time become worse at predicting the outcomes due to how the law and its interpretation changes over time [17]. Programs that predict the outcomes of court cases must be able to adapt quickly to new information in order to capture the change in interpretation. Such changes are typically produced by landmark cases, which signal that updating is required. An ADF approach, such as the one developed, can be adapted quickly by changing the questions and factors, exploiting its modular nature. Adapting a machine learning program when a landmark case arises would be far more problematic. All of the past cases would be called into question: although many would be unaffected by the new ruling, some will now give a misleading picture of the law. Either all cases must be analysed to identify those rendered ineffective, which would remove many of the advantages of such systems, or there has to be time to acquire a substantial training set reflecting the new understanding of the law.

While there are several benefits to our approach, the major drawback is the time and expertise required to ascribe the factors that are fed into the program to describe a new case. We believe that this is the point at which learning from texts can produce benefits by answering the questions instantiating the base level factors, similar to the approaches by Ashley and Brüninghaus [28] and Branting et al. [29]. The aim of employing machine learning in this way is that we keep the benefits of good old-fashioned AI (through the ADF), which is needed in order to satisfy the lawyers who demand extremely high accuracy and explainability, while speeding up and reducing the expertise needed for processing new individual cases.

5. Conclusion

We have presented an argumentation-based representation of Article 6 of the European Convention on Human Rights, which is the right to a fair trial. The representation is an Abstract Dialectical Framework produced using the ANGELIC methodology, where the domain is represented as a tree, with the root node being a verdict, followed by children which are issues, whose children are abstract factors. The leaf nodes are base level factors, which are ascribed to a case by answering questions about the facts of the case. This framework was implemented in Prolog to enable reasoning about cases within the domain. Our framework and the Prolog program were tested with cases that concern Article 6 (and perhaps other articles) and were resolved in the ECHR.

Whilst the exercise in itself has been an instructive demonstration of the application of a computational model of argument to a real world domain, of further interest is that we were able to compare our approach to a machine learning approach to predicting cases in the exact same domain. Our success in producing a high level of accuracy in the performance of the program, being able to readily adapt the program as the law evolves, *and* being able to accompany this with strong explanatory features, addresses several limitations associated with machine learning approaches.

References

[1] Ashley KD. Case-based models of legal reasoning in a civil law context. In: International congress of comparative cultures and legal systems of the instituto de investigaciones jurídicas. Universidad Nacional Autonoma de México, Mexico City; 2004. p. 1–30.

[2] MacCormick DN, Summers RS, Goodhart AL. Interpreting precedents: a comparative study. Routledge; 2016.

[3] McCarty LT. Reflections on TAXMAN: An experiment in Artificial Intelligence and legal reasoning. Harvard Law Review. 1976;90:837.

[4] Rissland EL, Ashley KD. A case-based system for Trade Secrets law. In: Proceedings of the 1st International Conference on Artificial Intelligence and Law. ACM; 1987. p. 60–66.

[5] Bench-Capon T. HYPO's legacy: introduction to the virtual special issue. Artificial Intelligence and Law. 2017;25(2):1–46.

[6] Prakken H. A tool in modelling disagreement in law: preferring the most specific argument. In: Proceedings of the 3rd international conference on Artificial intelligence and law; 1991. p. 165–174.

[7] Brüninghaus S, Ashley KD. Predicting outcomes of case based legal arguments. In: Proceedings of the 9th International conference on Artificial Intelligence and Law. ACM; 2003. p. 233–242.

[8] Atkinson K, Bench-Capon T. Reasoning with Legal Cases: Analogy or Rule Application? In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 12–21.

[9] Prakken H. Modelling accrual of arguments in ASPIC+. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 103–112.

[10] Westermann H, Walker VR, Ashley KD, Benyekhlef K. Using Factors to Predict and Analyze Landlord-Tenant Decisions to Increase Access to Justice. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 133–142.

[11] Al-Abdulkarim L, Atkinson K, Bench-Capon T, Whittle S, Williams R, Wolfenden C. Noise induced hearing loss: Building an application using the ANGELIC methodology. Argument & Computation. 2019;10(1):5–22.

[12] Philipps L. A Neural Network to Identify Legal Precedents. In: Proc. of the 9th Symposium on Legal Data Processing in Europe. Council of Europe, Publishing and Documentation Service; 1089. p. 99–106.

[13] Bench-Capon T. Neural networks and open texture. In: Proceedings of the 4th international conference on Artificial intelligence and law; 1993. p. 292–297.

[14] Hunter D. Looking for law in all the wrong places: Legal theory and legal neural networks. In: Proceedings of JURIX 1994; 1994. p. 55–64.

[15] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.

[16] Vapnik V. The nature of statistical learning theory. Springer science & business media; 2013.

[17] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. Available on-line, 2019;p. 1–30.

[18] Bench-Capon T. The Need for Good Old fashioned AI and Law. In: International Trends in Legal Informatics: A Festschrift for Erich Schweighofer. Editions Weblaw, Bern; 2020. p. 23–36.

[19] Doshi-Velez F, Kortz M, Budish R, Bavitz C, Gershman S, O'Brien D, et al. Accountability of AI under the law: The role of explanation. arXiv preprint arXiv:171101134. 2017;.

[20] Al-Abdulkarim L, Atkinson K, Bench-Capon T. Accommodating change. Artificial Intelligence and Law. 2016;24(4):409–427.

[21] Al-Abdulkarim L, Atkinson K, Bench-Capon T. A methodology for designing systems to reason with legal cases using ADFs. Artificial Intelligence and Law. 2016;24(1):1–49.

[22] Atkinson K, Bench-Capon T, Routen T, Sánchez A, Whittle S, Williams R, et al. Realising ANGELIC Designs Using Logiak. In: Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference. vol. 322. IOS Press; 2019. p. 151.

[23] Aleven V. Teaching case-based argumentation through a model and examples. University of Pittsburgh; 1997.

[24] Brewka G, Woltran S. Abstract dialectical frameworks. In: Twelfth International Conference on the Principles of Knowledge Representation and Reasoning; 2010. p. 102–111.

[25] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93–124.

[26] Atkinson K, Bench-Capon T. Relating the ANGELIC Methodology and ASPIC+. In: COMMA; 2018. p. 109–116.

[27] Al-Abdulkarim L, Bench-Capon T, Atkinson K. Statement types in legal argument. In: Legal Knowledge and Information Systems: JURIX 2016: The 29th Annual Conference. vol. 294. IOS Press; 2016. p. 3.

[28] Ashley KD, Brüninghaus S. Automatically classifying case texts and predicting outcomes. Artificial Intelligence and Law. 2009;17(2):125–165.

[29] Branting K, Weiss B, Brown B, Pfeifer C, Chakraborty A, Ferro L, et al. Semi-Supervised Methods for Explainable Legal Prediction. In: Proc. of the 17th International Conf. on AI and Law; 2019. p. 22–31.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200489

Generating Adversarial Examples for Topic-Dependent Argument Classification

Tobias MAYER [a], Santiago MARRO [a], Elena CABRIO [a] and Serena VILLATA [a]
[a] *Université Côte d'Azur, CNRS, Inria, I3S, France* [1]

Abstract. In the last years, several empirical approaches have been proposed to tackle argument mining tasks, e.g., argument classification, relation prediction, argument synthesis. These approaches rely more and more on language models (e.g., BERT) to boost their performance. However, these language models require a lot of training data, and size is often a drawback of the available argument mining data sets. The goal of this paper is to assess the *robustness* of these language models for the argument classification task. More precisely, the aim of the current work is twofold: first, we generate adversarial examples addressing linguistic perturbations in the original sentences, and second, we improve the robustness of argument classification models using adversarial training. Two empirical evaluations are addressed relying on standard datasets for AM tasks, whilst the generated adversarial examples are qualitatively evaluated through a user study. Results prove the robustness of BERT for the argument classification task, yet highlighting that it is not invulnerable to simple linguistic perturbations in the input data.

Keywords. Argument Mining, Argument Classification, Robustness, Adversarial training

1. Introduction

Argument(ation) Mining (AM) [9,2,8] is the research area aiming at extracting and classifying argumentative structures from text. One subtask is topic-dependent argument classification, where the goal is to find relevant arguments for a given topic or claim from heterogeneous sources. This task is currently addressed by employing state-of-the-art deep learning methods, that recently benefit from pre-trained Language Models (LM) like BERT [3]. The idea underlying LM pre-training is to learn a task-independent understanding of natural language in an unsupervised fashion, from vast amounts of unlabeled text. After learning this general knowledge about a language, the model is then fine-tuned on tasks where the amount of available annotated data is significantly smaller, as it holds for AM annotated datasets. However, AM is a very context-dependent task and requires deep Natural Language Understanding (NLU), raising the research question: *How well does the pre-trained NLU scale in fine-tuned models for specific tasks such as argument classification?* In this paper, we answer this question by breaking it down into the following subquestions: *i) How vulnerable are argument classification models to adversarial*

[1] This work is partly funded by the French government labelled PIA program under its IDEX UCA JEDI project (ANR-15-IDEX-0001) and supported through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

attacks? and *ii) Can the robustness of argument classification models be improved with adversarial training?*
To answer these questions, we evaluate the efficiency of simple linguistic attacks against topic-dependent argument classification models based on LM pre-training. We generate eight different types of perturbations ranging from punctuation deletion to various word-based transformations, i.e. substitution or insertion, preserving the semantics of the sentence. The purpose of these attacks is to make the model more robust with adversarial training. The way we evaluate our approach to assess and improve the robustness of argument classification models is twofold: on the one side, we evaluate the success rate of each perturbation type on a model trained without any adversarial examples, and on the other side, we evaluate the improvement in performance on the original test set after augmenting the training data during adversarial training. For our experimental setting, we rely on two standard datasets in argument mining, namely the *UKP Sentential Argument Mining Corpus* [15], and the *IBM Debater: Evidence Sentences* corpus [14].
To summarize, the main contributions of this paper are the following:

- we propose different ways of creating linguistically simple perturbations and evaluate their impact on current state-of-the-art LM-based argument classification models, with respect to both in-domain and cross-topic performance;
- we address a user study to assess the quality of the generated perturbations;
- we empirically evaluate the effect of adversarial training for argument classification.

Obtained results highlight the effectiveness of adversarial training for argument classification. Furthermore, they point out the relatively robustness of LM that are nevertheless not invulnerable to simple changes to the input data. To the best of our knowledge, this is the first approach to generate natural language adversarial example for AM tasks.

In the following, Section 2 presents the related work. In Section 3, we discuss the methodology and background for adversarial attacks in NLP, and then we focus on adversarial training on the argument classification task. We detail our experimental setting[2], including the used datasets and the generated perturbations in Section 4, and we discuss the obtained results in Section 5. Concluding remarks and future work directions end the paper.

2. Related Work

Despite recent breakthroughs in modelling natural language understanding, the employed neural architectures still lack interpretability. They are black boxes for which it is hard to determine what they exactly learn or are receptive for. In this context, it was found that deep neural networks (DNN) are vulnerable to adversarial attacks; small changes to the input which fool the model into predicting a wrong label. Originally, crafting adversarial examples and attacking DNNs stems from the image processing domain [16,4,18]. Most of the employed methods there are gradient-based. These techniques cannot be easily adopted in the natural language processing domain. Images consist of pixels, which are represented as real value vectors: it is possible to slightly change the pixel values in a way which manipulates the gradients in a forward pass of a model to change the

[2] Code available at: `https://gitlab.com/tomaye/comma2020-adversarial_examples`

prediction, while the image is still perceived as unchanged to a human. On the other hand, modifying a sentence in a way that a human will not notice that change is almost impossible. The main problem here is that while pixel values are represented in a continuous space, words - that can also be represented in a continuous space in the form of real value vectors, i.e., embeddings, - per se are in a discrete space. Theoretically, one could find a vector in the embedding space which changes the prediction of a model, but constructing this vector from a discrete space of words is impossible in most of the cases. So, the recommended option is to create a perturbation on a linguistic level in the target sentence. But as said before, adding a word is most likely perceived by a human, contradicting the idea of an unnoticeable difference. Furthermore, adding even a single word might drastically change the semantics of a sentence. Given these two challenges, adversarial examples in the NLP domain need to be carefully designed. Due to the nature of the problem, only limited work on the perceivability has been done so far. The main work focuses on semantic preserving techniques accepting that the perturbation might be noticed by the human eye [20].

A strategy to generate adversarial examples are black-box approaches. Contrary to white-box approaches, they do not need any model specific knowledge except the input and output. Recent black-box approaches comprise methods concatenating, editing or substituting words in the input sentence [20]. There are also approaches which work on changing the underlying syntax by creating paraphrases [6]. We experimented with this automatic paraphrasing technique to generate adversarial examples. While this is a highly interesting topic, for the argument classification datasets the produced paraphrases were ungrammatical most of the time. So, we decided not to further pursue this kind of perturbation and exclude them from our experiments. An intuitive way of creating perturbations is to replace words with semantically similar alternatives, e.g., synonyms. Alzantot et al. [1] employ an approach where they replace each word of a sentence until the prediction changes. We do apply the same technique of replacing words with semantically similar alternatives, but with a different strategy: we only replace one word at a time minimizing the risk of producing a meaningless sentence. Moreover, we also add adverbs which change the semantics, strictly speaking, but do not change the label from argumentative to non-argumentative. Concerning the model we are attacking, previous work has shown that self-attentive models are more robust than recurrent architectures [5]. While in this work the authors used a white-box approach to precisely aim at weak points of the self-attending model, we went for a model independent black-box strategy. The generated adversarial examples lay the foundation to evaluate the robustness of argument classification models and to improve it with adversarial training.

3. Preliminaries

In this section, we introduce the terminology and give an overview of the methodology for adversarial attacks on deep neural networks (DNN) for NLP. We closely follow the definitions given in [20,18] and explain which setting we chose for the topic-dependent argument classification task.

Perturbation: A perturbation is a minor change to the test input example for the DNN. The goal is to change the prediction of the model, while the modification of the input example should not be perceived by humans. As previously mentioned, the notion of being imperceptible by humans is not as easily applicable to text, because most of the time a change in characters or even words is more obvious to human judgment than a slight adjustment to pixel values. Thus, for NLP the point of perceivability is rather interpreted as preserving the semantics of the original sentence with being still grammatical as a further constraint. Both of these constraints are challenging NLP tasks by themselves and have not been fully solved so far. As a consequence, automatically generated perturbations might violate these constraints raising the necessity for a human evaluation of the generated perturbations.

Granularity of Perturbation: The notion of granularity follows the thought above. While slight changes in single characters might not be that perceivable and preserve semantics as well as syntax, deleting, inserting or replacing words is a different level of perturbation. Even changes on sentence level are possible, e.g. paraphrasing or even adding whole sentences as it was done for attacking reading comprehension models [7]. For the argument classification task, the majority of our perturbations are on word level, since we wanted to evaluate the robustness of the targeted DNN language model against comparatively simple linguistic attacks.

Adversarial Example: An adversarial example x' is a perturbation of an input example x, where the modification indeed changes the prediction Y of the model, so that $y' \neq y$.

Attack Target: An adversarial attack can be targeted to change only specific labels in a multi-class classification setting. For argument classification, we do not see the necessity to specifically target the attacks against a certain label for two reasons: first, argument classification is usually limited to a two or three class classification problem, and second we do not want to make any assumptions about the architecture of the model we are attacking, leading us to the next point.

Model Knowledge: There are different strategies to generate adversarial examples depending on the availability of knowledge about the DNN the attacks are aimed at. White-box approach have access to all the information of the model, e.g. architecture, (hyper-) parameters, loss and activation function, training data, or confidence scores. On the contrary, the black-box approaches have only access to the input and output of a model [11]. We selected a specific model to attack, i.e. BERT, but since there are and will be other self-attending architectures based on language model pre-training, we do not want our perturbations to be limited to only BERT and decided to go for a black-box approach ignoring valuable information like the attention scores.

Adversarial Training: Currently, the only defense strategy against adversarial attacks is adversarial training where the DNN is re-trained with adversarial examples [20,16]. One strategy is also to include inputs which are unlikely to occur naturally. This defense strategy aims at reducing the *"fundamental blind spots"* [4] of a model making the model more robust against divers input. With respect to NLP and specifically to argument classification, this means that including ungrammatical examples in training the model is justified. After all, argument classification is based on representations of full sentences, which are created from word level representations independent of the grammaticality of the sentence.

Evaluation Metric: The evaluation of adversarial attacks can be measured by the degree it decreases the performance of a DNN. We decided to not do that, because we cannot ensure the same number of generated perturbations per input example and thus might bias the results. Another prominent way to evaluate the perturbation efficiency is the success rate. This is the percentage of adversarial examples over the number of generated perturbations.

Robustness: In our terminology, robustness refers to the ability of a model to correctly classify unseen test data from the same domain as the training data. Contrary to that, we refer to generalizability as the concept of being able to exploit the already acquired knowledge in a new domain. For argument classification, this means that when training and test set talk about the same topics, e.g. *abortion*, adversarial attacks are testing robustness. For the case when the test set contains topics which are never seen during training, we talk about (cross-topic) generalizability of a model. Our main goal with adversarial training is to increase the robustness of a model, not its generalizability.

4. Experiment Setup

This section describes *i)* the datasets used for training and testing and the attacked DNN, *ii)* the different types of generated perturbations, and *iii)* a qualitative evaluation of the perturbations through a user study.

4.1. Data and Target Model

As previously mentioned, the application domain for the adversarial attacks in our work is topic-dependent argument classification. For this task, there are two major corpora available: 1) The *UKP Sentential Argument Mining Corpus* [15], which is a collection of 25,492 sentences annotated as an *ArgumentFor (**Arg+**), ArgumentAgainst (**Arg-**)* or *NoArgument (**NoArg**)* to a specific topic. The corpus comprises 8 different topics, i.e. *abortion, cloning, death penalty, gun control, marijuana legalization, minimum wage, nuclear energy* and *school uniforms*, and 2) the *IBM Debater: Evidence Sentences* [14], which is a collection of sentences from online debate portals annotated with *evidence (**Arg**)* or *no evidence (**NoArg**)* in regard to one of the 118 topics. Following existing experimental setups from the literature [14,13], the training set comprises 83 topics (4,065 sentences) and the test set 35 (1,718 sentences).

Self-attentive transformer models like BERT [3], which use LM pre-training, have become a mighty tool for many NLP tasks. This also applies to argument mining. Following recent state-of-the-art on topic-dependent argument classification [13], we evaluate the adversarial attacks on the BERT base model. The input for BERT consists of the input sentence concatenated with the topic. As introduced before, our perturbations are black-box methods not taking advantage of model specific knowledge, e.g. attention score. Thus, they can be easily transferred to other architectures in the future.

We conducted two lines of experiments. The first one to test the success rate of the perturbations, and the second one to evaluate adversarial training. For both lines, training and performance evaluation were based on the code provided by Reimers et al. [13]. Hyper-parameters for fine-tuning the models were also replicated without any changes. The only difference is that we do not split the training data into a development set, since we are not tuning any parameters. For both lines of experiments, there are three different scenarios: 1) a model were the train (80%) and test (20%) sets comprise all eight topics of the UKP corpus (**UKP all**); 2) the leave-one-out training (**UKP x-topic**), where seven topics of the UKP corpus were used for training and the eighth is used for testing. In total, this results in eight different models. The results in this scenario are reported as the average over the eight models; 3) in the last scenario, a model is trained on the IBM corpus with the train-test split described above (**IBM x-topic**).

For the first line of experiments, i.e., perturbation evaluation, the success rate of a perturbation is evaluated on a model trained without any adversarial examples. Only perturbations from the test set are considered in calculating the success rate. For each perturbation, we computed a label-wise success rate. For the second line of experiments, i.e., adversarial training, only perturbations of the training set are considered for augmenting the training data. We re-trained every model under the same conditions as before, but with the only difference being the augmented training data. The evaluation of an adversarially trained model is done on the same unmodified test set as the normally trained counterpart to guarantee comparability.

4.2. Perturbation Types

In the following, we introduce the eight different methods we used to generate perturbations for given input examples. The perturbation generation methods are based on word or token types. Hence, the number of generated perturbations per input example varies. To give an idea of the order of magnitude, we report the average number of generated perturbations for each test set of the two corpora.

Named Entities (NE) The first method we propose consists of replacing a named entity in the input sentence. To achieve this, we constructed a list of named entities for each of the four standard categories, i.e., *PER*, *LOC*, *ORG*, *MISC*, present in the CoNLL 2003 Shared Task dataset for named entity recognition [17]. Using this list, we then generate for each NE present in the original sentence one new perturbation replacing the entity with a different entity from the same category. In order to preserve the semantics, we used pre-trained word embeddings (fastText) as a means of distance, and selected the closest neighbours. If the original input sentence does not contain a NE, no perturbations are generated. Accordingly, the average number of generated perturbations per input sentence varies. On the UKP dataset we produced an average of 3.11 perturbations per sentence. The IBM dataset contains more NEs per sentence, therefore the produced number of perturbations per example is higher, namely 10.15.

Example 4.1 *Original sentence: According to **FBI** statistics, 46,313 Americans were murdered with firearms during the time period of 2007 to 2011.*
*Adversarial attack: According to **U.S. Bureau of Investigation** statistics, 46,313 Americans were murdered with firearms during the time period of 2007 to 2011.*

Adjectives This method is similar to the list-based attack proposed in [1], where words in the input sentence are replaced with a word from a list of semantically similar words. Contrary to the aforementioned work, we only replace one word per perturbation. Specifically, we exchange adjectives with their synonyms, e.g. *big* with *large*, producing one perturbation example for each adjective in the sentence. The synonyms were taken from the WordNet interface in the NLTK. For the UKP dataset, we have an average of 2.12 adjectives per sentence, while for the IBM dataset we generate 2.9 perturbations per sentence.

Punctuation This is the only modification of a sentence on character-level. Here, all the punctuation, e.g., "." or ",", is removed from the original input sentence. Naturally, this method provides one perturbation per sentence.

Scalar Adverbs This method is about adding or replacing emphasising modal adverbs, such as *considerably*, or trigger words for scalar implicature, such as *comparatively* or *largely*. They are added before a verb or an adjective. As will be shown in succeeding sections, the positioning algorithm needs to be improved, since some adverbs should be placed only after the word, while others should be placed only before the word or can take both positions. The average amount of perturbations generated per input sentences is around 3.94 for the UKP dataset and 4.67 for the IBM one.

Example 4.2 Original sentence: *It is possible to fuel nuclear power plants with other fuel types than uranium.*
Adversarial attack: *It is **totally** possible to fuel nuclear power plants with other fuel types than uranium.*

Nouns Similar to the adjectives method we proposed, this list-based attack exchanges a noun with its hyponym. Again, we only replace one word per perturbation producing one perturbation example for each noun in the sentence. This method generated an average of 12.19 perturbations per sentence on the UKP dataset, whilst the number increases to 17 for the IBM dataset.

Example 4.3 Original sentence: *When it comes to infertile couples, should not they be granted the **opportunity** to produce clones of themselves?*
Adversarial attack: *When it comes to infertile couples, should not they be granted the **chance** to produce clones of themselves?*

Conjunctions This method consists of adding adverbial conjunctions, such as *furthermore* or *nonetheless*, at the beginning of the input sentence. If the sentence already begins with an adverbial conjunction, the sentence is skipped. This attack delivers an average of 2.69 perturbations per sentence on the UKP dataset and 2.88 on the IBM.

Speculative Adverbs They are modal adverbs related to the possibility property of verbs. This method is similar to the aforementioned scalar adverbs perturbation. Another list-based attack where modal adverbs related to the possibility property of verbs, such as *certainly*, are added directly before a verb. In this case, we obtained an average of 1.67 perturbations per sentence on the UKP dataset and 1.75 on the IBM.

Example 4.4 Original sentence: *Even the gateway effect — the theory that cannabis leads to other drugs — was discarded long ago.*
Adversarial attack: *Even the gateway effect — the theory that cannabis **indeed** leads to other drugs — was discarded long ago.*

Topic Alternatives Previous work has shown that including the topic in the BERT input increases the performance of the model [13]. Thus, exchanging the topic with alternatives is a relevant perturbation to evaluate. For each topic in the two corpora, we created a list of alternatives. For example, *arms limitation* for *gun control* or *capital punishment* for *death penalty*. While we created an average of 4.25 alternatives per topic for UKP dataset, for the IBM dataset on average, there were 2.75 alternatives per topic.

4.3. User Study: Quality of Generated Perturbations

As an additional evaluation criteria of the generated perturbations, we conducted a user study about the preservation of semantics between the original sentence and the sentence after the modification. Both versions of a sentence were presented to the user and the user was asked if the two sentences 1) have the same meaning, 2) do not share the same meaning, or 3) if the transformed sentence is not meaningful, where "not meaningful" could mean either that the sentence has become ungrammatical or that it does not make sense anymore. For each answer option there was also a text field giving the possibility to voluntarily provide a justification of their decision. In total, 72 pairs of sentences were presented to each participant comprising every type of perturbation, but the topic alternative and punctuation deletion. We excluded the topic alternatives from the study, because the topic is an independent part of the model input and does not modify the grammaticality or semantics of a sentence. Same holds for the deletion of punctuation, which only changes the semantics of a sentence in some rare case of rhetorical questions. Moreover, the participant thinking of proper punctuation might have shifted their focus from the actual task, i.e. semantic similarity. The sentence length of each pair of sentences was controlled to have a difference of maximum one standard deviation from the mean sentence length of the sentences in the dataset. Participants in the user study were mainly non-native speakers with a higher educational degree (Masters degrees or Ph.D.) and a fluent level of English. In total, 31 people completed the questionnaire.

The perturbation method with the highest percentage of preserving the meaning of the sentence, i.e. 93.68%, is adding conjunctive adverbs. Naturally, this barely impacts the meaning of a single sentence. For the NE replacement, 71.3% of the people found the exchange as meaningful. The main criticism was that the new named entity, especially when they were acronyms, was unknown to the participant. Overall, employing word embeddings as a distance criteria to select NEs of the same type preserves the meaningfulness in most cases. Replacing an adjective with its synonym was in 61.04% of the cases found to be meaningful. While for the other cases, it was reported that the selected synonym was not suitable for the given context. Similar feedback was gathered for the hyponym replacement of nouns. Here, in 52.53% of the cases the selected noun did not fit the context, as either being too specific or unrelated to the topic. Inserting speculative adverbs was perceived as not changing the meaning of a sentence in 57.82% of the cases. A main observation reported by the participants is the change in credibility or certainty of the mentioned studies and other evidence, e.g. changing facts to opinions. Indeed, this does change the semantics of a sentence, but with respect to an argument classifier the uncertainty of an evidence does not matter as much as that it is correctly detected as being an argument. From this point of view, despite the study results, we consider this perturbation method a valid and meaningful transformation. Compared with the other perturbation types, adding and replacing scalar adverbs caused with 57.33% the most cases of changes of a meaning of a sentence. The participants found that this transformation often breaks the grammaticality of a sentence. A future challenge is to find the right place to insert such adverbs, because some of them can either precede the target word or come only after it. Moreover, one has to consider if a target word can scale. For example, *genetic, mandatory* or *guilty* cannot be compared. There is no such thing as *fairly mandatory*. These points need to be address in future work.

5. Results and Discussion

In this section, we present and discuss the results of our two lines of experiments. First, the success rates for each perturbation type, and second, the adversarial training.

5.1. Adversarial Attacks

Table 1 reports on the success rate (the percentage) of adversarial examples over the total of generated perturbations.

Perturbation Type	**UKP all**			**UKP x-topic**			**IBM x-topic**	
	Arg+	Arg-	NoArg	Arg+	Arg-	NoArg	Arg	NoArg
Named Entities	7.06	7.30	2.02	6.14	7.22	2.30	1.51	0.18
Adjectives	10.90	10.02	6.70	12.16	10.37	5.89	3.79	0.03
Punctuation	8.86	9.74	4.21	10.41	10.61	4.34	2.78	0.19
Scalar Adverbs	5.87	7.15	3.41	7.39	7.57	3.29	2.01	0.08
Nouns	13.91	14.56	7.35	15.08	14.65	7.6	8.43	0.53
Spec. Adverbs	6.31	6.89	2.99	7.49	6.82	2.53	1.42	0.06
Conjunctions	5.87	7.29	4.33	9.66	9.52	4.56	3.64	0.4
Topic Alternatives	0.81	1.33	0.29	1.07	1.13	0.41	1.14	0.08

Table 1. Label-wise success rate of each perturbation type on the different test scenarios.

Looking at the in-domain test scenario, i.e., UKP all, one can observe that the Arg- label is more affected by the attacks than the Arg+ label, with exception of the adjectives. The adjective and noun replacement have the highest success rates in attacking the models. For adjectives, this could be explained with the fact that they usually carry sentiments whose perception might differ if they appear in a pro or con argument. For nouns, the replacement with hyponyms has the highest success rate, but given that in the human evaluation only in 47.47% of the cases the perturbation was perceived as meaningful, we cannot consider results with respect to this perturbation as fully reliable.

Overall, the positive classes, Arg+, Arg- and Arg, showed to be more vulnerable to attacks than the no argument class. Usually, the structure of the task at hand, which features in the data one tries to learn, is associated with the positive class. Meaning that the complementary class does not necessarily contain a distinctive pattern in the feature space, because it contains everything which is not wanted. Hence, it cannot be as efficiently attacked as the learnt patterns for the positive classes. Unexpectedly, deleting the punctuation resulted in a comparatively high success rate. After reviewing the attention scores of the model, we found that, contrary to our expectations, the model tends to attend to punctuation. This observation needs to be confirmed at a larger scale, though. Exchanging the topic with alternative wording resulted in an insignificant success rate not affecting the model. Concerning the cross topic evaluation, the UKP x-topic shows partially higher vulnerability than its in-domain counterpart. Since cross domain is the harder task, the confidence scores are lower for unseen test data, and with that the overall performance compared to in-domain models. A less confident model is easier to attack, explaining the higher success rates. Interestingly, the IBM x-topic is not as vulnerable to attacks as the UKP x-topic model. Again, as can be noticed in Table 2, the overall performance of the IBM model is higher. Since in both cases the same model architecture is employed, the only difference is the data. The IBM dataset seems to be more structurally uniform than the UKP dataset, explaining why test performance is higher and the success rate of attacks lower. Another point supporting this is that the exchange of NEs, which the IBM corpus contains more per sentence than the UKP one, barely changes the classification of an input example. This connotes that, in the case of the IBM data, NEs are not as important for the model justifying that they can be exchanged without losing the argumentative function of a sentence. Even though this further justifies our named entity perturbation method, it is ineffective in this case. Overall, BERT-based topic-dependent argument classification models are relatively robust against minor changes to the input, but still vulnerable to a certain degree. In roughly 5-10% of the cases, adding a meaning preserving word changes the prediction of the model.

5.2. Adversarial Training

The most common strategy to defend from adversarial attacks and make a model more robust is adversarial training. This is covered in our second line of experiments, whose results are reported in Table 2.

	UKP all	UKP x-topic	IBM x-topic
standard training	73.70	60.9	77.58
adversarial training	80.22	59.3	78.57

Table 2. Results in macro f_1 for models with and without adversarial training.

For the in-domain scenario (UKP all), one can observe an increase of 6.5 points in f_1-score compared to the model trained without adversarial examples. This shows that adding linguistic variants of the training data helps in predicting unseen test data from the same domain. Intuitively this makes sense, arguments are often rephrased differently or are re-used as targets for undercutting, for example. With respect to BERT, this raises questions. In the aforementioned experiments on perturbation efficiency, we have seen that BERT seems to be quite robust against our adversarial attacks. Also, in previous works, models based on language model pre-training advanced the state-of-the-art, which was said to be due to the natural language understanding capabilities learnt during pre-training. Accordingly, this should mean that slight variations of the input are covered by the language model. The increase in performance with adversarial training shows that this supposed NLU capability is either not fully utilized or blurred during fine-tuning, or was limited in the first place. We assume it is a mixture of both, since other experiments in different domains show that BERT-like models are more robust than recurrent networks [5], but also that the language modelling capabilities of self-attentive models are limited [12,19]. Even if the success rates of our perturbations are only between 5-10%, added up these make quite a number of examples, which BERT is vulnerable to. Adding these linguistic variations to the training data, though, boosts the NLU capabilities making the model more receptive for them. Note that this way the training data is increased by roughly a factor of twenty. This indeed shows that adversarial training helps in-domain predictions and improves the robustness of a model, as intended. Table 3 shows examples where adversarial training corrected the model prediction.

A justified doubt coming up here is the question of overfitting. *Did the adversarial training really help in NLU or did it just improve learning the dataset?* In the latter case,

topic	**sentence**	*pred*$_1$	*pred*$_2$
gun control	Five women are murdered with guns every day in the United States.	NoArg	Arg+
school uniforms	Up to now , this uniform is still in use , making it the ‘ oldest uniform in history. ’	Arg+	NoArg
cloning	I find this reasoning absolutely ridiculous, since a person is a person despite their genetic source or if artificially created.	Arg-	Arg+

Table 3. Examples were adversarial training improved the model prediction. *pred*$_1$ model prediction before adversarial training, *pred*$_2$ model prediction after adversarial training, which is also the true label.

one would see a decline in cross domain evaluation, because the model is overly focused on in-domain specific features. As can be seen in Table 2, the cross domain performance is not dropping significantly with adversarial training. Both models are still in an acceptably similar range compared with their normally trained counterpart. The UKP x-topic losses 1.6 f_1-score, while the IBM model even shows a slight increases of roughly 1 f_1-score. Meaning that the generalizability of the models is preserved, ergo they did not overfit on the training domain. So *why is it that adversarial training helps in-domain, but does not improve the cross domain performance?* At this point, we like to repeat the aforementioned distinction between robustness and generalizability. For us, robustness is more related to the ability to understand language in the sense of linguistic flexibility; being able to understand differently worded phrases about the same thing. Generalizability, on the other hand, is the ability of a model to transfer and apply already learnt patterns to a new domain. In our case, an increase in performance for the models tested on cross topics is related to the generalizability. While depending on the task of the application field, generalizability and robustness have a strong overlap, we think, one has to carefully distinguish them for argument mining. Usually, cross domain in AM means that the model should be able to detect arguments for a topic unseen during training. Assuming the new topic is not somehow related to the topics seen during training, this means, the model has to infer everything associated with a given input sentence and decide if this can be an argument related to the topic or not. The problem is one can only conditionally infer new arguments from existing arguments in the semantic space. If the two arguments are structurally similar to a certain degree (or use similar key components), it is possible. But finding new arguments for an unseen domain is beyond language modelling. It requires also a deep understanding of knowledge and common sense. Especially the latter two cannot be efficiently learnt from word co-occurrences alone [19,10]. As a result, it is not surprising that augmenting training data with alternative wording of the data does not improve generalizability. After all, the examples added for adversarial training are mostly noise with respect to the new unseen test domain; noise, which is not negatively affecting the generalizability of the BERT model.

6. Conclusion

This paper presents the first approach to test the robustness of argument classification models through adversarial examples. We investigate different ways to produce meaningful adversarial examples, and we assess their quality through a user study. Furthermore, we demonstrate the effectiveness of adversarial training and we empirically show that it helps to improve robustness without impacting generalizability. Obtained results highlight that BERT is robust but still vulnerable to simple changes to the input.

For the future, a further evaluation of the robustness of argument classification models is needed. This goes beyond the weaknesses of the here presented approach, such as controlling the selection of synonyms and hyponyms or the positioning and selection algorithm for adverbs. Combinations of different perturbation types are worth exploring. As well as white-box approaches [5], where the target words are carefully selected dependent on model parameters. Another highly interesting and relevant field is the evaluation of paraphrases as a means to attack models. As a more general goal, experiments are required to find the right balance between augmenting the training data with adversarial examples and noise for efficient adversarial training.

References

[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *Proc. of EMNLP 2018*, pages 2890–2896, 2018.

[2] E. Cabrio and S. Villata. Five years of argument mining: a data-driven analysis. In *Proc. of IJCAI 2018*, pages 5427–5433, 2018.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT 2019*, pages 4171–4186, 2019.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR 2015*, 2015.

[5] Y.-L. Hsieh, M. Cheng, D.-C. Juan, W. Wei, W.-L. Hsu, and C.-J. Hsieh. On the robustness of self-attentive models. In *Proc. of ACL 2019*, pages 1520–1529, 2019.

[6] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proc. of NAACL 2018*, pages 1875–1885, 2018.

[7] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proc. of EMNLP 2017*, pages 2021–2031, 2017.

[8] J. Lawrence and C. Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2019.

[9] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Techn.*, 16(2):10, 2016.

[10] T. Mayer. Enriching language models with semantics. In *Proc. of ECAI 2020*, 2020.

[11] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proc. of ACM AsiaCCS 2017*, pages 506–519, 2017.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

[13] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych. Classification and clustering of arguments with contextualized word embeddings. In *Proc. of ACL 2019*, pages 567–578, 2019.

[14] E. Shnarch, C. Alzate, L. Dankin, M. Gleize, Y. Hou, L. Choshen, R. Aharonov, and N. Slonim. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proc. of ACL 2018*, pages 599–605, 2018.

[15] C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych. Cross-topic argument mining from heterogeneous sources. In *Proc. of EMNLP 2018*, pages 3664–3674, 2018.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. of ICLR 2014*, 2014.

[17] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *CoRR*, 2003.

[18] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *CoRR*, 2017.

[19] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL 2019*, pages 4791–4800, 2019.

[20] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200490

Dataset Independent Baselines for Relation Prediction in Argument Mining

Oana COCARASCU [a,c], Elena CABRIO [b], Serena VILLATA [b], Francesca TONI [a]
[a] *Imperial College London, UK*
[b] *Université Côte d'Azur, CNRS, Inria, I3S, France*
[c] *King's College London, UK*

Abstract. Argument(ation) Mining (AM) is the research area which aims at extracting argument components and predicting argumentative relations (i.e., *support* and *attack*) from text. In particular, numerous approaches have been proposed in the literature to predict the relations holding between arguments, and application-specific annotated resources were built for this purpose. Despite the fact that these resources were created to experiment on the same task, the definition of a single relation prediction method to be successfully applied to a significant portion of these datasets is an open research problem in AM. This means that none of the methods proposed in the literature can be easily ported from one resource to another. In this paper, we address this problem by proposing a set of dataset independent strong neural baselines which obtain homogeneous results on all the datasets proposed in the literature for the argumentative relation prediction task in AM. Thus, our baselines can be employed by the AM community to compare more effectively how well a method performs on the argumentative relation prediction task.

Keywords. Argument Mining, Relation Prediction, Machine Learning Methods

1. Introduction

Argument(ation) Mining (AM) is "the general task of analyzing discourse on the pragmatics level and applying a certain argumentation theory to model and automatically analyze the data at hand" [16]. Two tasks are crucial in AM [22,6,20]: *1)* argument component detection within the input natural language text, aiming at the identification of the textual boundaries of the arguments and their classification (claim, premise); and *2)* relation prediction, aiming at identifying (support, attack) relations between argumentative components, possibly identified in the first stage. In this paper we focus on the second task. Despite the high volume of approaches tackling the relation prediction task with satisfying results (see [6] for an extensive list), a problem arises: these solutions heavily rely on the peculiar features of the dataset taken into account for the experimental setting and are hardly portable from one application domain to another. On the one side, this issue can be explained by the huge number of heterogeneous application domains where argumentative text may be analysed (e.g., online reviews, blogs, political debates, legal cases). On the other side, it represents a drawback for the comparison of the different approaches proposed in the literature, which are often presented as solutions addressing the relation prediction task from a dataset independent point of view. A side drawback for

	essays	micro	nk	db	ibm	com	web	cdcp	ukp	aif
# attacks	497	108	378	141	1069	296	1301	0	5935	9854
# supports	4841	263	353	179	1325	462	1329	1220	4759	7543

Table 1. Datasets' statistics.

the AM community is therefore a lack of large annotated resources for this task, as most available resources cannot be successfully reused, being highly context-based. Even the employment of pretrained language models (e.g., BERT [12]) does not address this issue.

In this paper, we tackle this issue by proposing a set of strong cross-dataset baselines based on different neural architectures. Our baselines are shown to perform homogeneously over all the datasets proposed in the literature for the relation prediction task in AM, differently from individual methods proposed in the literature. Our contribution is to bestow the AM community with a set of strong cross-dataset baselines to compare with in order to demonstrate how well a relation prediction method for AM performs.

We focus on two types of argumentative relations: *attack* and *support*, given that the majority of datasets target only these two types of relations. We define neural baselines to address the corresponding binary classification problem, analysing, to the best of our knowledge, all available datasets for this task, ranging from persuasive essays to user-generated content, to political speeches. Given two arguments, we are interested in determining the argumentative relation between the first, called *child* argument, and the second, called *parent* argument, using a neural model. For example, the child argument *People know video game violence is fake* may attack the parent argument *Youth playing violent games exhibit more aggression*. In our baselines, each of the two arguments is represented using embeddings as well as other features. We propose three neural network architectures for the classification task, two concerned with the way child and parent are passed through the network (*concat* model and *mix* model), and an attention-based model. We also explore BERT as an alternative to our baselines: although this is used successfully to boost performances for other tasks in Natural Language Processing, it is generally not competitive for relation prediction with the datasets we consider.

We conduct experiments with a number of datasets, chosen either because they were specially created for relation prediction in AM or because they can be easily transformed to be used for this task. These are: Essays (essay) [33], Microtexts (micro) [29], Nixon-Kennedy (nk) [23], Debatepedia (db) [5], IBM (ibm) [1], ComArg (com) [3], Web-content (web) [7], CDCP (cdcp) [28], UKP (ukp) [34], AIFdb (aif) [2,10,18,31]. Datasets' statistics can be found in Table 1[1].

2. Neural baselines for relation prediction

We use four types of features: word embeddings, sentiment features, syntactic features, computed for both child and parent, and textual entailment from child to parent. We refer to the last three types of features as *standard* features. Word embeddings are distributed representations of texts in an n-dimensional space. Textual entailment represents the class (amongst entailment, contradiction, or neutral) obtained using AllenNLP[2], a textual entailment model based on decomposable attention [27]. The features related to

[1] For more details about the individual datasets, we refer the reader to the relevant publications.

[2] `https://allennlp.org`

sentiment are based on manipulation of SentiWordNet [15] and the sentiment of the entire (child and parent) texts analysed using the VADER sentiment analyser [17]. Every WordNet synset [24] can be associated to three scores describing how objective, positive, and negative it is. For every word in the (child and parent) texts, we select the first synset and compute its positive score and its negative score. In summary, the features related to sentiment for a text t that consists of n words, $W_i = 1 \ldots w_n$, are the following: (i) sentiment score ($\sum_{w_i} pos\_score(w_i) - neg\_score(w_i)$), (ii) number of positive/negative/neutral words in t, (iii) sentiment polarity class and score of t. Syntactic features consist of text statistics (e.g., number of words) and word statistics with respect to part-of-speech tags (i.e., number of words, nouns, verbs, first person singular, etc.) and lexical diversity (i.e., number of unique words divided by the total number of words in text t).

We describe the three neural architectures we propose for determining the argumentative relation (of attack or support) holding between child and parent. For all, we report only configurations of the architectures and number/size of the hidden layers which performed the best[3]. For our models, we use GRUs [11] as they take less time to train and are more efficient.

Concat model (C). In this model, each of the child and parent embeddings is passed through a GRU. We concatenate the standard features of the child and of the parent. The merged standard vector is then concatenated with the outputs of the GRUs. The resulting vector is passed through 2 dense layers (of 256 neurons and 64 neurons, with sigmoid as activation function), and then to softmax to determine the argumentative relation.

Mix model (M). In this model, we first concatenate the child and parent embeddings and then pass them through a GRU, differently from the concat model where we pass each embedding vector through a GRU first. We concatenate the standard features that we obtain for the child and for the parent. The merged standard vector is then concatenated with the output of the GRU. From this stage, the network resembles the concat model: the resulting vector is passed through 2 dense layers (of 256 neurons and 64 neurons, with sigmoid as activation function), to be then finally passed to softmax.

Attention model (A). Inspired by the demonstrated effectiveness of attention-based models [36,35], we combine the GRU-based model with attention mechanisms. Each of the child and parent embeddings is passed through a GRU and we compute attention in two directions. We concatenate the standard features of the child and of the parent. The merged standard vector is then concatenated with the outputs of the GRUs. The resulting vector is passed through a single dense layers (128 neurons, with sigmoid as activation function), that is then passed to softmax.

3. Experimental results

Non-neural baselines. For training we have used the larger datasets, *aif*, *essay*, *ibm* and *web*. We resampled the minority class from the *essay* dataset and used our models on the oversampled dataset. We did not use for training the *ukp* dataset as the parent is a topic instead of an argument. The models were then tested on the remaining datasets, with the average being computed on testing datasets. We report the F_1 performance of

[3] We also experimented with 1 and 2 hidden layers, and hidden layer sizes of 32, 64, 128, and 256, trying all possible combinations towards best configurations. We did not consider a higher number of hidden layers due to the small size of the data.

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
non-neural baselines	RF	F_1 A	0.32	0.24	0.40		0.33	0.38	-	0.39	0.55	0.44	0.381	0.508
		F_1 S	0.57	0.74	0.64		0.67	0.59	0.85	0.53	0.59	0.54	0.636	
	RF	F_1 A	0.57	0.40	0.45	0.53	0.43		-	0.60	0.52	0.57	0.509	0.490
		F_1 S	0.44	0.47	0.52	0.41	0.57		0.51	0.45	0.50	0.38	0.472	
	SVM	F_1 A		0.34	0.36	0.33	0.29	0.38	-	0.42	0.42	0.40	0.368	0.503
		F_1 S		0.71	0.67	0.65	0.67	0.59	0.84	0.57	0.56	0.49	0.639	
	SVM	F_1 A	0.49	0.35	0.39	0.39	0.38		-	0.56	0.57	0.520	0.456	0.498
		F_1 S	0.50	0.54	0.52	0.59	0.60		0.67	0.46	0.47	0.500	0.539	

Table 2. Experimental results for non-neural baselines with F_1 for **A**ttack and for **S**upport. The blanks indicate the training dataset. The Average (Avg) and the Macro (Macr) Avg exclude the results for the training datasets.

the *attack* class (A) and the *support* class (S) for the non-neural baselines in Table 2. We used Random Forests (RF) [4] with 15 trees in the forest and gini impurity criterion and SVM with linear kernel using LIBSVM [9], obtained as a result of performing a grid search, as it is the most commonly used algorithm in the works that experiment on the datasets we considered [1,3,8,23,25]. On top of the *standard* features used for our neural models, for the baselines we added the following features: TF-IDF, number of common nouns, verbs and adjectives between the two texts as in [23], a different sentiment score $\frac{\text{nr\_pos}-\text{nr\_neg}}{\text{nr\_pos}+\text{nr\_neg}+1}$ as in [1], with all features being normalized.

Neural baselines with non-contextualised word embeddings. Table 3 shows the best baselines for relation prediction in AM. We experimented with GloVe (300-dimensional) embeddings [30], using pre-trained word representations in all our models. We used 100 as the sequence size as we noticed that there are few instances with more than 100 words. We used a batch size of 32 and trained for 10 epochs (as a higher number of epochs led to overfitting). We report the results using embeddings and *syntactic* features and the results with *all* the features presented in Section 2. We also conducted a feature ablation experiment (with embeddings being always used) and observed that syntactic features contribute the most to performance, with the other types of features bringing small improvements when used together only with embeddings. In addition, we have run experiments using two datasets for training to test whether combining two datasets improves performance. During training, we used one of the large datasets (*aif*, *essay*, *ibm*, *web*) and one of the remaining datasets (represented as blanks in the table).

Amongst the proposed architectures, the attention model generally performs better. Using only a single dataset for training, the model that performs the best is the *mix* model using *all* features and trained on the *essay* dataset. The best results are obtained when using another dataset along one of the larger datasets for training. This is because combining data from two domains we are able to learn better the types of argumentative relations. When using *syntactic* features, adding *micro*, *cdcp*, and *ukp* does not improve the results compared to using a single dataset for training. Indeed, *cdcp* has only one type of relation (i.e. support) resulting in an imbalanced dataset, and in *ukp*, the parent argument is a topic, which does not improve the prediction task. When using *all* features, *micro*, *com*, *ukp*, and *nk* do not contribute to an increase in performance. The best performing model is the attention mechanism trained on the *web* and *essay* datasets using *syntactic* features (0.544 macro average F_1).

Neural baselines with contextualised word embeddings. Contextualised word embeddings such as the Bidirectional Encoder Representations from Transformers (BERT) em-

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
embed. + syntactic	C G	F_1 A		0.35	0.43	0.48	0.31	0.45	-	0.58		0.43	0.433	0.526
		F_1 S		0.71	0.68	0.58	0.70	0.54	0.77	0.47		0.50	0.619	
	A G	F_1 A		0.37	0.58		0.53	0.53	-	0.61	0.59	0.55	0.537	0.526
		F_1 S		0.61	0.60		0.42	0.50	0.72	0.43	0.38	0.47	0.516	
	A G	F_1 A		0.36	0.48	0.43	0.39		-	0.52	0.45	0.51	0.449	**0.544**
		F_1 S		0.75	0.66	0.62	0.68		0.79	0.52	0.56	0.54	0.640	
all features	M G	F_1 A		0.37	0.43	0.43	0.40	0.46	-	0.71	-	0.46	0.466	0.532
		F_1 S		0.71	0.64	0.61	0.70	0.55	0.78	0.11	0.78	0.51	0.599	
	A G	F_1 A		0.36	0.54		0.50	0.51	-	0.59	0.59	0.55	0.520	0.535
		F_1 S		0.67	0.63		0.49	0.51	0.74	0.47	0.41	0.49	0.551	
	A G	F_1 A		0.43	0.54	0.49	0.46		-	0.59	0.63	0.63	0.539	0.539
		F_1 S		0.68	0.55	0.57	0.56		0.65	0.46	0.38	0.47	0.540	

Table 3. Experimental results with F_1 for **A**ttack and for **S**upport for the **C**oncat, **M**ix, and **A**ttention architectures, with **G**loVE embeddings. The blanks indicate the training datasets. The Average (Avg) and the Macro (Macr) Avg do not include the results for the training datasets.

			essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg	Macr Avg
BERT + syntactic	4B	F_1 A			0.55	0.47	0.53	0.50	-	0.56	0.48	0.45	0.506	0.526
	2D	F_1 S			0.61	0.57	0.59	0.49	0.69	0.47	0.48	0.46	0.545	
	4B	F_1 A		0.36	0.48	0.40	0.45	0.42	-	0.53		0.37	0.430	0.525
	1D	F_1 S		0.69	0.67	0.61	0.62	0.57	0.79	0.50		0.50	0.619	
	4B	F_1 A	0.50	0.36	0.46		0.50		-	0.52	0.47	0.50	0.473	**0.537**
	2D	F_1 S	0.61	0.62	0.59		0.61		0.74	0.52	0.50	0.61	0.600	
all features	4B	F_1 A			0.53	0.50	0.54	0.51	-	0.59	0.51	0.49	0.524	0.529
	1D	F_1 S			0.59	0.56	0.55	0.47	0.67	0.45	0.48	0.49	0.533	
	3B	F_1 A	0.48	0.34	0.48		0.45		-	0.45	0.50	0.54	0.463	0.532
	2D	F_1 S	0.57	0.65	0.60		0.64		0.73	0.55	0.52	0.54	0.600	

Table 4. Experimental results with F_1 for **A**ttack and for **S**upport relations. *X*B stands for the number of BERT layers used (*X*=3,4) and *Y*D stands for the number of dense layers (*Y*=1,2) used before the final layer that predicts the class. The blanks indicate the training datasets. The Average (Avg) and the Macro (Macr) Avg do not include the results for the training datasets.

beddings [12] analyse the entire sentence before assigning embeddings to individual words. We employ BERT embeddings to test whether they bring any improvements to the classification task. While for GloVe vectors we do not need the original, trained model in order to use the embeddings, for the BERT embeddings we require the pre-trained language models that we can then fine tune using the datasets of the downstream task. We try different combinations: using 3 or 4 BERT layers and using 1 dense layer (of 64 neurons) or 2 dense layers (of 128 and 32 neurons, respectively) before the final layer that determines the class. Table 4 shows the results with BERT embeddings instead of GloVe, using feature ablation (*syntactic* vs *all* features) and two datasets for training to test whether this can improve performance. The best results are obtained using 4 BERT layers and 2 dense layers (0.537 macro average F_1). However, this best BERT baseline does not outperform the best results with the attention model and GloVe embeddings.

4. Related work

In terms of results reported on the datasets we have conducted our experiments on, most works perform a cross-validation evaluation or, in the case of datasets consisting of several topics, the models proposed are trained on some of the topics and tested on the remaining topics. For *essay*, an Integer Linear Programming model was used to achieve 0.947 F_1 for *support* and 0.413 F_1 for *attack* on the testing dataset using cross-validation to select the model [33]. Using SVM, 0.946 F_1 for *support* and 0.456 F_1 for *attack* were obtained [33]. Using a modification of the Integer Linear Programming model to accommodate the lack of some features used for *essay* but not present in *micro*, 0.855 F_1 was obtained for *support* and 0.628 F_1 for *attack*. On *micro*, an evidence graph model was used to achieve 0.71 F_1 using cross-validation [29]. On *nk*, 0.77 F_1 for *attack* and 0.75 F_1 for *support* were obtained using SVM and cross-validation [23]. SVM accuracy results on the testing dataset using coverage (i.e. number of claims identified over the number of total claims) were reported in [1] as follows: 0.849 accuracy for 10% coverage, 0.740 accuracy for 60% coverage, 0.632 accuracy for 100% coverage. RF were evaluated on *web* and *aif* using cross-validation, achieving 0.717 F_1 and 0.831 F_1, respectively [8]. Structured SVMs were evaluated in a cross-validation setting on *cdcp* and *ukp* using various types of factor graphs, full and strict [25]. On *cdcp*, F_1 was 0.493 on the full graph and 0.50 on the strict graph, whereas on *ukp*, F_1 was 0.689 on the full graph and 0.671 on the strict graph. No results on the two-class datasets were reported for *db*, *com*, and *ukp*. The results on *ukp* treat either supporting and attacking arguments as a single category or consider three types of relations: *support*, *attack*, *neither*. The latter type of reporting results on three classes is also given on the *com*.

Some other works have started investigating the dataset independence in AM. [26] showed how models may overlook textual content when provided with the context surrounding the span by relying on contextual markers for predicting relations and tested their method on the *essay* dataset. [21] integrated (claim and other domain) lexicon information into neural networks with attention tested on *ukp*. [19] experimented with span representations, originally developed for other tasks, on the *essay* dataset. Other works have used contextualised word embeddings for relation prediction in AM [13,32]. More recently, [14] proposed and tested on *ukp* an argument retrieval system.

5. Discussion and Conclusion

Dataset independence is one of the biggest challenges in AM. An AM model for relation prediction trained on every individual dataset we considered in this paper would perform better than any general baseline on that dataset. We believe an AM model would require leveraging a diverse corpus to be of use in a real-world system. Most works have previously focused on a moderate-sized corpus distributed across a small set of topics [14]. This paper is a step towards the applicability of AM techniques across datasets. Our baselines perform homogeneously in terms of average over all existing datasets for relation prediction in AM while using generic features. We propose as baseline the model that performed the best, with the baseline using attention mechanism with GloVe embeddings and syntactic features trained on the *web* and *essay* datasets (0.544 macro average F_1). The results for the *attack* class are generally worse than those for *support* as the datasets that are used in training (e.g. *essay*, *ibm*) have fewer instances for the *attack* class than for *support* (see Table 1). The datasets differ at granularity: some consist of pairs of sentences (e.g., *ibm*) whereas others include pair of multiple-sentence arguments (e.g., *nk*). Additionally, the argumentative on relations can be domain-specific and their semantic nature may vary between corpora (e.g., *com*). We considered the unified task of determining *support* or *attack* between any two texts.

Embeddings represent the differentiating feature for the models we experimented with. Whilst word embeddings are often used as the first data processing layer in a deep learning model, we employed TF-IDF features for the non-neural models that we considered as baselines. Other works that address the task of relation prediction make use of features specific to the single dataset of interest, making it difficult to test those models on other datasets. For instance, for the *essay* dataset, [33] use structural features such as number of preceding and following tokens in the covering sentence, number of components in paragraph, number of preceding and following components in paragraph, relative position of the argument component in paragraph. For the other datasets, [34] use topic similarity features (as the *parent* argument is a topic), [23] use the position of the topic and similarity with other related/unrelated pair from the dataset, keyword embeddings of topics from the dataset. We have used only general purpose features that are meaningful for all datasets addressing the relational AM task. Surprisingly, BERT embeddings (achieving state-of-the-art performances in many tasks [12]) do not bring improvements here, compared to non-contextualised word embeddings.

To conclude, several resources have been built recently for the task of argumentative relation prediction, covering different topics like political speeches, Wikipedia articles, persuasive essays. Given the heterogeneity of these different kinds of text, it is hard to compare cross-dataset the different proposed approaches. We addressed this non-portability issue by making a broad comparison of different deep learning methods using both non-contextualised and contextualised word embeddings for a large set of datasets for the argumentative relation prediction task, an important and still widely open problem. We proposed a set of strong dataset-independent baselines based on several neural architectures and have shown that our models perform homogeneously over all existing datasets for relation prediction in AM.

References

[1] Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A., Slonim, N.: Stance classification of context-dependent claims. In: Proceedings of EACL. pp. 251–261 (2017)

[2] Bex, F., Modgil, S., Prakken, H., Reed, C.: On logical specifications of the argument interchange format. Journal of Logic and Computation 23(5), 951–989 (2013)

[3] Boltužić, F., Šnajder, J.: Back up your stance: Recognizing arguments in online discussions. In: Proceedings of the 1st Workshop on Argumentation Mining. pp. 49–58 (2014)

[4] Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)

[5] Cabrio, E., Villata, S.: Node: A benchmark of natural language arguments. In: Proceedings of COMMA. pp. 449–450 (2014)

[6] Cabrio, E., Villata, S.: Five years of argument mining: a data-driven analysis. In: Proceedings of IJCAI. pp. 5427–5433 (2018)

[7] Carstens, L., Toni, F.: Towards relation based argumentation mining. In: Proceedings of the 2nd Workshop on Argumentation Mining. pp. 29–34 (2015)

[8] Carstens, L., Toni, F.: Using argumentation to improve classification in natural language problems. ACM Transactions on Internet Technology 17(3), 30:1–30:23 (2017)

[9] Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology TIST 2(3), 27:1–27:27 (2011)

[10] Chesñevar, C.I., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G.R., South, M., Vreeswijk, G., Willmott, S.: Towards an argument interchange format. Knowledge Eng Review 21(4), 293–316 (2006)

[11] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In: Proceedings of EMNLP. pp. 1724–1734 (2014)

[12] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)

[13] Durmus, E., Ladhak, F., Cardie, C.: Determining relative argument specificity and stance for complex argumentative structures. In: Proceedings of ACL. pp. 4630–4641 (2019)

[14] Ein-Dor, L., Shnarch, E., Dankin, L., Halfon, A., Sznajder, B., Gera, A., Alzate, C., Gleize, M., Choshen, L., Hou, Y., Bilu, Y., Aharonov, R., Slonim, N.: Corpus wide argument mining - A working solution. In: Proceedings of AAAI. pp. 7683–7691 (2020)

[15] Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. pp. 417–422 (2006)

[16] Habernal, I., Gurevych, I.: Argumentation mining in user-generated web discourse. Computational Linguistics 43(1), 125–179 (2017)

[17] Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of ICWSM (2014)

[18] Iyad, R., Reed, C.: The argument interchange format. In: Argumentation in Artificial Intelligence, pp. 383–402. Springer (2009)

[19] Kuribayashi, T., Ouchi, H., Inoue, N., Reisert, P., Miyoshi, T., Suzuki, J., Inui, K.: An empirical study of span representations in argumentation structure parsing. In: Proceedings of ACL. pp. 4691–4698 (2019)

[20] Lawrence, J., Reed, C.: Argument mining: A survey. Computational Linguistics 45(4), 765–818 (2019)

[21] Lin, J., Huang, K.Y., Huang, H., Chen, H.: Lexicon guided attentive neural network model for argument mining. In: Proceedings of the 6th Workshop on Argument Mining. pp. 67–73 (2019)

[22] Lippi, M., Torroni, P.: Argumentation mining: State of the art and emerging trends. ACM Transactions on Internet Technology 16(2), 10 (2016)

[23] Menini, S., Cabrio, E., Tonelli, S., Villata, S.: Never retreat, never retract: Argumentation analysis for political speeches. In: Proceedings of AAAI. pp. 4889–4896 (2018)

[24] Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38, 39–41 (1995)

[25] Niculae, V., Park, J., Cardie, C.: Argument mining with structured SVMs and RNNs. In: Proceedings of ACL. pp. 985–995 (2017)

[26] Opitz, J., Frank, A.: Dissecting content and context in argumentative relation analysis. In: Proceedings of the 6th Workshop on Argument Mining. pp. 25–34 (2019)

[27] Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of EMNLP. pp. 2249–2255 (2016)

[28] Park, J., Cardie, C.: A corpus of eRulemaking user comments for measuring evaluability of arguments. In: Proceedings of LREC (2018)

[29] Peldszus, A., Stede, M.: Joint prediction in MST-style discourse parsing for argumentation mining. In: Proceedings of EMNLP. pp. 938–948 (2015)

[30] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of EMNLP. pp. 1532–1543 (2014)

[31] Reed, C., Wells, S., Devereux, J., Rowe, G.: AIF+: dialogue in the argument interchange format. In: Proceedings of COMMA. pp. 311–323 (2008)

[32] Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: Proceedings of ACL. pp. 567–578 (2019)

[33] Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. Computational Linguistics 43(3), 619–659 (2017)

[34] Stab, C., Miller, T., Schiller, B., Rai, P., Gurevych, I.: Cross-topic argument mining from heterogeneous sources. In: Proceedings of EMNLP (2018)

[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of NeurIPS. pp. 6000–6010 (2017)

[36] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: Proceedings of NAACL HLT. pp. 1480–1489 (2016)

Regular Papers

This page intentionally left blank

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200492

PageRank as an Argumentation Semantics

Emanuele ALBINI [a,b,1], Pietro BARONI [a], Antonio RAGO [b] and Francesca TONI [b]
[a] *Dip.to di Ingegneria dell'Informazione, Università degli Studi di Brescia, Italy*
[b] *Dept. of Computing, Imperial College London, UK*

Abstract. This paper provides an initial exploration on the relationships between *PageRank* and gradual argumentation semantics. After showing that PageRank, directly interpreted as an argumentation semantics for support frameworks, fails to satisfy some generally desirable properties, we propose a novel approach to reconstruct PageRank as gradual semantics of a suitably defined bipolar argumentation framework, while satisfying these desirable properties. The theoretical advantages of the approach are complemented by an illustration of its potential application to support the generation of better explanations of PageRank scores for end users.

Keywords. PageRank, Gradual Argumentation Semantics, Bipolar Argumentation Frameworks

1. Introduction

In the context of search engines, a user wants to find the (web) pages that are the most relevant to a search query, potentially among millions of them. The web has an essential feature: each piece of information (page) may link to other pieces of information (through hyperlinks), and therefore the web organization can be regarded as a directed graph, where pages are nodes and links are the edges. This is the idea that in 1999 inspired the revolutionary PageRank (PR) algorithm [1]: a method for computing a ranking score for every page based on the graph structure of the web. Given its conceptual simplicity and general formalization for any kind of directed graph, PR has been applied to many other domains where entities can be evaluated on the basis of their connections to other entities, including citation networks [2], recommendation systems [3], chemistry [4], biology [5] and neuroscience [6], and has been studied from several perspectives including an axiomatic characterization from a social choice theory perspective [7].

As well-known, graph-based representations are also pervasive in the field of computational argumentation. In particular Dung's abstract argumentation frameworks [8] are essentially directed graphs whose nodes are arguments and edges represent attacks. Dung's seminal proposal has been subsequently extended in several directions, e.g. bipolar argumentation frameworks [9] encompass also a notion of support, while in quantitative bipolar argumentation frameworks [10] a base score is assigned to each argument. In this context, the argument graph structure is the basis of the assessment of argument acceptability according to some *argumentation semantics* [11]: in Dung's traditional approach the evaluation is qualitative, while in further developments numerical argument assessments based on *gradual semantics* have been investigated [12,10].

[1] Corresponding Author: emanuele.albini19@imperial.ac.uk

Given the similarity between PR and gradual argumentation semantics as formal tools producing a numerical assessment of connected entities in a graph, it appears that drawing bridges between the two areas and exploring possible cross-fertilization opportunities represents an interesting research direction. This paper provides some initial contribution in this respect by first exploring the use of PR as a gradual semantics for *support argumentation frameworks* [13], then evidencing some limitations of this simplistic correspondence and proposing a novel approach to reconstruct PR as a semantics in suitably constructed *quantitative bipolar argumentation frameworks* (QBAFs) in which pages will be interpreted as arguments ignoring their content. Besides featuring better theoretical properties, this approach has the significant advantage of supporting more effective explanations of PR outcomes to users.

In a broader perspective this paper contribution is two-fold. On one hand we define a new gradual semantics for QBAFs based on PageRank. On the other hand, we support the idea of using argumentation frameworks, not only to model dialectical debates, but also to describe the mechanism of algorithms in order to present them in a dialectical form, with the aim of either generating explanations or enabling other practical applications.

The paper is organized as follows. In Section 2 we recall some background concepts on PR. In Section 3 we detail how PR can be directly interpreted as a gradual semantics in support argumentation frameworks, showing however that, as such, it does not satisfy some desirable properties in this context. In Section 4 we reconstruct PR as a gradual semantics of suitable QBAFs, achieving in this way the satisfaction of the above mentioned desirable properties. In Section 5 we discuss the advantages of the proposed approach, with particular reference to the explanation of PR outcomes. We conclude the paper and outline lines of future work in Section 6.

2. PageRank Background

We firstly recall the PR definition from the original paper [1], using a different but equivalent notation when necessary for our purposes.

We assume a set of pages/nodes $\mathcal{P} = \{u_1, u_2, ..., u_N\}$ and a set of links between the pages $\mathcal{L} \subseteq \mathcal{P} \times \mathcal{P}$, where $(u, v) \in \mathcal{L}$ indicates that there is a link from page u to page v. We say $N = |\mathcal{P}| > 0$ is the total number of pages, $O_u = \{v \in \mathcal{P}{:}(u, v) \in \mathcal{L}\}$ is the set of pages u points to and $I_u = \{v \in \mathcal{P}{:}(v, u) \in \mathcal{L}\}$ is the set of pages that point to u. We assume that $\forall u \in \mathcal{P}, \nexists(u, u) \in \mathcal{L}$, i.e. self-loops are ignored to prevent the manipulation of PR. We also assume that $\forall u \in \mathcal{P}, |O_u| > 0$, i.e. there are no *dangling* pages, that is, no pages without outgoing links (in practice, if such a page is found it is treated as having links towards all other pages as in [14]).

A *random surfer model* is used, which is based on the assumption that a user can either reach a page from a link in another page with probability $d \in]0; 1[$, referred to as *damping factor*, or land on a page directly with probability $1 - d$. Unless otherwise specified, we assume the value suggested in [1] of $d = 0.85$ and a uniform probability of directly landing on a page (i.e. we focus on *non-personalized PR*). In Section 6 we discuss how in future works these assumptions could be changed.

Definition 1. *[1] The* PageRank *of a set of pages is an assignment,* $R : \mathcal{P} \to]0, 1]$*, to the pages which satisfies:*

$$R(u) = (1-d) \cdot \frac{1}{N} + d \cdot \sum_{v \in I_u} \frac{R(v)}{|O_v|} \quad \forall u \in \mathcal{P}$$

Note that R is the solution of a system of linear equations derived from Def. 1 (we refer to R as both the assignment and the vector resulting from it). Notice also that, as described in [14], R is unique and $||R||_1 = 1$, i.e. the L_1 norm of R is 1.

The aim of PR is to give to every page a score that describes how relevant it is: the higher the score, the more important the page. Thus, these scores are based on their relevance, which is intended to approximate the amount of users visiting the page. The latter is calculated through a mathematical model aiming at probabilistically estimating the number of users. The assumption here is therefore that the higher the number of links to (from) a page, the more it (the less each page linked by it, resp.) will be visited and hence the higher (lower, resp.) its PR score should be.

3. PageRank as a Gradual Semantics

In this section we show how PR may be interpreted directly as a gradual argumentation semantics and examine its ability to satisfy some desirable properties. First, we recall some necessary formal notions from [10].

Definition 2. *[10] A* Quantitative Bipolar Argumentation Framework *(QBAF) is a 4-tuple $\langle \mathcal{X}, \mathcal{R}^-, \mathcal{R}^+, \tau \rangle$, comprising:*

- *a finite set of arguments $\mathcal{X}$*
- *a binary attack relation between arguments $\mathcal{R}^- \subseteq \mathcal{X} \times \mathcal{X}$*
- *a binary support relation between arguments $\mathcal{R}^+ \subseteq \mathcal{X} \times \mathcal{X}$*
- *a total function $\tau : \mathcal{X} \to \mathbb{I}$, with $\tau(\alpha)$ the* base score *of α*

where $\mathbb{I}$ is a set equipped with a preorder $\leq$ where, as usual, $a<b$ denotes $a \leq b$ and $b \not\leq a$. Given a QBAF, a total function $\sigma : \mathcal{X} \to \mathbb{I}$, called a gradual semantics, *may be used to assign a* strength *to each argument. We define an sQBAF as a QBAF such that $\mathcal{R}^- = \emptyset$. Finally, we let $\mathcal{R}^-(\alpha) = \{\beta \in \mathcal{X} : (\beta, \alpha) \in \mathcal{R}^-\}$ and $\mathcal{R}^+(\alpha) = \{\beta \in \mathcal{X} : (\beta, \alpha) \in \mathcal{R}^+\}$, and similarly $\mathfrak{R}^-(\alpha) = \{\beta \in \mathcal{X} : (\alpha, \beta) \in \mathcal{R}^-\}$ and $\mathfrak{R}^+(\alpha) = \{\beta \in \mathcal{X} : (\alpha, \beta) \in \mathcal{R}^+\}$.*

A *web graph* $\langle \mathcal{P}, \mathcal{L} \rangle$ can be interpreted as an sQBAF where the pages (nodes) are arguments and the links between them (edges) are supports, as follows.

Definition 3. *Given a set of pages $\mathcal{P}$ and a set of links $\mathcal{L}$, a* PageRank Argumentation Framework *(PRAF) is an sQBAF defined as $PR = \langle \mathcal{X}, \emptyset, \mathcal{R}^+, \tau \rangle$, where:*

- *$\mathcal{X} = \mathcal{P}$ is the set of arguments corresponding to the set of pages*
- *$\mathcal{R}^+ = \mathcal{L}$ is the set of supports corresponding to the set of links between pages*
- *$\tau : \mathcal{X} \mapsto \mathbb{I} = [\frac{1-d}{|\mathcal{X}|}, 1]$ is the base score, defined as a constant function:*

$$\tau(\alpha) = \frac{1-d}{|\mathcal{X}|} \quad \forall \alpha \in \mathcal{X}$$

Given Def. 1 and the notes on loops and dangling nodes in Section 2, Remark 1 can be trivially derived.

Remark 1. *Given a PRAF it always holds that:*

- *each argument has at least one outgoing link:* $|\mathfrak{R}^{+}(\alpha)| > 0,\ \forall \alpha \in \mathcal{X}$
- *there are no self-supports:* $\nexists(\alpha, \alpha) \in \mathcal{R}^{+},\ \forall \alpha \in \mathcal{X}$.

We then interpret PR as a gradual semantics for sQBAF.

Definition 4. *The PageRank semantics is a gradual semantics* $\sigma : \mathcal{X} \mapsto \mathbb{I}$ *such that:*

$$\sigma(\alpha) = \tau(\alpha) + d \cdot \sum_{\beta \in \mathcal{R}^{+}(\alpha)} \frac{\sigma(\beta)}{|\mathfrak{R}^{+}(\beta)|} \quad \forall \alpha \in \mathcal{X}$$

The following lemma is directly derived from Def. 4.

Lemma 1. *The codomain of* σ *is* $\mathbb{I} = [\frac{1-d}{|\mathcal{X}|}, 1]$ *where* $\perp > 0$.

In order to formally assess PR as an argumentation semantics, we now review some desirable properties for argument strength, called *group properties* (GPs) in [10], as they imply a group of other properties. Some preliminary definitions need to be recalled first. Given a QBAF $\langle \mathcal{X}, \mathcal{R}^{-}, \mathcal{R}^{+}, \tau \rangle$ and a gradual semantics σ, for any $A \subseteq \mathcal{X}$, we refer to the multiset $\{\sigma(\beta) : \beta \in A\}$ as A_σ. Given $A, B \subseteq \mathcal{X}$, A is *strength equivalent to* B, denoted $A \overset{\sigma}{=} B$, iff $A_\sigma = B_\sigma$; A is *at least as strong as* B, denoted $A \overset{\sigma}{\geq} B$, iff there exists an injective mapping f from B to A such that $\forall \alpha \in B$, $\sigma(f(\alpha)) \geq \sigma(\alpha)$; and A is *stronger than* B, denoted $A \overset{\sigma}{>} B$, iff $A \overset{\sigma}{\geq} B$ and $B \overset{\sigma}{\ngeq} A$.

GPs are then defined as follows (some being reformulated in more general or more specific ways wrt [10], where useful for our present purposes):

GP1. If $\mathcal{R}^{-}(\alpha) = \emptyset$ and $\mathcal{R}^{+}(\alpha) = \emptyset$ then $\sigma(\alpha) = \tau(\alpha)$.
GP2. If $\mathcal{R}^{-}(\alpha) \neq \emptyset$ and $\mathcal{R}^{+}(\alpha) = \emptyset$ then $\sigma(\alpha) < \tau(\alpha)$.
GP3. If $\mathcal{R}^{-}(\alpha) = \emptyset$ and $\mathcal{R}^{+}(\alpha) \neq \emptyset$ then $\sigma(\alpha) > \tau(\alpha)$.
GP4. If $\sigma(\alpha) < \tau(\alpha)$ then $\mathcal{R}^{-}(\alpha) \neq \emptyset$.
GP5. If $\sigma(\alpha) > \tau(\alpha)$ then $\mathcal{R}^{+}(\alpha) \neq \emptyset$.
GP6. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{=} \mathcal{R}^{-}(\beta)$, $\mathcal{R}^{+}(\alpha) \overset{\sigma}{=} \mathcal{R}^{+}(\beta)$ and $\tau(\alpha) = \tau(\beta)$ then $\sigma(\alpha) = \sigma(\beta)$.
GP7. If $\mathcal{R}^{-}_{\sigma}(\alpha) \subsetneq \mathcal{R}^{-}_{\sigma}(\beta)$, $\mathcal{R}^{+}(\alpha) \overset{\sigma}{=} \mathcal{R}^{+}(\beta)$ and $\tau(\alpha) = \tau(\beta)$ then $\sigma(\beta) < \sigma(\alpha)$.
GP8. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{=} \mathcal{R}^{-}(\beta)$, $\mathcal{R}^{+}_{\sigma}(\alpha) \subsetneq \mathcal{R}^{+}_{\sigma}(\beta)$ and $\tau(\alpha) = \tau(\beta)$ then $\sigma(\alpha) < \sigma(\beta)$.
GP9. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{=} \mathcal{R}^{-}(\beta)$, $\mathcal{R}^{+}(\alpha) \overset{\sigma}{=} \mathcal{R}^{+}(\beta)$ and $\tau(\alpha) < \tau(\beta)$ then $\sigma(\alpha) < \sigma(\beta)$.
GP10. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{<} \mathcal{R}^{-}(\beta)$, $\mathcal{R}^{+}(\alpha) \overset{\sigma}{=} \mathcal{R}^{+}(\beta)$ and $\tau(\alpha) = \tau(\beta)$ then $\sigma(\beta) < \sigma(\alpha)$.
GP11. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{=} \mathcal{R}^{-}(\beta)$, $\mathcal{R}^{+}(\alpha) \overset{\sigma}{>} \mathcal{R}^{+}(\beta)$ and $\tau(\alpha) = \tau(\beta)$ then $\sigma(\beta) < \sigma(\alpha)$.

In [10], two general principles (and their strict counterparts) were also identified as a more synthetic way of describing the desirable properties of a gradual semantics.

The intuition for the first principle is that a difference in an argument's strength and base score must correspond to an imbalance in its attackers' and supporters' strengths.

Principle 1. *[10] A gradual semantics* σ *is* balanced *iff for any* $\alpha \in \mathcal{X}$*:*

1. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{=} \mathcal{R}^{+}(\alpha)$ *then* $\sigma(\alpha) = \tau(\alpha)$.
2. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{>} \mathcal{R}^{+}(\alpha)$ *then* $\sigma(\alpha) < \tau(\alpha)$.
3. If $\mathcal{R}^{-}(\alpha) \overset{\sigma}{<} \mathcal{R}^{+}(\alpha)$ *then* $\sigma(\alpha) > \tau(\alpha)$.

A gradual semantics σ is strictly balanced *iff σ is balanced and for any $\alpha \in \mathcal{X}$:*
4. If $\sigma(\alpha) < \tau(\alpha)$ then $\mathcal{R}^-(\alpha) \overset{\sigma}{>} \mathcal{R}^+(\alpha)$.
5. If $\sigma(\alpha) > \tau(\alpha)$ then $\mathcal{R}^-(\alpha) \overset{\sigma}{<} \mathcal{R}^+(\alpha)$.

In [10] it is shown that if σ is balanced then it satisfies GP1 to GP3 and if it is strictly balanced then it satisfies GP1 to GP5.

The second principle requires that the strength of an argument depends monotonically on its base score and on the strengths of its attackers and supporters. To introduce this principle formally, we first recall the notion of shaping triple of an argument, where for any $\alpha \in \mathcal{X}$, the *shaping triple* of α is $(\tau(\alpha), \mathcal{R}^+(\alpha), \mathcal{R}^-(\alpha))$, denoted $\mathcal{ST}(\alpha)$. Given $\alpha, \beta \in \mathcal{X}$, $\mathcal{ST}(\beta)$ is said to be: *as boosting as* $\mathcal{ST}(\alpha)$, denoted as $\mathcal{ST}(\alpha) \simeq \mathcal{ST}(\beta)$, iff $\tau(\alpha) = \tau(\beta)$, $\mathcal{R}^+(\alpha) \overset{\sigma}{=} \mathcal{R}^+(\beta)$, and $\mathcal{R}^-(\beta) \overset{\sigma}{=} \mathcal{R}^-(\alpha)$; *at least as boosting as* $\mathcal{ST}(\alpha)$, denoted as $\mathcal{ST}(\alpha) \preceq \mathcal{ST}(\beta)$, iff $\tau(\alpha) \leq \tau(\beta)$, $\mathcal{R}^+(\alpha) \overset{\sigma}{\leq} \mathcal{R}^+(\beta)$, and $\mathcal{R}^-(\beta) \overset{\sigma}{\leq} \mathcal{R}^-(\alpha)$; or *strictly more boosting than* $\mathcal{ST}(\alpha)$, denoted as $\mathcal{ST}(\alpha) \prec \mathcal{ST}(\beta)$, iff $\mathcal{ST}(\alpha) \preceq \mathcal{ST}(\beta)$ and $\mathcal{ST}(\beta) \npreceq \mathcal{ST}(\alpha)$.

Principle 2. *[10] A gradual semantics σ is* monotonic *iff:*
1. for any $\alpha, \beta \in \mathcal{X}$, if $\mathcal{ST}(\alpha) \simeq \mathcal{ST}(\beta)$ then $\sigma(\alpha) = \sigma(\beta)$;
2. if $\mathcal{ST}(\alpha) \preceq \mathcal{ST}(\beta)$ then $\sigma(\alpha) \leq \sigma(\beta)$.
A gradual semantics σ is strictly monotonic iff σ is monotonic and:
3. for any $\alpha, \beta \in \mathcal{X}$, if $\mathcal{ST}(\alpha) \prec \mathcal{ST}(\beta)$ then $\sigma(\alpha) < \sigma(\beta)$.

In [10] it is shown that if σ is (strictly) monotonic then it satisfies GP6 to GP11.

We will now show that the PR semantics σ satisfies some, but not all, of the desirable properties for gradual semantics. We will consider whether or not the properties are satisfied by the semantics σ when applied to a generic QBAF, in Proposition 1 and 2, or when applied to a PRAF (denoted as $\langle PR, \sigma \rangle$), in Proposition 3 and 4 (see Table 1 for a compact summary). Note that in the first case, if attacks are present in the QBAF, they are simply ignored by the definition of the semantics, and some of the properties may not hold for this mere reason. Proofs for the *satisfied* GPs and principles have been omitted for lack of space, as they are not essential for this paper.

Proposition 1. *σ satisfies GP1, GP3, GP4, GP5 but not GP2, and thus is not balanced.*

Proof. GP2 does not hold as when $\mathcal{R}^+(\alpha) = \emptyset$, $\sigma(\alpha) = \tau(\alpha)$ independently of $\mathcal{R}^-(\alpha)$, which is ignored in the definition of σ. □

Proposition 2. *σ satisfies GP8 and GP9 but not GP6, GP7, GP10 and GP11, and thus is not monotonic.*

Proof. GP6: in the framework in Figure 1, we have $\mathcal{R}^+(\beta) \overset{\sigma}{=} \mathcal{R}^+(\delta)$ but $\sigma(\beta) \neq \sigma(\delta)$. GP7 and GP10 cannot hold as attackers do not affect σ. GP11: in the framework in Figure 1, we have $\mathcal{R}^+(\zeta) \overset{\sigma}{>} \mathcal{R}^+(\eta)$ but $\sigma(\zeta) < \sigma(\eta)$. □

Proposition 3. *$\langle PR, \sigma \rangle$ is strictly balanced and thus satisfies GP1 to GP5.*

Proposition 4. *$\langle PR, \sigma \rangle$ satisfies GP7 to GP10 but not GP6 or GP11 (provable by the counter-examples in Proposition 2), and thus is not monotonic.*

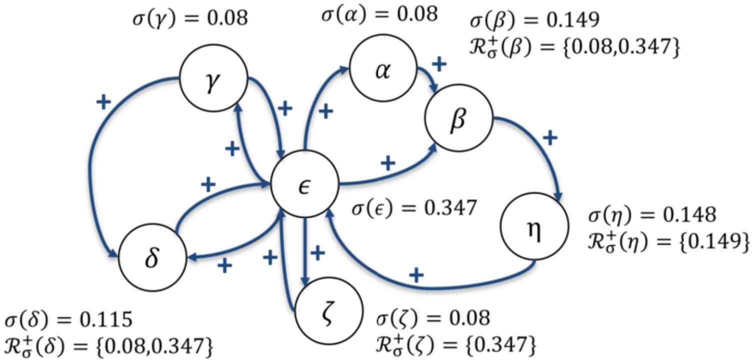

σ(γ) = 0.08
σ(α) = 0.08
σ(β) = 0.149
$\mathcal{R}^+_\sigma(\beta)$ = {0.08,0.347}
γ
α
β
ε
η
δ
ζ
+
σ(ε) = 0.347
σ(η) = 0.148
$\mathcal{R}^+_\sigma(\eta)$ = {0.149}
σ(δ) = 0.115
$\mathcal{R}^+_\sigma(\delta)$ = {0.08,0.347}
σ(ζ) = 0.08
$\mathcal{R}^+_\sigma(\zeta)$ = {0.347}

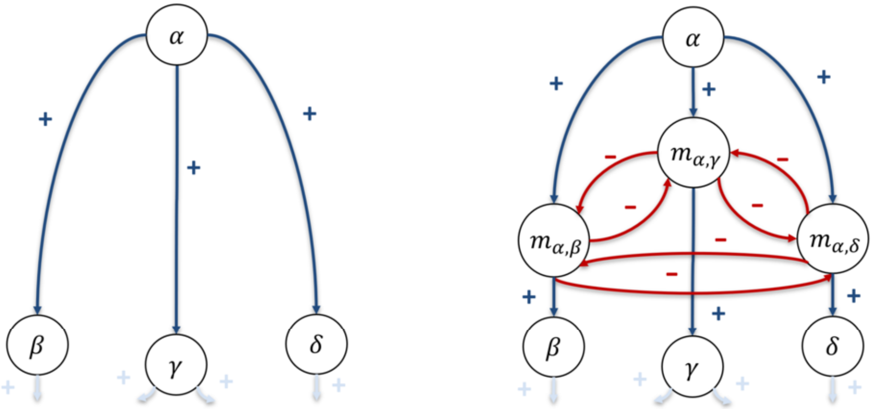
Figure 1. Counter-example to GP6 and GP11 for the PR semantics σ in Proposition 2.

We have thus shown that directly interpreting PR as a gradual semantics for an sQBAF does not give rise to a satisfactory outcome in terms of formal properties. Indeed, while using PR as a semantics is somehow straightforward, it does not appear fully appropriate from a modeling perspective, as it does not provide a suitable argumentative counterpart to some key aspects of PR. In particular, note that, as a consequence of the PR definition, the strength of each node depends not only on the strengths of its supporters but also on the cardinality of their outgoing supports. This has quite counter-intuitive effects from an argumentation perspective. For example, consider the situation where two nodes have the same strength $\sigma(\alpha) = \sigma(\beta)$, but α has one outgoing support, while β has ten: the latter's support to each of its children is actually ten times 'less powerful' (i.e. it transfers $1/10$ of the strength) than the former's. It follows that a node γ supported by α only and a node δ supported by β only would have different strengths even if their supporters appear to be equivalent (formally the shaping triples of γ and δ are the same). This is the main reason for the lack of many desirable properties and calls for an alternative approach, which we introduce next.

4. PageRank as a Gradual Semantics in a Meta-Argumentation Framework

In this section, we introduce an alternative approach to capture PageRank as an argumentation semantics. To this purpose we transform the sQBAF corresponding to a set of linked pages into a QBAF including additional meta-arguments and attacks between them. The underlying intuition is that each additional meta-argument can be understood as a vehicle of support from one page to another and that supports from the same page are in mutual conflict as they 'compete' in drawing strength from the same source.

In particular, as shown in Figure 2, we add a meta-argument on every support relationship in the original PRAF, and all the meta-arguments supported by the same page attack each other. While the 'regular' arguments still represent the pages, these new meta-arguments correspond to the links between them. This increases the expressivity of the representation, allowing in particular attacks between the meta-arguments corresponding to links from the same page in order to describe the fact that they 'compete' for conveying strength, as mentioned above, and therefore the more links originating from the same page, the lower the strength transferred through each of them.



(a) PRAF

(b) MPRAF

Figure 2. Example of a transformation from a PRAF to an MPRAF.

Definition 5. *Given a PRAF $PR = \langle \mathcal{X}, \emptyset, \mathcal{R}^+, \tau \rangle$, the* PageRank Meta-Argumentation Framework *(MPRAF) derived from PR is a QBAF $\langle \mathcal{X} \cup \mathcal{M}, \widehat{\mathcal{R}}^-, \widehat{\mathcal{R}}^+, \widehat{\tau} \rangle$, where:*

- $\mathcal{M} = \{m_{\alpha,\beta} : (\alpha, \beta) \in \mathcal{R}^+\}$ *is the set of meta-arguments*
- $\widehat{\mathcal{R}}^+ = \{(\alpha, m_{\alpha,\beta}), (m_{\alpha,\beta}, \beta) : \alpha, \beta \in \mathcal{X}, m_{\alpha,\beta} \in \mathcal{M}\}$ *is the set of supports*
- $\widehat{\mathcal{R}}^- = \{(m_{\alpha,\beta}, m_{\alpha,\gamma}) \in \mathcal{M} \times \mathcal{M} : (\alpha, \beta), (\alpha, \gamma) \in \mathcal{R}^+\}$ *is the set of attackers*
- $\widehat{\tau} : \mathcal{X} \cup \mathcal{M} \mapsto \widehat{\mathbb{I}} = [0, 1[$ *is the base score defined as the function:*

$$\widehat{\tau}(\alpha) = \begin{cases} 0 & \text{if } \alpha \in \mathcal{M} \\ \frac{1-d}{|\mathcal{X}|} & \text{if } \alpha \in \mathcal{X} \end{cases}$$

Figure 2 illustrates the transformation of a PRAF into an MPRAF: the supports go from a 'regular' argument to another through an intermediate meta-argument. The following remarks illustrate some of the properties of MPRAFs $\langle \mathcal{X} \cup \mathcal{M}, \widehat{\mathcal{R}}^-, \widehat{\mathcal{R}}^+, \widehat{\tau} \rangle$.

Remark 2. *For any $\alpha \in \mathcal{X}$, $\widehat{\mathcal{R}}^-(\alpha) = \emptyset$.*

Remark 3. *For any $m_{\alpha,\beta} \in \mathcal{M}$, $\exists! \alpha \in \widehat{\mathcal{R}}^+(m_{\alpha,\beta})$, $\exists! \beta \in \widehat{\mathfrak{R}}^+(m_{\alpha,\beta})$, $\alpha \in \mathcal{X}$ and $\beta \in \mathcal{X}$.*

Remark 4. *For any $m_{\alpha,\beta} \in \mathcal{M}$, $|\widehat{\mathcal{R}}^-(m_{\alpha,\beta})| + 1 = |\mathfrak{R}^+(\alpha)| = |\widehat{\mathfrak{R}}^+(\alpha)|$.*

Remark 5. *For any $\alpha \in \mathcal{X}$ where $\exists!\ m_{\alpha,\beta} : (\alpha, m_{\alpha,\beta}) \in \widehat{\mathcal{R}}^+$, $\widehat{\mathcal{R}}^-(m_{\alpha,\beta}) = \emptyset$.*

With reference to MPRAFs, we now define a gradual semantics $\widehat{\sigma}$, whose outcomes on 'regular' arguments coincide with the score produced by PR, as proved in Thm. 1.

Definition 6. *The* Meta-PageRank semantics *(M-PR) is a gradual semantics $\widehat{\sigma} : \mathcal{X} \cup \mathcal{M} \mapsto \widehat{\mathbb{I}}$ such that:*

$$\widehat{\sigma}(\alpha) = \widehat{\tau}(\alpha) + \sqrt{d} \cdot \frac{\sum_{\beta \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\alpha)| + 1} \quad \forall \alpha \in \mathcal{X} \cup \mathcal{M}$$

We now prove that, given a PRAF and corresponding MPRAF, for any $\alpha \in \mathcal{X}$, the strength $\widehat{\sigma}(\alpha)$ according to Def. 6 is the same as the strength $\sigma(\alpha)$ according to Def. 4, i.e. to the PR score.

Theorem 1. *Given a PRAF $\langle \mathcal{X}, \emptyset, \mathcal{R}^+, \tau \rangle$, denoted as PR, and the corresponding MPRAF $\langle \mathcal{X} \cup \mathcal{M}, \widehat{\mathcal{R}}^-, \widehat{\mathcal{R}}^+, \widehat{\tau} \rangle$, denoted as $\widehat{PR}$, with the semantics σ for PR and $\widehat{\sigma}$ for $\widehat{PR}$, for any argument $\alpha \in \mathcal{X}$ it holds that $\sigma(\alpha) = \widehat{\sigma}(\alpha)$.*

Proof. By Def. 6, $\widehat{\sigma}(\alpha) = \frac{1-d}{|\mathcal{X}|} + \sqrt{d} \cdot \frac{\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma)}{|\widehat{\mathcal{R}}^-(\alpha)|+1}$. By hypothesis $\alpha \in \mathcal{X}$, thus if $\gamma \in \widehat{\mathcal{R}}^+(\alpha)$ then $\gamma \in \mathcal{M}$, so we can rewrite γ as $m_{\beta,\alpha}$ where $\beta \in \mathcal{R}^+(\alpha)$. By the same hypothesis, we can derive, by Rem. 2, that $|\widehat{\mathcal{R}}^-(\alpha)| = 0$. This means that $\widehat{\sigma}(\alpha)$ can be rewritten as $\frac{1-d}{|\mathcal{X}|} + \sqrt{d} \cdot \sum_{m_{\beta,\alpha} \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(m_{\beta,\alpha})$. Expliciting $\widehat{\sigma}(m_{\beta,\alpha})$ by Def. 6 and recalling that, by Def. 5, $\tau(m_{\beta,\alpha}) = 0$ because $m_{\beta,\alpha}$ is a meta-argument, $\widehat{\sigma}(\alpha) = \frac{1-d}{|\mathcal{X}|} + \sqrt{d} \cdot \sum_{m_{\beta,\alpha} \in \widehat{\mathcal{R}}^+(\alpha)} \left(\sqrt{d} \cdot \frac{\sum_{\beta \in \widehat{\mathcal{R}}^+(m_{\beta,\alpha})} \widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(m_{\beta,\alpha})|+1} \right)$. We recall that, by Rem. 3, $\exists! \beta : \beta \in \widehat{\mathcal{R}}^+(m_{\beta,\alpha})$ because $m_{\beta,\alpha} \in \mathcal{M}$. Furthermore, we know by Rem. 4 that $|\widehat{\mathcal{R}}^-(m_{\beta,\alpha})| + 1 = |\mathfrak{R}^+(\beta)|$. Thus, $\widehat{\sigma}(\alpha) = \frac{1-d}{|\mathcal{X}|} + d \cdot \sum_{m_{\beta,\alpha} \in \widehat{\mathcal{R}}^+(\alpha)} \frac{\widehat{\sigma}(\beta)}{|\mathfrak{R}^+(\beta)|}$. This is equivalent to $\widehat{\sigma}(\alpha) = \frac{1-d}{|\mathcal{X}|} + d \cdot \sum_{\beta \in \mathcal{R}^+(\alpha)} \frac{\widehat{\sigma}(\beta)}{|\mathfrak{R}^+(\beta)|} = \sigma(\alpha)$. □

Lemma 2 proves that the codomain of $\widehat{\sigma}$ is $\widehat{\mathbb{I}}$.

Lemma 2. *The codomain of $\widehat{\sigma}$ on an MPRAF $\langle \mathcal{X} \cup \mathcal{M}, \widehat{\mathcal{R}}^-, \widehat{\mathcal{R}}^+, \widehat{\tau} \rangle$ is $\widehat{\mathbb{I}} =]0, 1]$. Moreover, for any $\alpha \in \mathcal{X} \cup \mathcal{M}$, if $\alpha \in \mathcal{X}$ then $\widehat{\sigma}(\alpha) \geq \frac{1-d}{|\mathcal{X}|}$, otherwise $\widehat{\sigma}(\alpha) > 0$.*

Proof. By Def. 6, $\widehat{\sigma}(\alpha)$ is the sum of $\widehat{\tau}(\alpha)$ and positive values. Hence if $\alpha \in \mathcal{X}$ then $\widehat{\sigma}(\alpha) \geq \frac{1-d}{|\mathcal{X}|} > 0$. Otherwise, if $\alpha \in \mathcal{M}$ then, by Defs. 5 and 6, $\widehat{\sigma}(\alpha) = \sqrt{d} \cdot \frac{\sum_{\beta \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\alpha)|+1} \geq \sqrt{d} \cdot \sum_{\beta \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\beta)$, and since $\beta \in \mathcal{X}$ then $\widehat{\sigma}(\beta) > 0 \quad \forall \beta$, hence $\widehat{\sigma}(\alpha) > 0$. By Theorem 1 and by Lem. 1, we have that if $\alpha \in \mathcal{X}$ then $\widehat{\sigma}(\alpha) \leq 1$. Otherwise, if $\alpha \in \mathcal{M}$ then, by Rem. 3, $\widehat{\mathcal{R}}^+(\alpha) = \{\beta\}$ and $\beta \in \mathcal{X}$, hence by Def. 6, $\widehat{\sigma}(\alpha) = \sqrt{d} \cdot \frac{\widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\alpha)|+1} \leq 1$. □

The next proposition sheds light on the intuition behind our MPRAFs, in that the support from non-meta-arguments is partitioned among the meta-arguments. Meta-arguments supported by the same ‘regular’ argument all have the same strength since according to the random surfer model the probability of clicking on links is uniform.

Proposition 5. *In an MPRAF $\langle \mathcal{X} \cup \mathcal{M}, \widehat{\mathcal{R}}^-, \widehat{\mathcal{R}}^+, \widehat{\tau} \rangle$, if a meta-argument $\alpha \in \mathcal{M}$ has attackers then $\widehat{\sigma}(\alpha) = \widehat{\sigma}(\gamma)$, $\forall \gamma \in \widehat{\mathcal{R}}^-(\alpha)$.*

Proof. By Def. 5, $\forall \gamma \in \widehat{\mathcal{R}}^-(\alpha) \quad \gamma \in \mathcal{M}$ and by Def. 5 and Rem. 3 $\forall \gamma \in \widehat{\mathcal{R}}^-(\alpha) \quad \widehat{\mathcal{R}}^+(\alpha) = \widehat{\mathcal{R}}^+(\gamma) = \{\beta\}$ where $\beta \in \mathcal{X}$ is the single supporter of α. By Def. 6, $\widehat{\sigma}(\alpha) = \widehat{\tau}(\alpha) + \sqrt{d} \cdot \frac{\sum_{\beta \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\alpha)|+1}$, and by Def. 5 and Rem. 3, $\widehat{\sigma}(\alpha) = \sqrt{d} \cdot \frac{\widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\alpha)|+1}$, and the same is true for any $\gamma \in \widehat{\mathcal{R}}^-(\alpha)$: $\widehat{\sigma}(\gamma) = \sqrt{d} \cdot \frac{\widehat{\sigma}(\beta)}{|\widehat{\mathcal{R}}^-(\gamma)|+1}$. By construction α and the elements of $\widehat{\mathcal{R}}^-(\alpha)$ all attack each other, thus $|\widehat{\mathcal{R}}^-(\alpha)| = |\widehat{\mathcal{R}}^-(\gamma)| \ \forall \gamma \in \widehat{\mathcal{R}}^-(\alpha)$, and the result follows. □

We now assess this framework and semantics with respect to the desirable properties.

Proposition 6. *$\widehat{\sigma}$ satisfies GP1, GP4, GP5, GP6, GP8, GP9 and GP11.*

Proof. GP1: by Def. 6, if $\widehat{\mathcal{R}}^+(\alpha) = \emptyset$ and $\widehat{\mathcal{R}}^-(\alpha) = \emptyset$ then the second term of the sum is always 0, therefore $\sigma(\alpha) = \tau(\alpha)$. GP4 holds because the GP's preconditions cannot be verified: by Lem. 2, $\forall \alpha \in \mathcal{X} \quad \widehat{\sigma}(\alpha) \geq \widehat{\tau}(\alpha)$. GP5: by Def. 6, $\widehat{\sigma}(\alpha) > \widehat{\tau}(\alpha)$ iff $\sum_{\beta \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\beta) > 0$. Thus, it must be the case that $\exists \beta \in \widehat{\mathcal{R}}^+(\alpha) : \widehat{\sigma}(\beta) > 0$, therefore $\widehat{\mathcal{R}}^+(\alpha) \neq \emptyset$ GP6: follows directly from Def. 6. GP8: if $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\beta)$ then $|\widehat{\mathcal{R}^-_\sigma}(\alpha)| = |\widehat{\mathcal{R}^-_\sigma}(\beta)|$ and if $\widehat{\mathcal{R}^+_\sigma}(\alpha) \subsetneq \widehat{\mathcal{R}^+_\sigma}(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) < \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. The result follows from Def. 6. GP9: if $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\beta)$ then $|\widehat{\mathcal{R}^-_\sigma}(\alpha)| = |\widehat{\mathcal{R}^-_\sigma}(\beta)|$ and if $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^+(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) = \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. The result follows from Def. 6. GP11: if $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\beta)$ then $|\widehat{\mathcal{R}^-_\sigma}(\alpha)| = |\widehat{\mathcal{R}^-_\sigma}(\beta)|$ and if $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{>} \widehat{\mathcal{R}}^+(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) > \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. The result follows from Def. 6. □

Proposition 7. *$\langle \widehat{PR}, \widehat{\sigma} \rangle$ is (not strictly) balanced and thus satisfies GP1 to GP3.*

Proof. Point 1: (A) If $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^+(\alpha) = \emptyset$ then the result follows by Def. 6. (B) Otherwise, if $\widehat{\mathcal{R}}^-(\alpha) \neq \emptyset$ then $\alpha \in \mathcal{M}$ and thus it has a single supporter β. There are two possible scenarios (B.i) $\exists! \gamma \in \mathcal{M} : (\beta, \alpha), (\beta, \gamma) \in \widehat{\mathcal{R}}^+$, then $\{\beta\} = \widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{>} \widehat{\mathcal{R}}^-(\alpha) = \{\gamma\}$ (which contradicts the hypothesis) because by Def. 6 $\widehat{\sigma}(\alpha) = \widehat{\sigma}(\gamma) < \widehat{\sigma}(\beta)$ (B.ii) $\exists_{>1} \gamma_1, ..., \gamma_n \in \mathcal{M} : (\beta, \alpha), (\beta, \gamma_1), ..., (\beta, \gamma_n) \in \widehat{\mathcal{R}}^+$, hence $|\widehat{\mathcal{R}}^-(\alpha)| > 1$, therefore it cannot hold that $\{\gamma_1, ..., \gamma_n\} = \widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^+(\alpha) = \{\beta\}$ (which contradicts the hypothesis), and by Def. 6 it holds again $\widehat{\sigma}(\alpha) = \widehat{\sigma}(\gamma_1) = ... = \widehat{\sigma}(\gamma_n) < \widehat{\sigma}(\beta)$, hence it cannot exists any injective mapping $f : \widehat{\mathcal{R}}^-(\alpha) \to \widehat{\mathcal{R}}^+(\alpha) : \forall \alpha \in \widehat{\mathcal{R}}^-(\alpha), \sigma(f(\alpha)) \geq \sigma(\alpha)$, and thus there is no strength-equivalency relationship between $\widehat{\mathcal{R}}^-(\alpha)$ and $\widehat{\mathcal{R}}^+(\alpha)$. Point 2. For $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{>} \widehat{\mathcal{R}}^+(\alpha)$ to hold $\widehat{\mathcal{R}}^-(\alpha) \neq \emptyset$, thus $\alpha \in \mathcal{M}$. Hence, we are in the same situation of (B) in the proof of Point 1, and therefore the precondition cannot hold and the result follows. Point 3. By Lem. 2, $\widehat{\sigma}(\alpha) > 0$ and if $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{<} \widehat{\mathcal{R}}^+(\alpha)$ then $\widehat{\mathcal{R}}^+(\alpha) \neq \emptyset$. Hence by Def. 6, $\widehat{\sigma}(\alpha) > \widehat{\tau}(\alpha)$. Point 4 holds because $\nexists \alpha : \widehat{\sigma}(\alpha) < \widehat{\tau}(\alpha)$. Point 5 does not hold. For example, consider the framework in Figure 2.b and in particular $m_{\alpha,\gamma} \in \mathcal{M}$ that it is supported by $\alpha \in \mathcal{X}$ and attacked by $m_{\alpha,\beta}, m_{\alpha,\delta} \in \mathcal{M}$. By Def. 5 and Lem. 2, we have that $\widehat{\sigma}(m_{\alpha,\gamma}) \leq \widehat{\sigma}(\alpha)$ and $\widehat{\sigma}(m_{\alpha,\gamma}) = \widehat{\sigma}(m_{\alpha,\beta}) = \widehat{\sigma}(m_{\alpha,\delta}) > 0$. Hence, $\widehat{\sigma}(m_{\alpha,\gamma}) > \widehat{\tau}(m_{\alpha,\gamma})$, but $\widehat{\mathcal{R}}^+(m_{\alpha,\gamma}) \overset{\sigma}{\ngeq} \widehat{\mathcal{R}}^-(m_{\alpha,\gamma})$ because no injective mapping exists from $\widehat{\mathcal{R}}^-(m_{\alpha,\gamma})$ to $\widehat{\mathcal{R}}^+(m_{\alpha,\gamma})$. Thus $\widehat{\mathcal{R}}^+(m_{\alpha,\gamma}) \overset{\sigma}{\ngtr} \widehat{\mathcal{R}}^-(m_{\alpha,\gamma})$.
□

Proposition 8. *$\langle \widehat{PR}, \widehat{\sigma} \rangle$ is strictly monotonic and thus satisfies GP6 to GP11.*

Proof. Point 1: if $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\beta)$ then $|\widehat{\mathcal{R}}^-(\alpha)| = |\widehat{\mathcal{R}}^-(\beta)|$ and if $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^+(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) = \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. The result follows from Def. 6. Point 3: if $\alpha, \beta \in \mathcal{X}$ then $\widehat{\tau}(\alpha) = \widehat{\tau}(\beta)$ and $\widehat{\mathcal{R}}^-(\beta) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\alpha) = \emptyset$, hence $|\widehat{\mathcal{R}}^-(\alpha)| = |\widehat{\mathcal{R}}^-(\beta)|$. If $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{<} \widehat{\mathcal{R}}^+(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) < \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. Thus, by Def. 6, $\widehat{\sigma}(\alpha) < \widehat{\sigma}(\beta)$. If $\alpha \in \mathcal{M}$ and $\beta \in \mathcal{X}$ then $\widehat{\tau}(\alpha) < \widehat{\tau}(\beta)$ and $\widehat{\mathcal{R}}^-(\beta) = \emptyset$. If $\widehat{\mathcal{R}}^-(\alpha) \overset{\sigma}{\geq} \emptyset$ then $|\widehat{\mathcal{R}}^-(\alpha)| \geq |\widehat{\mathcal{R}}^-(\beta)| = 0$. If $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{\leq} \widehat{\mathcal{R}}^+(\beta)$ then

Table 1. GPs and principles (**B**alance, **S**trict **B**alance, **M**onotonicity, **S**trict **M**onotonicity) satisfied by σ, $\langle PR, \sigma\rangle$, $\widehat{\sigma}$ and $\langle \widehat{PR}, \widehat{\sigma}\rangle$, where ✓ and × denote property satisfied and not satisfied, resp.

	GP1	GP2	GP3	GP4	GP5	GP6	GP7	GP8	GP9	GP10	GP11	B	SB	M	SM
σ	✓	×	✓	✓	✓	×	×	✓	✓	×	×	×	×	×	×
$\langle PR, \sigma\rangle$	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	✓	✓	×	×
$\widehat{\sigma}$	✓	×	×	✓	✓	✓	×	✓	✓	×	✓	×	×	×	×
$\langle \widehat{PR}, \widehat{\sigma}\rangle$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓

$\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) \leq \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. Thus, by Def. 6, $\widehat{\sigma}(\alpha) < \widehat{\sigma}(\beta)$. If $\alpha, \beta \in \mathcal{M}$ then $\widehat{\tau}(\alpha) = \widehat{\tau}(\beta)$. If $\widehat{\mathcal{R}}^-(\beta) \overset{\sigma}{\leq} \widehat{\mathcal{R}}^-(\alpha)$ then $|\widehat{\mathcal{R}}^-(\alpha)| \geq |\widehat{\mathcal{R}}^-(\beta)|$. If $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{\leq} \widehat{\mathcal{R}}^+(\beta)$ then $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) \leq \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$. Hence, by Def. 6, $\widehat{\sigma}(\alpha) \leq \widehat{\sigma}(\beta)$. For $\mathcal{ST}(\beta) \not\leq \mathcal{ST}(\alpha)$ to hold, either:

- $\widehat{\mathcal{R}}^-(\beta) \overset{\sigma}{<} \widehat{\mathcal{R}}^-(\alpha)$ and $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{=} \widehat{\mathcal{R}}^+(\beta)$, or
- $\widehat{\mathcal{R}}^-(\beta) \overset{\sigma}{=} \widehat{\mathcal{R}}^-(\alpha)$ and $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{<} \widehat{\mathcal{R}}^+(\beta)$, or
- $\widehat{\mathcal{R}}^-(\beta) \overset{\sigma}{<} \widehat{\mathcal{R}}^-(\alpha)$ and $\widehat{\mathcal{R}}^+(\alpha) \overset{\sigma}{<} \widehat{\mathcal{R}}^+(\beta)$.

In the first case, by construction of the framework PR, $|\widehat{\mathcal{R}}^-(\alpha)| < |\widehat{\mathcal{R}}^-(\beta)|$, thus $\widehat{\sigma}(\alpha) < \widehat{\sigma}(\beta)$. In the second case, $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) < \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$, thus $\widehat{\sigma}(\alpha) < \widehat{\sigma}(\beta)$. In the third case, $\sum_{\gamma \in \widehat{\mathcal{R}}^+(\alpha)} \widehat{\sigma}(\gamma) < \sum_{\gamma \in \widehat{\mathcal{R}}^+(\beta)} \widehat{\sigma}(\gamma)$ and $|\widehat{\mathcal{R}}^-(\alpha)| \leq |\widehat{\mathcal{R}}^-(\beta)|$, thus $\widehat{\sigma}(\alpha) < \widehat{\sigma}(\beta)$. Point 3 implies Point 2, thus the result follows. □

We have thus proven that, through MPRAF, in exchange for a little structural addition, it is possible to ensure equivalence with PR while at the same time satisfying more desirable properties from an argumentation semantics perspective. The value of the proposed approach is not purely theoretical however, as we discuss in next section.

5. Towards better PR explanations

As shown in Table 1, the M-PR semantics applied on an MPRAF satisfies almost all the desirable properties outlined in Section 3, including in particular *monotonicity*. This means that, from a dialectical viewpoint, the strength of an argument depends exclusively on its intrinsic strength, the reasons supporting it and the reasons against it, and any strengthening/weakening of these will affect the argument's strength intuitively. The satisfaction of monotonicity is achieved through the role ascribed to meta-arguments and is a key factor for exploiting MPRAFs for practical application, such as the generation of explanations of the PR score of a page. In this scenario, *monotonicity* is clearly a crucial factor because it allows a user to identify direct dependencies between the variations of the strength of arguments according to the attacks and supports linking them in the graph structure of the MPRAF. For this reason, MPRAFs are able to provide the end user a better understanding of the factors determining the PR score of a page, i.e. they support answering questions like "Which incoming links (and thus pages) contribute the most to the score of this page?".

To provide some preliminary empirical support to this claim, we ran some experiments on the Wikipedia dataset from Wikipedia Dumps consisting of 965,748 pages and 7,388,700 links, with an average link density of 7.65 links per page. While discussing more extensively our experiments is beyond the scope of this paper, we provide a concrete example of the explanatory advantages achievable with MPRAFs.

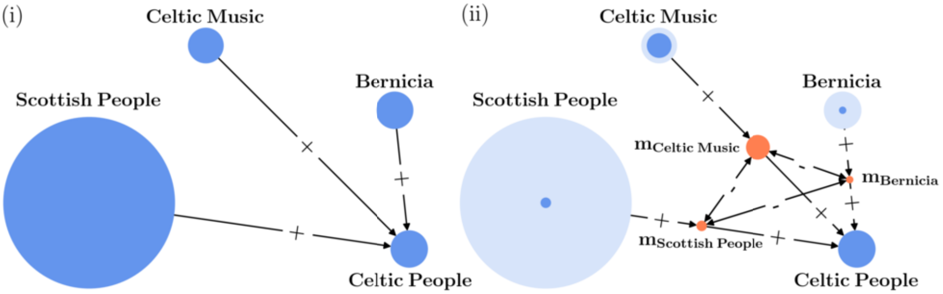


Figure 3. Excerpt of the PRAF (i) and MPRAF (ii) for the article *Celtic people* in the *simple* version of *Wikipedia* including the article and its direct supporters. Each bubble represents an argument and its size is proportional the the strength of the argument. In (ii) the opaque bubbles highlight the actual contribution of an argument to the *Celtic People* page, derived from the strengths of the corresponding meta-arguments.

Consider first Fig. 3.i, showing a magnification of the weighted view of the pages contributing to the score of the article *Celtic People*, with each page score represented by the size of the relevant bubble. Looking at this figure a user might (erroneously) deduce that the score of *Celtic People* is mostly determined by *Scottish People*, which is actually not the case (due to the high number of outgoing links from *Scottish People*). To realize this a user should both have a deeper understanding of PR's functioning and be shown a larger part of the graph, including all the pages linked by *Celtic People*'s supporters.

This undesirable overload is avoided by the MPRAF-based representation in Figure 3.ii. Here the meta-arguments show directly the actual support flowing from the supporters, and the user can appreciate that *Celtic Music* is the article providing most support to *Celtic People*. Besides better supporting direct explanations, the MPRAF-based representation appears to enable answering other kinds of user queries, like counterfactual questions of the kind: 'What would happen if a given link is suppressed?' A wider investigation on MPRAF-based explanations for PR outcomes is planned for future work.

6. Conclusions

Towards the more general goal of investigating connections between PageRank and argument evaluation, we have introduced a novel approach capable of reconstructing PR as a gradual argumentation semantics of a suitably defined bipolar argumentation framework, while ensuring the satisfaction of a set of generally desirable properties. We have then given an example of the practical yields of this theoretical achievement, concerning the generation of better explanations of PR scores to end users.

To the best of our knowledge, the investigation of the relationships between PR and argumentation semantics has not been previously considered in the literature. The work in [15] explores the application of PR to rank the relevance of arguments available on the web to support or attack a given stance. This is an interesting but different goal: in [15] PR is not related to any semantics notion and the links have a different meaning, relating the conclusion of an argument with the premises of another one. On a different but related line, some works, e.g. [16], have explored connections between argumentation semantics and matrix representations from network theory, whose relationships with our approach are worth future investigation.

Our proposal can be extended in several directions. On one hand, the investigation of PR-inspired gradual semantics for various kinds of argumentation frameworks could be pursued. In this respect it would be interesting to consider *weighted* versions of PR where a node strength can be distributed unevenly to its children and more generally the variants of PR considered in various domains [6]. On the other hand, one can notice that PR is essentially a mechanism to produce a score based on a relation of support, but it could be considered that in several domains where PR is applied, also other relations, in particular attack could be relevant for a proper scoring. Also, in the web domain, one could argue that the absence of a link from one page to another (where this link could instead be expected according to some criterion) could be interpreted as an attack diminishing the relevance of the non-linked page. Given the strong tradition on attack-based and bipolar evaluations in argumentation semantics, this suggests that the study of argumentation-inspired variants of PR may also represent a fruitful research direction.

References

[1] Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet And Web Information Systems. 1998;54(1999-66):1–17.

[2] Ma N, Guan J, Zhao Y. Bringing PageRank to the citation analysis. Information Processing and Management. 2008 3;44(2):800–810.

[3] Gori M, Pucci A. ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines. In: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI); 2007. p. 2766–2771.

[4] Hudelson M, Mooney BL, Clark AE. Determining polyhedral arrangements of atoms using PageRank. Journal of Mathematical Chemistry. 2012 9;50(9):2342–2350.

[5] Morrison JL, Breitling R, Higham DJ, Gilbert DR. GeneRank: Using search engine technology for the analysis of microarray experiments. BMC Bioinformatics. 2005 sep;6(1):233.

[6] Gleich DF. PageRank beyond the web. SIAM Review. 2015;57(3):321–363.

[7] Altman A, Tennenholtz M. Ranking systems: the PageRank axioms. In: Proc. 6th ACM Conf. on Electronic Commerce (EC); 2005. p. 1–8.

[8] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321–358.

[9] Cayrol C, Lagasquie-Schiex MC. On the Acceptability of Arguments in Bipolar Argumentation Frameworks. In: Proc. of the 8th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU); 2005. p. 378–389.

[10] Baroni P, Rago A, Toni F. From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. Int Journal Approximate Reasoning. 2019;105:252–286.

[11] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. Knowledge Engineering Review. 2011;26(4):365–410.

[12] Cayrol C, Lagasquie-Schiex MC. Graduality in Argumentation. Journal of Artificial Intelligence Research. 2005;23:245–297.

[13] Amgoud L, Ben-Naim J. Evaluation of Arguments from Support Relations: Axioms and Semantics. In: Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI); 2016. p. 900–906.

[14] Langville AN, Meyer CD. Deeper inside PageRank. Internet Mathematics. Internet Mathematics. 2004;1(3):335–380.

[15] Wachsmuth H, Stein B, Ajjour Y. "PageRank" for Argument Relevance. In: Proc. of the 15th Conf. of the European Chpt. of the Association for Computational Linguistics (EACL); 2017. p. 1117–1127.

[16] Corea C, Thimm M. Using Matrix Exponentials for Abstract Argumentation. In: Proc. of the 1st Int. Workshop on Systems and Algorithms for Formal Argumentation (SAFA); 2016. p. 10–21.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200493

Computing Skeptical Preferred Acceptance in Dynamic Argumentation Frameworks with Recursive Attack and Support Relations

Gianvincenzo ALFANO [1], Sergio GRECO and Francesco PARISI
DIMES Department, University of Calabria, Italy

Abstract. Attack-Support Argumentation Framework (ASAF) is an extension of the Bipolar Argumentation Framework that allows for attacks and supports not only between arguments but also targeting attacks and supports at any level. In this paper we propose an incremental approach for computing the skeptical preferred acceptance in dynamic ASAFs. Specifically, we investigate how the skeptical acceptance of a goal element (an argument, an attack, or a support) evolves when a given ASAF is updated by adding or retracting an argument, an attack, or a support, and propose an incremental algorithm for solving this problem. Our approach relies on identifying a portion of the given ASAF which is sufficient to determine the status of the goal w.r.t. the updated ASAF. We experimentally evaluate our approach showing that it outperforms the computation from scratch on average.

Keywords. Abstract argumentation, Higher-order interactions, Incremental computation

1. Introduction

Formal argumentation has emerged as one of the important fields in Artificial Intelligence [17,37,11]. In particular, Dung's framework is a simple, yet powerful formalism for modelling disputes between two or more agents [29]. An abstract Argumentation Framework (AF) consists of a set of *arguments* and a binary *attack* relation over the set of arguments that specifies the *interactions* between arguments: intuitively, if argument a attacks argument b, then b is acceptable only if a is not. Hence, arguments are abstract entities whose role is entirely determined by the interactions specified by the attack relation.

Dung's framework has been extended in many different ways, including the introduction of new kinds of interactions between arguments and/or attacks. In particular, the Bipolar Argumentation Framework is an interesting extension of the Dung's framework which allows for modelling the *support* between arguments [9,27,39]. Further extensions consider second-order interactions [20], e.g., attacks to attacks/supports, as well as more general forms of interactions such as Argumentation Frameworks with Recursive Attacks

[1] Corresponding Author: Gianvincenzo Alfano, e-mail: g.alfano dimes.unical.it

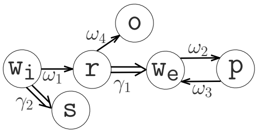


Figure 1. ASAF of Example 1

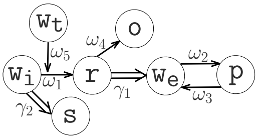


Figure 2. ASAF for winter scenario

[13] and Attack-Support Argumentation Frameworks (ASAFs) [31], where attacks and supports can be recursively attacked or supported.

Example 1. *Consider a scenario to decide whether to play tennis. Assume we have the following arguments:* $\mathtt{w_i}$ *(it is windy),* $\mathtt{r}$ *(it rains),* $\mathtt{w_e}$ *(the court is wet),* $\mathtt{p}$ *(play tennis),* $\mathtt{s}$ *(need a sweatshirt),* $\mathtt{o}$ *(tennis racket shop is open), and the implications: (ω_1) if it is windy, then it does not rain, (ω_2) if the court is wet, then we cannot play tennis, (ω_3) if we play tennis then the court is not wet, (ω_4) if it rains then the tennis racket shop is not open, (γ_1) if it rains, then the court is wet, and (γ_2) if it is windy, then we need a sweatshirt. This situation can be modeled by using the ASAF of Figure 1, where ω_1, ω_2 and ω_3 are* attacks *(denoted by $\rightarrow$), and γ_1 and γ_2 are* supports *(denoted by $\Rightarrow$).* □

Several interpretations of the notion of support have been proposed [25,27]. The *necessary* support [13,36] adopted in ASAF is intended to capture the following intuition: if a supports b, then the acceptance of a is necessary to get the acceptance of b; equivalently, accepting b implies accepting a. The meaning of an ASAF is given by extensions which also include attacks and supports that contribute to determine the set of accepted arguments. For instance, considering the well-known *preferred* semantics—one of the most popular argumentation semantics [22]—the framework of Figure 1 has a unique extension, that is the set $\{\mathtt{w_i}, \mathtt{s}, \mathtt{p}, \mathtt{o}, \omega_1, \omega_3, \gamma_1, \gamma_2\}$.

However, in practice, argumentation frameworks can be dynamic systems [12,15,16,18,26,34]. In fact, typically an ASAF represents a temporary situation, and new arguments, attacks and supports (at any level) can be added/removed to take into account new available knowledge. For instance, in our running example, assume now that there exists also an argument $\mathtt{w_t}$ (we are in the winter season) that attacks ω_1 (in the winter season ω_1 cannot be applied). The updated scenario can be modeled by an ASAF shown in Figure 2 where the new attack is labelled as ω_5 (ω_5 is an example of second-level attack).

Recently, there has been a growing interest in studying dynamics of different argumentation systems, considering the Dung framework [2,5,15,19,28], Bipolar AF and AF with second order attacks [3,4], ASAF [1], and structured argumentation formalisms [7,8]. This is motivated by the fact that most of the argumentation problems have high computational complexity [30,33]. In particular, skeptical reasoning under the preferred semantics is in the second level of the polynomial hierarchy. However, in practice, incremental computation techniques could improve performance, as they only require to reconsider the acceptance status of those arguments and interactions that are affected by the new information.

In this paper we propose an incremental approach for computing the skeptical preferred acceptance of a goal element of an ASAF after performing an update. Specifically, we propose a technique addressing the following problem: given an ASAF Δ, a goal element G whose skeptical preferred acceptance w.r.t. Δ is known, and an update u consisting of the addition/removal of an argument/attack/support, decide whether G is skeptically preferred accepted w.r.t. the updated ASAF $u(\Delta)$, that is, decide if G belongs to every preferred extension of $u(\Delta)$.

Contributions. We make the following contributions:

- Given an update and a goal element (an argument, an attack, or a support), we identify a set of elements, called *alterable set*, which contains the elements whose acceptance status may change after the update and propagate up to the goal.
- Given the alterable set, we define the *Proxy ASAF* that allows us to compute the skeptical preferred acceptance of a goal by focusing on a (potentially smaller) ASAF containing the alterable set as well as additional elements and interactions needed to determine the status of the elements in the alterable set.
- We introduce an incremental algorithm for computing the skeptical preferred acceptance of a goal within a dynamic ASAF. It enables the computation on the Proxy ASAF, provided that an external solver for ASAFs is given.
- Since to the best of our knowledge there is no available solver for the direct computation on ASAF, we propose a version of the algorithm that, using a translation of our problem to the AF domain, allows us to use any (non-incremental) state-of-the-art AF solver to compute the skeptical preferred acceptance for ASAFs
- We provide an experimental analysis showing the effectiveness of our approach.

To the best of our knowledge, this is the first paper addressing the problem of efficiently and incrementally computing skeptical acceptance for dynamic ASAFs.

2. Preliminaries

We start by briefly reviewing the Dung's framework [29] and the Attack-Support Argumentation Framework (ASAF) (for a full presentation of ASAF see [31]).

An abstract *Argumentation Framework* (AF) is a pair $\langle \mathbb{A}, \Sigma \rangle$, where $\mathbb{A}$ is a set of *arguments* and $\Sigma \subseteq \mathbb{A} \times \mathbb{A}$ is a set of *attacks*. An AF can be seen as a directed graph, whose nodes represent arguments and edges represent attacks.

Given an AF $\Lambda = \langle \mathbb{A}, \Sigma \rangle$ and a set $S \subseteq \mathbb{A}$ of arguments, an argument $a \in \mathbb{A}$ is said to be *i*) *attacked* (or, equivalently, *defeated*) w.r.t. S iff $\exists b \in S$ such that $(b, a) \in \Sigma$, and *ii*) *acceptable* w.r.t. S iff for every argument $b \in \mathbb{A}$ with $(b, a) \in \Sigma$, there is $c \in S$ such that $(c, b) \in \Sigma$. The sets of defeated and acceptable arguments w.r.t. S can be defined as follows (where Λ is understood):

- $Def(S) = \{a \in \mathbb{A} \mid \exists\, b \in S \,.\, (b, a) \in \Sigma\}$;
- $Acc(S) = \{a \in \mathbb{A} \mid \forall\, b \in \mathbb{A} \,.\, (b, a) \in \Sigma \;\Rightarrow\; b \in Def(S)\}$.

Given an AF $\langle \mathbb{A}, \Sigma \rangle$, a set $S \subseteq \mathbb{A}$ of arguments is said to be: (*i*) *conflict-free* iff $S \cap Def(S) = \emptyset$; (*ii*) *admissible* iff it is conflict-free and $S \subseteq Acc(S)$. Moreover, $S \subseteq \mathbb{A}$ is said to be a *preferred extension* iff it is conflict-free, $S = Acc(S)$, and maximal (w.r.t. $\subseteq$). The set of preferred extensions of an AF Λ will be denoted by $\mathcal{PR}(\Lambda)$.

Example 2. *Let* $\Lambda = \langle \mathbb{A}, \Sigma \rangle$ *be an AF where* $\mathbb{A} = \{\mathtt{r}, \mathtt{w_e}, \mathtt{p}\}$ *and* $\Sigma = \{(\mathtt{w_e}, \mathtt{p}), (\mathtt{p}, \mathtt{w_e})\}$. *The set of preferred extensions is* $\mathcal{PR}(\Lambda) = \{\{\mathtt{r}, \mathtt{w_e}\}, \{\mathtt{r}, \mathtt{p}\}\}$. □

Given an AF $\Lambda = \langle \mathbb{A}, \Sigma \rangle$ and an argument $G \in \mathbb{A}$, we say that G is skeptically preferred accepted w.r.t. Λ iff for each preferred extension E of Λ it holds that $G \in E$. For instance, for the AF in Example 2, we have that $\mathtt{r}$ is skeptically preferred accepted.

Attack-Support Argumentation Framework

An *Attack-Support Argumentation Framework (ASAF)* [31] ASAF is a triple $\langle A, \Omega, \Gamma \rangle$, where A is a set of arguments, $\Omega \subseteq A \times (A \cup \Omega \cup \Gamma)$ is a set of attacks, and $\Gamma \subseteq A \times (A \cup \Omega \cup \Gamma)$ is a set of supports. It is assumed that Γ is acyclic and $\Omega \cap \Gamma = \emptyset$.

In the following, given an ASAF $\langle A, \Omega, \Gamma \rangle$, to simplify the notation, we use symbols ω_i (or simply ω) to denote attacks (e.g., $\omega = (a, X) \in \Omega$) and symbols γ_i (or simply γ) to denote supports (e.g., $\gamma = (b, Y) \in \Gamma$); we also use δ to denote an element in $\Omega \cup \Gamma$. Moreover, given an ASAF $\langle A, \Omega, \Gamma \rangle$, for any attack or support $\delta = (a, Y) \in \Omega \cup \Gamma$, we use $\mathtt{s}(\delta)$ and $\mathtt{t}(\delta)$ to denote, respectively, the source argument a and the target element Y of δ. Note that Y can be an argument, an attack, or a support. Attacks and supports whose target is an argument are said to be *first-level* interactions, while attacks and supports whose target is an interaction of level i are said to be interactions of level $i + 1$.

An ASAF Δ can be represented by a graph-like structure $\mathcal{G}_\Delta$ where an argument $a \in A$ is a node in $\mathcal{G}_\Delta$, an attack $\omega = (a, X) \in \Omega$ is graphically denoted as an edge $a \xrightarrow{\omega} X$ in $\mathcal{G}_\Delta$, and a support $\gamma = (b, Y) \in \Gamma$ is graphically denoted as an edge $b \overset{\gamma}{\Longrightarrow} Y$ in $\mathcal{G}_\Delta$. For instance, the graph in Figure 1 represents the ASAF of Example 1, that is, an ASAF $\Delta = \langle A, \Omega, \Gamma \rangle$, where $A = \{\mathtt{w_i}, \mathtt{r}, \mathtt{s}, \mathtt{o}, \mathtt{w_e}, \mathtt{p}\}$, $\Omega = \{\omega_1 = (\mathtt{w_i}, \mathtt{r}), \omega_2 = (\mathtt{w_e}, \mathtt{p}), \omega_3 = (\mathtt{p}, \mathtt{w_e}), \omega_4 = (\mathtt{r}, \mathtt{o})\}$, and $\Gamma = \{\gamma_1 = (\mathtt{r}, \mathtt{w_e}), \omega_2 = (\mathtt{w_i}, \mathtt{s})\}$.

Attacks and supports in an ASAF can also be attacked and supported, and extensions may contain arguments, attacks and supports. The semantics proposed in [31] combines the interpretation of attacks of Argumentation Frameworks with Recursive Attacks [13] with that of Bipolar AFs with necessary support [25], as formalized in what follows.

Given an ASAF $\langle A, \Omega, \Gamma \rangle$, a *support path* $a_0 \overset{+}{\Rightarrow} X$ from a_0 to X is is a sequence of n supports $a_0 \overset{\gamma_1}{\Longrightarrow} a_1 \overset{\gamma_2}{\Longrightarrow} \dots a_{n-1} \overset{\gamma_n}{\Longrightarrow} X$, where each a_i (with $0 \leq i < n$) is an argument and X is either an argument, an attack, or a support. We use $\Gamma^+ = \{(a, X) \mid a \in A,\ X \in (A \cup \Omega \cup \Gamma),\ a \overset{+}{\Rightarrow} X\}$ to denote the set of pairs (a, X) such that there exists a (not empty) support path from a to X.

Given an element $X \in (A \cup \Omega \cup \Gamma)$ and an attack $\omega \in \Omega$, we say that ω *(directly or indirectly) attacks* X (denoted by $\omega\ \mathtt{def}\ X$) if either $\mathtt{t}(\omega) = X$ or $\mathtt{t}(\omega) = \mathtt{s}(X)$. Moreover, given a set $S \subseteq A \cup \Omega \cup \Gamma$, we say that ω *extendedly defeats* X *given* S (denoted as $\omega\ \mathtt{def_S}\ X$) if either $\omega\ \mathtt{def}\ X$ or there exists $b \in A$ such that $\mathtt{t}(\omega) = b$ and either $(b, X) \in (\Gamma \cap S)^+$ or $(b, s(X)) \in (\Gamma \cap S)^+$. For any ASAF Δ and $S \subseteq A \cup \Omega \cup \Gamma$, the *defeated* and *acceptable* sets (given S) are:

- $Def(S) = \{X \in A \cup \Omega \cup \Gamma \mid \exists\, \omega \in \Omega \cap S \,.\, \omega\ \mathtt{def_S}\ X\}$
- $Acc(S) = \{X \in A \cup \Omega \cup \Gamma \mid \forall \omega \in \Omega \,.\, \omega\ \mathtt{def_S} X \Rightarrow \omega \in Def(S)\}$.

The notions of *conflict-free*, *admissible sets*, and the *preferred extensions* for ASAF can be defined as done earlier (before Example 2) for the AF but considering $S \subseteq A \cup \Omega \cup \Gamma$ and by using the definitions of defeated and acceptable sets reported above.

Finally, given an ASAF $\Delta = \langle A, \Omega, \Gamma \rangle$ and an element $G \in A \cup \Omega \cup \Gamma$, we say that G is skeptically preferred accepted w.r.t. Δ iff for each preferred extension E of Δ it holds that $G \in E$. In the following, we use $SA_\Delta(G)$ to denote the skeptical preferred acceptance (either *true* or *false*) of G w.r.t. ASAF Δ.

Example 3. *Let* $\Delta = \langle \{\mathtt{w_i}, \mathtt{r}, \mathtt{s}, \mathtt{o}, \mathtt{w_e}, \mathtt{p}, \mathtt{w_t}\}, \{\omega_1{=}(\mathtt{w_i}, \mathtt{r}), \omega_2 = (\mathtt{w_e}, \mathtt{p}), \omega_3 = (\mathtt{p}, \mathtt{w_e}), \omega_4 = (\mathtt{r}, \mathtt{o}), \omega_5 = (\mathtt{w_t}, (\mathtt{w_i}, \mathtt{r}))\}, \{\gamma_1 = (\mathtt{r}, \mathtt{w_e}), \gamma_2 = (\mathtt{w_i}, \mathtt{s})\}\rangle$ *be the ASAF of Figure 2. The set of preferred extensions of* Δ *is* $\mathcal{PR}(\Delta) = \{\{\mathtt{w_i}, \mathtt{r}, \gamma_1, \mathtt{s},\ \mathtt{w_e}, \omega_2,\ \mathtt{w_t}, \omega_4,\ \omega_5,\ \gamma_2\},$

$\{\mathtt{w_i}, \mathtt{r}, \gamma_1, \mathtt{s}, \mathtt{p}, \omega_3, \mathtt{w_t}, \omega_4, \omega_5, \gamma_2\}\}$. *Thus, the set of elements X of Δ that are skeptically accepted (i.e., those for which $SA_\Delta(X) = true$) is* $\{\mathtt{w_i}, \mathtt{r}, \gamma_1, \mathtt{s}, \mathtt{w_t}, \omega_4, \omega_5, \gamma_2\}$. □

Updates for ASAF

An *update* consists of the addition (resp., removal) of an attack or a support not present (resp., present) in a given ASAF, as next formalized.

Definition 1 (Update for ASAF)**.** *Let $\Delta = \langle A, \Omega, \Gamma\rangle$ an ASAF, and $\delta \in A \times (A \cup \Omega \cup \Gamma)$. An* update *$u$ over Δ is of one of the forms below, and when applied to Δ yields the updated ASAF $u(\Delta) = \langle A, \Omega', \Gamma'\rangle$, with Ω' and Γ' defined as follows:*

- *$u = +\delta$ where $\delta \notin (\Omega \cup \Gamma)$. If δ is an attack, then $\Omega' = \Omega \cup \{\delta\}$ and $\Gamma' = \Gamma$, otherwise $\Omega' = \Omega$, $\Gamma' = \Gamma \cup \{\delta\}$ and Γ' is acyclic.*
- *$u = -\delta$ where $\delta \in \Omega \cup \Gamma$ and there is no $\delta' \in \Omega \cup \Gamma$ such that $\mathbf{t}(\delta') = \delta$. In this case, $\Omega' = \Omega \setminus \{\delta\}$ and $\Gamma' = \Gamma \setminus \{\delta\}$.*

In the following, for simplicity, we write $\pm\delta$ for the addition or removal of an attack or a support $(\mathbf{s}(\delta), \mathbf{t}(\delta))$. Then, for an update $u = +\delta$, the interaction δ must not belong to the attack and support relations of the ASAF it will be applied on, and the source and target of δ must belong to the ASAF; moreover, the support relation of the updated ASAF must remain acyclic. Moreover, for an update $u = -\delta$, the interaction δ cannot be targeted by any other interaction in the ASAF.

As for an update u consisting of the addition (resp. deletion) of a set of *isolated* arguments (i.e., arguments not connected to any other element in the graph), it is easy to see that if $u(\Delta)$ is obtained from Δ through the addition (resp. deletion) of a set S of isolated argument, then every argument in S is trivially skeptically preferred accepted (resp., not accepted) w.r.t. $u(\Delta)$. Indeed, if E is an extension for Δ, then $E' = E \cup S$ (resp. $E' = E \setminus S$) is an extension for $u(\Delta)$ containing every (resp., none) argument in S. Of course, if arguments in S are not isolated, for addition we can first add isolated arguments and then add interactions (attacks or supports) involving these arguments, while for deletion we can first delete all interactions involving arguments in S and then delete isolated arguments. Thus we do not consider these kinds of updates in the following, and w.l.o.g. focus on updates consisting of the addition or deletion of an attack or a support.

3. Incremental Computation of Skeptical Preferred Acceptance

In this section, given an ASAF and an update for it, we propose an incremental technique for computing the skeptical preferred acceptance of a given goal element.

First we identify a set of *alterable* elements, that is, a set of arguments, attacks, and supports whose acceptance status may change after performing an update, and such that the change may impact on the acceptance status of the goal. We start by defining the set of elements that are reachable from a given element X of an ASAF. This set includes X and its *neighbors*, i.e. the target of X and, if X is an argument, also the interactions originating from X and the targets of such interactions, as formalized in what follows.

Definition 2 (Set of neighbors)**.** *Let $\Delta = \langle A, \Omega, \Gamma\rangle$ be an ASAF. The set $N_\Delta(X)$ of neighbors of an element $X \in A \cup \Omega \cup \Gamma$ is:*
i) $\{X, \mathbf{t}(X)\}$ *if* $X \in \Omega \cup \Gamma$, *ii)* $\{X\} \cup \{Y, \mathbf{t}(Y) \mid X = \mathbf{s}(Y),\ Y \in \Omega \cup \Gamma\}$ *if* $X \in A$.

For instance, for the ASAF Δ of Figure 2, we have that $N_\Delta(\mathtt{w_i}) = \{\mathtt{w_i}, \omega_1, \mathtt{r}, \gamma_2, \mathtt{s}\}$, $N_\Delta(\omega_5) = \{\omega_5, \omega_1\}$, and $N_\Delta(\omega_1) = \{\omega_1, \mathtt{r}\}$. The set of elements that are reachable from X consists of $N_\Delta(X)$ plus the elements which are reachable from $N_\Delta(X)$.

Definition 3 (Reachable elements). *Let $\Delta = \langle A, \Omega, \Gamma\rangle$ be an ASAF. Given $X, Y \in A \cup \Omega \cup \Gamma$ we say that Y is reachable from X in Δ iff either i) $Y \in N_\Delta(X)$ or ii) $\exists\, Z \in A \cup \Omega \cup \Gamma$ such that $Z \in N_\Delta(X)$ and Y is reachable from Z in Δ.*
We use $Reach_\Delta(X)$ to denote the set of elements of Δ that are reachable from X in Δ.

For the ASAF Δ of Figure 2, $Reach_\Delta(\omega_5) = \{\omega_5, \omega_1, \mathtt{r}, \omega_4, \gamma_1, \mathtt{o}, \mathtt{w_e}, \omega_2, \mathtt{p}, \omega_3\}$.

In the following, we use Δ^u to denote the larger ASAF between Δ and $u(\Delta)$, that is, Δ^u is *i)* the updated ASAF $u(\Delta)$ if u is an addition update (it includes the interaction added through u), *ii)* the original ASAF Δ if u is a deletion update (the removed interaction is also considered in Δ^u).

We are now ready to define the alterable set for an ASAF w.r.t. a given update.

Definition 4 (Alterable Set). *Let $\Delta = \langle A, \Omega, \Gamma\rangle$ be an ASAF, $u = \pm\delta$ an update, and $G \in A \cup \Omega \cup \Gamma$ a (goal) element. Let*

- $Alt_0(\Delta, u, G) = \begin{cases} \emptyset & if\ G \notin Reach_{\Delta^u}(\delta); \\ N_{\Delta^u}(\delta) & otherwise. \end{cases}$
- $Alt_{i+1}(\Delta, u, G) = Alt_i(\Delta, u, G) \cup \{Z \mid Z \in N_{\Delta^u}(Y),\ Y \in Alt_i(\Delta, u, G),\ G \in Reach_{\Delta^u}(Z)\}.$

Let n be the natural number such that $Alt_n(\Delta, u, G) = Alt_{n+1}(\Delta, u, G)$. Then alterable set $Alt(\Delta, u, G)$ is $Alt_n(\Delta, u, G)$.

Thus, the alterable set is iteratively defined by $n+1$ steps (with $n \leq |A| + |\Omega| + |\Gamma|$), each of them consisting of the addition of at least a neighbor of an element in the set built at the previous step and allowing to reach the goal G. It is easy to see that, for any element G, it is the case that $Alt(\Delta, u, G) \subseteq Reach_{\Delta^u}(\delta)$, where $u = \pm\delta$.

Example 4. *Consider the ASAF Δ of Figure 2, the update $u = -\omega_5$, and assume that $\mathtt{p}$ is the goal element. Note that, differently from the introduction, the update considered here is a deletion. Then, $Alt_0(\Delta, u, \mathtt{p}) = \{\omega_5, \omega_1\}$ as $\mathtt{p} \in Reach_{\Delta^u}(\omega_5)$. $Alt_1(\Delta, u, \mathtt{p}) = Alt_0(\Delta, u, \mathtt{p}) \cup \{\mathtt{r}\}$, $Alt_2(\Delta, u, \mathtt{p}) = Alt_1(\Delta, u, \mathtt{p}) \cup \{\gamma_1, \mathtt{w_e}\}$ (herein, ω_4 and $\mathtt{o}$ are not included as they do not allow to reach the goal in Δ^u), $Alt_3(\Delta, u, \mathtt{p}) = Alt_2(\Delta, u, \mathtt{p}) \cup \{\omega_2, \mathtt{p}\}$. Finally, $Alt_4(\Delta, u, \mathtt{p}) = Alt_3(\Delta, u, \mathtt{p}) \cup \{\omega_3\}$, and thus $Alt(\Delta, u, \mathtt{p}) = \{\omega_5, \omega_1, \mathtt{r}, \gamma_1, \mathtt{w_e}, \omega_2, \mathtt{p}, \omega_3\} \subseteq Reach_{\Delta^u}(\omega_5)$.*

The following theorem states that, after performing an update, the skeptical preferred acceptance of an element does not change if the alterable set is empty.

Theorem 1. *Let $\Delta = \langle A, \Omega, \Gamma\rangle$ be an ASAF, u an update, $u(\Delta)$ the updated ASAF, and G a goal element in $A \cup \Omega \cup \Gamma$. Therefore, if $Alt(\Delta, u, G) = \emptyset$ then $SA_{u(\Delta)}(G) = SA_\Delta(G)$.*

If the alterable set is not empty, we identify a (potentially small) portion of the given ASAF, called *Proxy ASAF*, that is sufficient to perform the computation of the skeptical preferred acceptance of the goal without considering the entire ASAF.

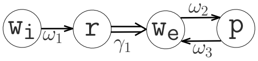


Figure 3. Proxy ASAF

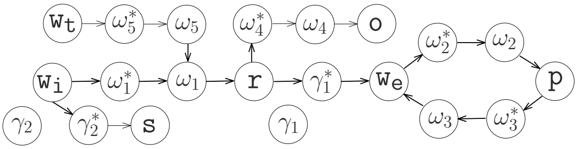


Figure 4. AF for the ASAF of Figure 2

Before defining the Proxy ASAF, we introduce some notation. Given an ASAF $\Delta = \langle A, \Omega, \Gamma \rangle$, for a set $S \subseteq A \cup \Omega \cup \Gamma$ of elements of Δ, we use $Reach_{\Delta}^{-1}(S) = \{Y \in A \cup \Omega \cup \Gamma \mid X \in S, X \in Reach_{\Delta}(Y)\}$ to denote the set of elements from which the elements in S are reachable in Δ. Moreover, we use $\Delta\!\downarrow_S$ to denote the *restriction* of an ASAF $\Delta = \langle A, \Omega, \Gamma \rangle$ to a set S of elements, that is $\Delta\!\downarrow_S = \langle A_S, \Omega_S, \Gamma_S \rangle$, where $A_S = A \cap S$, $\Omega_S = \{\omega \in \Omega \mid \mathbf{s}(\omega) \in A_S \wedge \mathbf{t}(\omega) \in (A_S \cup \Omega_S \cup \Gamma_S)\}$, and $\Gamma_S = \{\gamma \in \Gamma \mid \mathbf{s}(\gamma) \in A_S \wedge \mathbf{t}(\gamma) \in (A_S \cup \Omega_S \cup \Gamma_S)\}$.

The Proxy ASAF is the restriction of the updated ASAF $u(\Delta)$ to the alterable set plus the elements of $u(\Delta)$ that can reach an element in that set.

Definition 5 (Proxy ASAF). *Let $\Delta = \langle A, \Omega, \Gamma \rangle$ be an ASAF, $u = \pm\delta$ an update, and $G \in A \cup \Omega \cup \Gamma$ a goal element. Let $S = Alt(\Delta, u, G)$. The Proxy ASAF of Δ w.r.t u and G is $PASAF(\Delta, u, G) = u(\Delta)\!\downarrow_{S \cup Reach_{u(\Delta)}^{-1}(S)}$.*

Example 5. *Continuing from Example 4, the Proxy ASAF $PASAF(\Delta, u, \mathtt{p})$ is given by considering the restriction of the updated ASAF $u(\Delta)$ to the alterable set $S = Alt(\Delta, u, \mathtt{p})$ union $Reach_{u(\Delta)}^{-1}(S) = \{\mathtt{w_i}\}$, as reported in Figure 3.*

Observe that $PASAF(\Delta, u, G)$ is empty if $Alt(\Delta, u, G)$ is empty. In this case we can use the result of Theorem 1 to compute the skeptical acceptance. In contrast, the following theorem tells us how to use the Proxy ASAF to compute the skeptical preferred acceptance when the alterable set is not empty.

Theorem 2. *Let $\Delta = \langle A, \Omega, \Gamma \rangle$ be an ASAF, u an update, $u(\Delta)$ the updated ASAF, and a goal element $G \in A \cup \Omega \cup \Gamma$. If $Alt(\Delta, u, G) \neq \emptyset$ then G is skeptically preferred accepted w.r.t. $u(\Delta)$ iff it is skeptically preferred accepted w.r.t. the Proxy ASAF $PASAF(\Delta, u, G)$.*

Example 6. *Continuing from Example 5, $\mathtt{p}$ is skeptically preferred accepted w.r.t. the ASAF $u(\Delta)$ since $\mathtt{p}$ is skeptically preferred accepted w.r.t. the Proxy ASAF $PASAF(\Delta, u, \mathtt{p})$ of Figure 3 whose unique preferred extension is $\{\mathtt{w_i}, \mathtt{p}, \omega_1, \gamma_1, \omega_3\}$.*

3.1. Incremental Algorithm

The results of Theorems 1 and 2 allow us to define Algorithm 1 to decide the skeptical preferred acceptance of an element G w.r.t. an ASAF Δ updated by $u = \pm\delta$. Given the initial skeptical preferred acceptance $SA_{\Delta}(G)$, the skeptical preferred acceptance $SA_{u(\Delta)}(G)$ w.r.t. the updated ASAF is incrementally computed, thus enabling consecutive invocations of the algorithm to perform sequences of updates. Algorithm 1 works as follows. First the alterable set is computed at Line 1. Using result of Theorem 1, if the alterable set is empty then the acceptance status of G does not change after the update, and the algorithm returns the initial status at Line 3. Otherwise, the Proxy ASAF

Algorithm 1 *ASAF-SA*($\Delta, u, G, SA_{\Delta}(G)$, ASAF-Solver)

```
Input: ASAF $\Delta = \langle A, \Omega, \Gamma \rangle$, update $u$, goal $G \in A \cup \Omega \cup \Gamma$, initial skeptical preferred acceptance $SA_{\Delta}(G)$, function ASAF-Solver computing the skeptical preferred acceptance of a goal element for an ASAF.
Output: updated skeptically preferred acceptance of $G$ w.r.t $u(\Delta)$.
1: Let $S = Alt(\Delta, u, G)$
2: if $S = \emptyset$ then
3:         return $SA_{\Delta}(G)$;
4: Let $\Delta_P = PASAF(\Delta, u, G)$
5: return ASAF-Solver($G$, $\Delta_P$)
```

is built at Line 5 and, using Theorem 2, the skeptical acceptance of G can be computed by invoking an external ASAF-Solver that decides whether G is skeptically accepted by performing the computation on the Proxy ASAF (Line 5).

Algorithm 1 assumes that an ASAF solver is given. That is, in principle, our approach enables any external solver for ASAF to be used for the incremental computation of the preferred skeptical acceptance. However, to the best of our knowledge, currently there is no solver that directly performs the computation of skeptical acceptance on ASAFs (this is also due to the fact that the ASAF proposal is a quite recent, compared to Dung's framework for which several solvers have become available during the last few years). Therefore, instead of performing the computation on the Proxy ASAF, we leverage on a transformation of the Proxy ASAF to a Dung's framework to eventually compute the skeptical acceptance of the given goal. This makes our approach working with any available AF solver for the computation of the skeptical preferred acceptance. As explained below, we use the meta-AF approach recently proposed in [1] for computing ASAF extensions that can be adopted also for our scope.

Enabling the computation at the AF level

In this section, we first briefly review the transformation presented in [1] that allow us to characterize an ASAF in terms of an AF whose extensions (under preferred, grounded, complete, and stable semantics) are in a one-to-one correspondence with those of the given ASAF. Then, we show how to use this result to compute the skeptical acceptance.

An *AF for an ASAF* is an AF that encodes every argument, attack, and support of the given ASAF. The set of arguments of the AF consists of the arguments of Δ plus a pair of arguments, ω and ω^*, for each attack ω in Δ and a pair of arguments, γ and γ^*, for each support γ in Δ. Arguments ω and ω^* determine whether ω is accepted or not, and are used to propagate defeats on the source of ω to the attack itself. Argument γ represents the support itself and is used to determine whether it is accepted or not, whereas argument γ^* is used to propagate defeats on the source of γ to its target. Then, the attacks of the AF are as follows. For each attack ω in Δ, the AF contains a chain of 3 attacks starting in the source of ω and ending in its target, with intermediate arguments ω^* and ω; moreover, if the target of ω is a support γ, then an attack between ω and both γ^* and γ is added to the AF. For each support γ in Δ, the AF contains a chain of 2 attacks starting in the source of γ and ending in its target, with intermediate argument γ^*; finally, if the target of γ is a support γ_1, an attack between γ^* and γ_1^* is added.

Definition 6 (AF for ASAF [1])**.** *Let* $\Delta = \langle A, \Omega, \Gamma \rangle$ *be an ASAF. The* AF for Δ *is* $\Lambda_{\Delta} = \langle \mathbb{A}_{\Delta}, \Sigma_{\Lambda} \rangle$*, where:*

- $\mathbb{A}_\Delta = A \cup \{\omega, \omega^* \mid \omega \in \Omega\} \cup \{\gamma, \gamma^* \mid \gamma \in \Gamma\}$.
- $\Sigma_\Lambda = \{(\mathbf{s}(\omega), \omega^*), (\omega^*, \omega), (\omega, \mathbf{t}(\omega)) \mid \omega \in \Omega\} \cup \{(\omega, \mathbf{t}(\omega)^*) \mid \omega \in \Omega, \mathbf{t}(\omega) \in \Gamma\}$
  $\cup \{(\mathbf{s}(\gamma), \gamma^*), (\gamma^*, \mathbf{t}(\gamma)) \mid \gamma \in \Gamma\} \cup \{(\gamma^*, \mathbf{t}(\gamma)^*) \mid \gamma \in \Gamma, \mathbf{t}(\gamma) \in \Gamma\}$.

Example 7. *The AF for the ASAF Δ of Figure 2 is Λ_Δ shown in Figure 4. For instance, the attack $\omega_1 = (\mathtt{w_i}, \mathtt{r})$ corresponds to the chain of attacks from $\mathtt{w_i}$ to $\mathtt{r}$ through ω_1 and ω_1^*, while $\omega_5 = (\mathtt{w_t}, \omega_1)$ corresponds to the attacks $(\mathtt{w_t}, \omega_5^*), (\omega_5^*, \omega_5), (\omega_5, \omega_1)$.*

In [1], it is shown that there exists a one-to-one correspondence between the preferred extensions of an ASAF Δ and the preferred extensions of the AF Λ_Δ for Δ, modulo meta-arguments ω^* and γ^*. This equivalence between extensions of an ASAF and extensions of the corresponding AF allow us to state the following result.

Theorem 3. *Let $\Delta = \langle A, \Omega, \Gamma \rangle$ be an ASAF, Λ_Δ the AF for Δ, and G an element in $A \cup \Omega \cup \Gamma$. Therefore, G is skeptically preferred accepted w.r.t. Δ iff the argument corresponding to G is skeptically preferred accepted w.r.t. Λ_Δ.*

Algorithm 2: a variant of Algorithm 1 using an AF solver. To perform the computation of the skeptical preferred acceptance by using a state-of-the-art AF solver, we modify Algorithm 1 as follows. Let ASAFtoAF be a function that takes as input an ASAF Δ and returns the corresponding AF Λ_Δ. Then, the invocation of the ASAF solver at Line 5 of Algorithm 1 is replaced by AF-Solver($\overline{G}$, ASAFtoAF(Δ_P)), where AF-Solver is a function computing the skeptical preferred acceptance of a given argument w.r.t. a given AF, and $\overline{G}$ is the argument of Λ_Δ corresponding to G. Let Algorithm 2 be the so-obtained algorithm. As stated next it is sound and complete.

Theorem 4. *If AF-Solver is sound and complete, for any goal element G of Δ, Algorithm 2 returns $SA_{u(\Delta)}(G)$ w.r.t. the updated ASAF $u(\Delta)$.*

4. Empirical Evaluation

We implemented a C++ prototype and compared the performance of: 1) the *incremental approach*, that is Algorithm 2 where AF-Solver is *μ-toksia* [35], the winner of the last ICCMA edition for the task DS-pr (i.e., computing the skeptical preferred acceptance of an argument of an AF); and 2) the *computation from scratch*, that is the computation of the skeptical preferred acceptance of the goal element w.r.t the updated ASAF by running AF-Solver (i.e., *μ-toksia*) directly on the AF for the updated ASAF.

Dataset. Although there are several benchmark generators and solvers for Dung's AFs [38], only a benchmark has been recently proposed for ASAFs [1]. Following [1], we generated a set of benchmark ASAFs by starting from AFs used as benchmark at ICCMA'19. Specifically, we use an AF dataset consisting of 326 AFs and, given a benchmark AF Λ, we generate an ASAF as follows: 30% of attacks in Λ are transformed into first-level supports; 12% (resp. 3%) of attacks in Λ are transformed into second-level supports towards a support (resp. an attack); 3% (resp. 2%) of attacks in Λ are transformed into third-level supports towards a support (resp. an attack); 12% (resp. 3%) of attacks in Λ are transformed into second-level attacks towards an attack (resp. a support); 2% (resp. 3%) of attacks in Σ are transformed into third-level attacks towards an attack (resp. a support); the remaining 30% of attacks in Λ are kept as first-level attacks of the

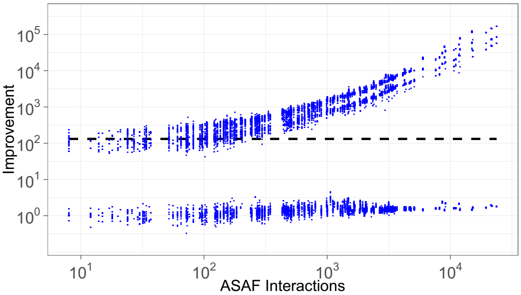


Figure 5. Improvement of the incremental approach over the computation from scratch (log scales). The dashed black line represents the median value.

resulting ASAF. This benchmark generation process aimed at preserving AFs' topology as much as possible. However, the process of generating ASAF benchmarks starting from AF benchmarks is challenging because we require specific amounts of different kind of attacks and supports, and we also need to check that the sub-graph induced by first-level supports is acyclic. Hence, to make it feasible, for each dataset, we generated an ASAF Δ if the number of arguments $|\mathbb{A}_\Delta|$ of the AF Λ_Δ for Δ does not exceed the number of arguments of the biggest AF in the original dataset. Therefore, starting from the AF dataset, we obtained an ASAF dataset consisting of 284 ASAFs $\Delta = \langle A, \Omega, \Gamma \rangle$ with a number of arguments $|A| \in [5, 10K]$ and a number of interactions $|\Omega \cup \Gamma| \in [8, 310K]$.

Methodology. For each ASAF Δ in the dataset, we consider a (randomly chosen) goal element and an update u selected among one of the possible 12 types (addition/deletion of an attack/support towards an argument/attack/support). Next, we compute the updated skeptical preferred acceptance of the goal element in the updated ASAF $u(\Delta)$ by calling Algorithm 2. Finally, we compute the *improvement* of Algorithm 2 over the computation from scratch as t_s/t_{A_2} where *i*) t_s is the time needed by the computation from scratch, and *ii*) t_{A_2} is the time needed by Algorithm 2. Thus, the improvement tells us how many times Algorithm 2 is faster than the computation from scratch. The experiments have been carried out on an Intel Core i7-3770K CPU 3.5GHz, 12GB RAM, running Ubuntu.

Results. Figure 5 reports the improvement versus the number of ASAF interactions (i.e., $|\Omega \cup \Gamma|$). Each data point refers to a run concerning an update and a goal. We also report the median of the improvement (dashed black line). Since μ-toksia ran into memory capacity saturation when computing the skeptical acceptance for $4,9\%$ of the AFs for the ASAFs in the dataset, we report the results for the remaining 244 ASAFs having number of arguments $|A| \in [5, 10K]$ and number of interactions $|\Omega \cup \Gamma| \in [8, 23.7K]$.

The results in Figure 5 show that, for a given goal and update, the improvement can be either very large or limited. This is due to the fact that either *i*) the alterable set is empty, and thus the algorithm immediately recognizes that acceptance status of the goal does not change after the update, or *ii)* the Proxy ASAF is built to compute the skeptical acceptance of the goal by invoking the external solver. Case *i*) occurs for 56% of the data points, and the average improvement in this case is 5836. The average improvement in the other case is 1.53, that is, the incremental computation takes 65% of the amount of time needed by the computation from scratch. In particular, although the size of the Proxy ASAF is 70.1% of that of the input ASAF on average, there is an overhead due to the construction of the Proxy ASAF that, to some extent, mitigates the benefit of the local computation on the smaller ASAF. Finally, the running time of Algorithm 2 is slightly more than that of the computation from scratch for only 3.8% of the data points. However, overall the incremental algorithm outperforms the computation from scratch, as confirmed by the median value of the improvements which is equal to 131 (the average is 3287, but is skewed by huge values of improvements in Figure 5).

5. Conclusions and Future Work

There has been an extensive body of work on managing changes in argumentation (a survey can be found in [28]). Besides the works mentioned in the introduction, other significant efforts coping with dynamics aspects of AFs include [10,14,21,23]. Similarly to what is done in this paper, some approaches focused on local computation in dynamic AFs [2,15,34,32] but with the aim of recomputing extensions. Recently, as discussed in Section 3.1, an algorithm for the incremental computation of an extension of dynamic ASAFs has been proposed in [1]. Moreover, an incremental approach to computing skeptical acceptance in Dung's frameworks has been proposed in [5], where the ideal extension is used for the computation and it is incrementally maintained. To the best of our knowledge, this is the first paper proposing an incremental technique for the computation of skeptical preferred acceptance in dynamic ASAFs. Due to the generality of ASAF, our technique can be also applied to restricted frameworks such as Argumentation Frameworks with Recursive Attacks (AFRAs) [13] and AFNs [36].

As future work we plan to investigate similar approaches for Recursive Argumentation Framework with Necessities (RAFN) [24], where a support may come also from a set of arguments, as well as extending our technique to deal with other semantics and considering the problem of enumerating extensions (as done for AFs [6]).

References

[1] G. Alfano, A. Cohen, S. Gottifredi, S. Greco, F. Parisi, and G. R. Simari. Dynamics in abstract argumentation frameworks with recursive attack and support relations. In *ECAI 2020 (to appear)*.

[2] G. Alfano, S. Greco, and F. Parisi. Efficient computation of extensions for dynamic abstract argumentation frameworks: An incremental approach. In *IJCAI*, pages 49–55, 2017.

[3] G. Alfano, S. Greco, and F. Parisi. Computing extensions of dynamic abstract argumentation frameworks with second-order attacks. In *IDEAS*, pages 183–192, 2018.

[4] G. Alfano, S. Greco, and F. Parisi. A meta-argumentation approach for the efficient computation of stable and preferred extensions in dynamic bipolar argumentation frameworks. *Intelligenza Artificiale*, 12(2):193–211, 2018.

[5] G. Alfano, S. Greco, and F. Parisi. An efficient algorithm for skeptical preferred acceptance in dynamic argumentation frameworks. In *IJCAI*, pages 18–24, 2019.

[6] G. Alfano, S. Greco, and F. Parisi. On scaling the enumeration of the preferred extensions of abstract argumentation frameworks. In *SAC*, pages 1147–1153, 2019.

[7] G. Alfano, S. Greco, F. Parisi, G. I. Simari, and G. R. Simari. Incremental computation of warranted arguments in dynamic defeasible argumentation: the rule addition case. In *SAC*, pages 911–917, 2018.

[8] G. Alfano, S. Greco, F. Parisi, G.I. Simari, and G.R. Simari. An incremental approach to structured argumentation over dynamic knowledge bases. In *KR*, pages 78–87, 2018.

[9] L. Amgoud, C. Cayrol, and M.-C. Lagasquie-Schiex. On the bipolarity in argumentation frameworks. In *NMR*, pages 1–9, 2004.

[10] L. Amgoud and S. Vesic. Revising option status in argument-based decision systems. *J. Log. Comp.*, 22(5):1019–1058, 2012.

[11] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, Henry Prakken, C. Reed, G. R. Simari, M. Thimm, and Serena Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.

[12] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. W. N. van der Torre, and S. Villata. On the input/output behavior of argumentation frameworks. *AI*, 217:144–197, 2014.

[13] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida. AFRA: Argumentation Framework with Recursive Attacks. *IJAR*, 52(1):19–37, 2011.

[14] P. Baroni, M. Giacomin, and G. Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005.

[15] P. Baroni, M. Giacomin, and B. Liao. On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method. *AI*, 212:104–115, 2014.

[16] R. Baumann. Splitting an argumentation framework. In *LPNMR*, pages 40–53, 2011.

[17] T.J.M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *AI*, 171:619 – 641, 2007.

[18] P. Bisquert, C. Cayrol, F. Dupin de Saint-Cyr, and M.-C. Lagasquie-Schiex. Characterizing change in abstract argumentation systems. In *Trends in Belief Revision and Argumentation Dynamics*, volume 48, pages 75–102. 2013.

[19] S. Bistarelli, F. Faloci, F. Santini, and C. Taticchi. Studying dynamics in argumentation with Rob. In *COMMA*, pages 451–452, 2018.

[20] G. Boella, D. M. Gabbay, L. W. N. van der Torre, and S. Villata. Support in abstract argumentation. In *COMMA*, pages 111–122, 2010.

[21] G. Boella, S. Kaci, and L. W. N. van der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In *ECSQARU*, pages 107–118, 2009.

[22] M. W. A. Caminada, W. Dvorák, and S. Vesic. Preferred semantics as socratic discussion. *J. of Log. and Comp.*, 26(4):1257–1292, 2016.

[23] C. Cayrol, F. Dupin de Saint-Cyr, and M.-C. Lagasquie-Schiex. Revision of an argumentation system. In *KR*, pages 124–134, 2008.

[24] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasquie-Schiex. Structure-based semantics of argumentation frameworks with higher-order attacks and supports. In *COMMA*, pages 29–36, 2018.

[25] C. Cayrol and M.-C. Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *IJAR*, 54(7):876–899, 2013.

[26] G. Charwat, W. Dvorák, S. A. Gaggl, J. P. Wallner, and S. Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *AI*, 220:28–63, 2015.

[27] A. Cohen, S. Gottifredi, A. J. Garcia, and G. R. Simari. A survey of different approaches to support in argumentation systems. *The Knowl. Eng. Rev.*, 29(5):513–550, 2014.

[28] S. Doutre and J.-G. Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *A & C*, 9(3):223–248, 2018.

[29] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI*, 77(2):321–358, 1995.

[30] P. E. Dunne and M. Wooldridge. Complexity of abstract argumentation. In *Argumentation in Artificial Intelligence*, pages 85–104. 2009.

[31] S. Gottifredi, A. Cohen, A. J. Garcia, and G. R. Simari. Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. *AI*, 262:336–368, 2018.

[32] S. Greco and F. Parisi. Incremental computation of deterministic extensions for dynamic argumentation frameworks. In *JELIA*, pages 288–304, 2016.

[33] M. Kröll, R. Pichler, and S. Woltran. On the complexity of enumerating the extensions of abstract argumentation frameworks. In *IJCAI*, pages 1145–1152, 2017.

[34] B. Liao, L. J., and R. C. Koons. Dynamics of argumentation systems: A division-based method. *AI*, 175(11):1790–1814, 2011.

[35] A. Niskanen and M. Järvisalo. μ-toksia participating in ICCMA 2019. *Third ICCMA*, 2019.

[36] F. Nouioua and V. Risch. Argumentation frameworks with necessities. In *SUM*, pages 163–176, 2011.

[37] I. Rahwan and G. R. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

[38] M. Thimm and S. Villata. The first international competition on computational models of argumentation: Results and analysis. *AI*, 252:267–294, 2017.

[39] S. Villata, G. Boella, D. M. Gabbay, and L. W. N. van der Torre. Modelling defeasible and prioritized support in bipolar argumentation. *AMAI*, 66(1-4):163–197, 2012.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200494

An Adjustment Function for Dealing with Similarities

Leila AMGOUD [a] and Victor DAVID [b,1]
[a] *CNRS - IRIT, France*
[b] *Université Paul Sabatier - IRIT, France*

Abstract. The paper investigates gradual semantics that are able to deal with similarity between arguments. Following the approach that defines semantics with evaluation methods, i.e., a couple of aggregation functions, the paper argues for the need of a novel function, called *adjustment function*. The latter is responsible for taking into account similarity when it is available. It aims at reducing the strengths of attackers according to the possible similarities between them. The reason is that similarity is seen as redundancy that should be avoided, otherwise a semantics may return inaccurate evaluations of arguments. The paper proposes a novel adjustment function that is based on the well-known weighted h-Categorizer, and investigates its formal properties.

Keywords. Argumentation, Similarity, Adjustment Function, Gradual Semantics.

1. Introduction

Argumentation is a reasoning approach, which justifies claims by *arguments*. It starts by generating arguments and their links (forming an argumentation graph), then evaluates the arguments by so-called *semantics*, and finally identifies winning claims. An argumentation graph can be enriched with various additional information like weights on arguments, which can represent votes [1] or certainty degrees [2], weights on links between arguments, which can represent relevance [3,4] or again votes of users [5]. A similarity measure assessing how alike are pairs of arguments may also be provided [6,7].

Existence of similarity between arguments is inevitable in practice, as arguments generally share information. Hence, developing semantics that are able to take into account similarity is crucial for discarding any redundancy that may lead to inaccurate evaluation of arguments. This is particularly the case for gradual semantics and more precisely those that satisfy the *Counting* principle from [8], which states that every alive attacker affects its target. Consequently, the authors in [9] proposed some reasonable properties on how a gradual semantics should deal with similarity. Furthermore, they proposed three gradual semantics that deal with similarity. They all extend *h*-Categorizer [10] but differ in the way they remove redundancy that is due to similarity. However, the three approaches suffer from weaknesses as described in the related work section.

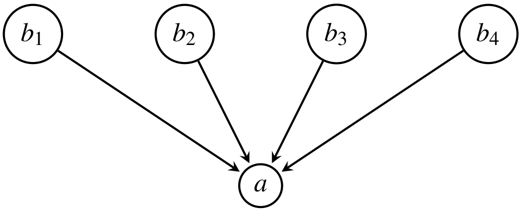
In this paper, we start first by extending the general framework for gradual semantics that was proposed in [11]. That framework defines a gradual semantics by an evaluation

[1] Corresponding Author: victor.david@irit.fr

method, which is a tuple of three aggregation functions. This approach makes clear the different operations done by a semantics. We start by relaxing the strong constraint that all arguments are dissimilar. Indeed, we assume availability of similarities between arguments. Then, we extend the definition of evaluation method by introducing a fourth function, called adjustment function. It is responsible for reducing the strengths of attackers of an argument according to the similarity between them. Let us consider the following example of argumentation graph on the reduction of carbon emissions. Four arguments (b_1, b_2, b_3, b_4) are given and they all attack an argument (a):

- b_1: decreasing the population implies lower carbon emissions,
- b_2: reducing the use of aircraft implies lower carbon emissions,
- b_3: reducing distant imports implies lower carbon emissions,
- b_4: increasing local trade implies lower carbon emissions.

We can graphically represent this debate as follows:



In this example we can observe that there are some similarities between the arguments b_i. The more similar ones are b_3 and b_4 as they are about the same idea with dual writing. The argument b_2 has some similarity with b_3 and b_4 because reducing distant imports implies reducing the use of aircraft but not only (freighters, trains, trucks). And for the argument b_1 even if indirectly population touch everything, we can assume that this premise based on demography is different from the others (talking about transport or economy). Note that it is important to avoid these redundancies when evaluating the argument a. For that purpose, a reasonable semantics would start by evaluating the strength of each of the attackers b_i, then readjust those values by taking into account similarity. For instance, if a semantics assigns the value 1 to both b_3 and b_4 because they are not attacked, at the second step it may for instance decide to keep the whole strength of b_3 and set the value of b_4 to 0 due to the full similarity between the two arguments.

Another contribution of the paper consists of proposing a novel readjustment function. The latter distributes the burden of redundancy among attackers. In the previous example, the new function will decrease the value of both b_3, b_4. Furthermore, the function is based on the well-known weighted h-Categorizer that was proposed in the literature for a completely different purpose. Indeed, it is used as a gradual semantics for evaluating arguments. We investigate the properties of the novel function and compare it with the existing ones.

The paper is organised as follows: Section 2 introduces the argumentation framework, we are interested and extend the framework of gradual semantics proposed in [11]. Section 3 presents the novel readjustment function, and Section 4 investigates its properties. Section 5 is devoted to related work and the last section concludes.

2. Background

Throughout the paper, we denote by $\mathscr{U}$ the universe of all possible arguments, and consider argumentation frameworks as tuples made of a non-empty and finite subset of $\mathscr{U}$. Every argument has an initial weight that may represent different information (certainty degree of the argument's premises, credibility degree of its source, and so on). Arguments can attack each other and every attack is assigned a weight representing for instance its relevance as in case of analogical arguments [3]. We also assume availability of a similarity measure that assesses how alike are pairs of arguments.

For the sake of simplicity, all weights and similarities are elements of the unit interval [0,1]. The greater the value, the stronger the argument, or the more relevant an attack, or the more similar the pair of arguments.

Definition 1 (Similarity Measure) *A* similarity measure *on a set $X \subseteq_f \mathscr{U}$*[2] *is a function* $\mathbf{s} : X \times X \to [0,1]$ *such that:*

- $\forall a \in X$, $\mathbf{s}(a,a) = 1$,
- $\forall a,b \in X$, $\mathbf{s}(a,b) = \mathbf{s}(b,a)$.

The first condition states that every argument is fully similar to itself and the second states that similarity is a symmetric notion.

Definition 2 (AF) *An argumentation framework (AF) is a tuple* $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$, *where*

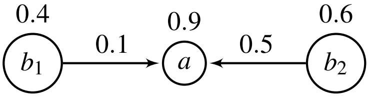
- $\mathscr{A} \subseteq_f \mathscr{U}$
- $\mathbf{w} : \mathscr{A} \to [0,1]$
- $\mathscr{R} \subseteq \mathscr{A} \times \mathscr{A}$
- $\sigma : \mathscr{R} \to [0,1]$
- $\mathbf{s} : \mathscr{A} \times \mathscr{A} \to [0,1]$

For $a,b \in \mathscr{A}$, $\mathbf{w}(a)$ denotes the initial weight of a, $\mathbf{s}(a,b)$ is the degree of similarity between a and b, $(a,b) \in \mathscr{R}$ means a attacks b, a is called *attacker* of b, $\sigma(a,b)$ is the degree of relevance of the attack, and $\mathtt{Att}(a)$ denotes the set of all attackers of a. Finally, the notation $\mathbf{s} \equiv 0$ denotes that there are no similarities between arguments.

In [12], the authors introduced for the first time *gradual semantics*, i.e, formal methods that evaluate strengths of arguments. Formally, they are functions that assign to every argument in an argumentation framework a value from an ordered scale. Examples of such semantics are h-Categorizer [10], Trust-based semantics [13], (DF)-Quad [14,15] and those proposed in [8].

In [11], the authors studied argumentation frameworks of the form $\langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \equiv 0 \rangle$, and have shown that a gradual semantics is defined using three functions. In order to better motivate those functions, let us consider the graph depicted below and focus on the argument a:

[2] $X \subseteq_f \mathscr{U}$ means X is a finite subset of $\mathscr{U}$.



In order to assess the strength of a, a gradual semantics proceeds in three steps:

1. To assess the *strength of every attack* (b_i, a). The idea is to combine the strength of the attacker b_i with the relevance degree of the attack $\sigma(b_i, a)$. This is done by a function **h**. Since b_1, b_2 are not attacked, assume a semantics that keeps their initial weights, i.e., the strength of b_1 is 0.4 and the strength of b_2 is 0.6. Hence, $\alpha_1 = \mathbf{h}(0.4, \sigma(b_1, a)) = \mathbf{h}(0.4, 0.1)$ and $\alpha_2 = \mathbf{h}(0.6, \sigma(b_2, a)) = \mathbf{h}(0.6, 0.5)$, where α_i represents the strength of the attack (b_i, a).
2. To assess the *strength of the group of attacks* on a. This is done by an aggregation function **g**, hence $\delta = \mathbf{g}(\alpha_1, \alpha_2)$.
3. To evaluate the impact of the two attacks on the initial weight of a. This is done by a function **f**, hence $\lambda = \mathbf{f}(\mathbf{w}(a), \delta) = \mathbf{f}(0.9, \delta)$. This function returns the strengths of arguments, thus the strength of a is λ.

The tuple $\mathcal{M} = \langle \mathbf{f}, \mathbf{g}, \mathbf{h} \rangle$ of the three functions is called in [11] an *evaluation method* of a gradual semantics. An important question is: how a semantics should consider similarities when they are available and at which step of the above process? As discussed in [9], a gradual semantics should take into account similarities between the attackers of an argument. In the above graph, assume that b_1 and b_2 are fully similar, i.e., $\mathbf{s}(b_1, b_2) = 1$. Note that b_1 is redundant wrt b_2, and thus considering both α_1 and α_2 will lead to an inaccurate evaluation of the argument a. Indeed, a will loose a lot of weight due to redundant information. Hence, before computing the strength of the group of attacks using the aggregation function **g**, we introduce an *adjustment function* **n** that readjusts the two values considering the similarity between b_1 and b_2. This operation results in a *decrease* in the strength of the group of attacks. For instance, this function may keep the greatest value among α_1 and α_2 and sets the other to 0. Assume that $\alpha_1 > \alpha_2$, then $\mathbf{n}(\alpha_1, \alpha_2) = (\alpha_1, 0)$ and the strength of the group would be $g(\alpha_1, 0)$. Note that such function ignores the attack from b_2. In the next section, we provide a novel adjustment function that distributes the burden of redundancy among the two attackers. Before that, let us first extend the definition of evaluation method.

Definition 3 (EM) *An* evaluation method *(*EM*) is a tuple* $\mathcal{M} = \langle \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{n} \rangle$ *such that:*

- $\mathbf{f} : [0,1] \times \mathtt{Range}(\mathbf{g})$ [3] $\rightarrow [0,1]$,
- $\mathbf{g} : \bigcup_{k=0}^{+\infty} [0,1]^k \rightarrow [0, +\infty[$,
- $\mathbf{h} : [0,1] \times [0,1] \rightarrow [0,1]$,
- $\mathbf{n} : \bigcup_{k=0}^{+\infty} \; ([0,1] \times \mathscr{U})^k \rightarrow [0,1]^k$.

Note also that the function **n** takes as input two kinds of input: k numerical values and k arguments. The reason is that the same values may not be adjusted in the same way depending on the similarity between the arguments to which they refer.

Let us now define formally a gradual semantics that deals with similarity.

[3] $\mathtt{Range}(\mathbf{g})$ denotes the co-domain of **g**

Definition 4 (Gradual Semantics) *A* gradual semantics $\mathscr{S}$ *based on an evaluation method* $\mathscr{M} = \langle \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{n} \rangle$ *is a function assigning to every* AF, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$*, a weighting* $\mathtt{Deg}^{\mathscr{S}}_{\mathbf{G}} : \mathscr{A} \to [0,1]$ *such that for every* $a \in \mathscr{A}$, $\mathtt{Deg}^{\mathscr{S}}_{\mathbf{G}}(a) =$

$$\mathbf{f}\Bigg(\mathbf{w}(a), \mathbf{g}\Bigg(\mathbf{n}\Big((\mathbf{h}(\mathtt{Deg}^{\mathscr{S}}_{\mathbf{G}}(b_1), \sigma(b_1,a)), b_1), \cdots, (\mathbf{h}(\mathtt{Deg}^{\mathscr{S}}_{\mathbf{G}}(b_k), \sigma(b_k,a)), b_k)\Big)\Bigg)\Bigg),$$

where $\{b_1, \cdots, b_k\} = \mathtt{Att}(a)$. $\mathtt{Deg}^{\mathscr{S}}_{\mathbf{G}}(a)$ *is the* strength *of a.*

It has been shown in [11] that most of existing gradual semantics are instances of the above definition. An example is weighted h-Categorizer defined as follows:

Definition 5 (Weighted h-Categorizer Gradual Semantics) *Weighted h-categorizer semantics is a function* $\mathscr{S}_{\mathtt{wh}}$ *transforming any* AF, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \equiv 0 \rangle$*, into a weighting* $\mathtt{Deg}^{\mathscr{S}_{\mathtt{wh}}}_{\mathbf{G}} : \mathscr{A} \to [0,1]$ *such that for every* $a \in \mathscr{A}$,

$$\mathtt{Deg}^{\mathscr{S}_{\mathtt{wh}}}_{\mathbf{G}}(a) = \begin{cases} \mathbf{w}(a) & \textit{iff } \mathtt{Att}(a) = \emptyset \\ \frac{\mathbf{w}(a)}{1 + \sum\limits_{b \in \mathtt{Att(a)}} \mathtt{Deg}^{\mathscr{S}_{\mathtt{wh}}}_{\mathbf{G}}(b) \times \sigma(b,a)} & \textit{else} \end{cases}$$

The above semantics uses an evaluation method $\mathscr{M} = \langle \mathbf{f}, \mathbf{g}, \mathbf{h} \rangle$ such that:

$$\mathbf{f}_{\mathtt{frac}}(x_1, x_2) = \frac{x_1}{1+x_2} \;\Big|\Big|\; \mathbf{g}_{\mathtt{sum}}(x_1, \cdots, x_n) = \sum_{i=1}^{n} x_i \;\Big|\Big|\; \mathbf{h}_{\mathtt{prod}}(x_1, x_2) = x_1 \times x_2$$

3. A Novel Adjustment Function

Throughout this section, we assume an arbitrary but fixed argumentation framework $\langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ and an arbitrary gradual semantics for evaluating its arguments. In what follows, we focus on the adjustment function of this semantics. We define this function, denoted by $\mathbf{n}_{\mathtt{wh}}$. The new function is nothing else than weighted h-Categorizer that is used in the literature as a gradual semantics for evaluating the strength of arguments. An important question is: why a gradual semantics can itself play the role of an adjustment function? The answer lies in the great analogy between the two: both aim at reducing strengths of arguments according to a set of other arguments. Another key question is: on which argumentation framework is the semantics applied? Recall that an input of any adjustment function is a tuple of the form $((x_1, b_1), \cdots, (x_n, b_n))$, with $x_i \in [0,1]$ is given by the gradual semantics that is used and $b_i \in \mathscr{A}$. For every such input, we create an argumentation framework $\langle \mathscr{A}', \mathbf{w}', \mathscr{R}', \sigma', \mathbf{s}' \rangle$ such that:

- $\mathscr{A}' = \{b_1, \cdots, b_n\}$
- For every $b_i \in \mathscr{A}'$, $\mathbf{w}'(b_i) = x_i$
- $\mathscr{R}' = (\mathscr{A}' \times \mathscr{A}') \setminus \{(b_i, b_i) \mid i = 1, \cdots, n\}$
- For every $(b_i, b_j) \in \mathscr{R}'$, $\sigma'((b_i, b_j)) = \mathbf{s}(b_i, b_j)$
- $\mathbf{s}' \equiv 0$

The framework contains thus the set of attackers whose strengths should be readjusted, the initial weight of every argument is its value assigned by the semantics, the attack relation is symmetric and the weight of every attack is the similarity degree between its target and its source. Weighted h-Categorizer is applied to this framework and the values assigned to arguments correspond to their readjusted values.

Definition 6 ($\mathbf{n}_{\mathtt{wh}}$) *Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an* AF*, $x_1, \cdots, x_k \in [0,1]$, and $b_1, \cdots, b_k \in \mathscr{A}$. We define the adjustment function $\mathbf{n}_{\mathtt{wh}}$ as follows:*

$$\mathbf{n}_{\mathtt{wh}}((x_1,b_1),\cdots,(x_k,b_k)) = (\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_1),\cdots,\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_k))$$

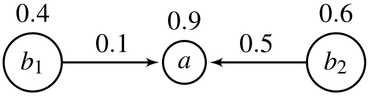
where $\mathbf{G}' = \langle \mathscr{A}', \mathbf{w}', \mathscr{R}', \sigma', \mathbf{s}' \rangle$, such that:

- $\mathscr{A}' = \{b_1, \cdots, b_k\}$,
- $\mathbf{w}'(b_1) = x_1, \cdots, \mathbf{w}'(b_k) = x_k$,
- $\mathscr{R}' = \{(b_1,b_2), \cdots, (b_1,b_k), \cdots, (b_k,b_1), \cdots, (b_k,b_{k-1})\}$,
- *For every $(b_i,b_j) \in \mathscr{R}'$, $\sigma'((b_i,b_j)) = \mathbf{s}(b_i,b_j)$,*
- $\mathbf{s}' \equiv 0$.

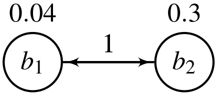
Hence, the strength x_i of every attacker b_i will be readjusted to $\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_i)$, where

$$\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_i) = \frac{x_i}{1 + \sum\limits_{j \in \{1,\cdots,n\} \setminus \{i\}} \mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_j) \times \mathbf{s}(b_j,b_i)}.$$

Example 1 *Let us illustrate the above definition on the graph below.*



Assume that b_1 and b_2 are fully similar ($\mathbf{s}(b_1,b_2) = 1$) and let us consider a semantics that satisfies the Maximality principle from [8] according to which every non-attacked argument keeps its initial weight. Hence, the strength of b_1 is 0.4 and the strength of b_2 is 0.6. Assume also that to deal with the weight of relevance we use $\mathbf{h}_{\mathtt{prod}}$ then the adjustment function takes thus the tuples $(0.04,b_1),(0.3,b_2)$ as input. It builds the following argumentation framework:



$\mathbf{n}_{\mathtt{wh}}$ *evaluates the arguments of the above graph using weighted h-Categorizer. It is easy to check that $\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_1) = 0.03$, $\mathtt{Deg}_{\mathbf{G}'}^{\mathscr{S}_{\mathtt{wh}}}(b_2) = 0.29$. So, $\mathbf{n}_{\mathtt{wh}}((0.04,b_1),(0.3,b_2)) = (0.03, 0.29)$ meaning that the readjusted value of b_1 and b_2 are respectively* 0.03 *and* 0.29.

4. Properties

This section shows that the function $\mathbf{n}_{\mathtt{wh}}$ satisfies reasonable properties. The first result shows that $\mathbf{n}_{\mathtt{wh}}$ can be used by a gradual semantics. Namely, the gradual semantics that is based on the evaluation method $\langle \mathbf{f}_{\mathtt{frac}}, \mathbf{g}_{\mathtt{sum}}, \mathbf{h}_{\mathtt{prod}}, \mathbf{n}_{\mathtt{wh}} \rangle$ assigns a unique strength to every argument.

Theorem 1 *There exists a unique semantics that is based on the evaluation method* $\langle \mathbf{f}_{\mathtt{frac}}, \mathbf{g}_{\mathtt{sum}}, \mathbf{h}_{\mathtt{prod}}, \mathbf{n}_{\mathtt{wh}} \rangle$.

Proof In [11] the authors show that the weighted h-categorizer semantics can be defined by the evaluation method $\langle \mathbf{f}_{\mathtt{frac}}, \mathbf{g}_{\mathtt{sum}}, \mathbf{h}_{\mathtt{prod}} \rangle$ and it has a unique semantics, i.e. it converge. More, the adjustment function $\mathbf{n}_{\mathtt{wh}}$ is the weighted h-categorizer semantics which modifies each weight of argument in its degree. Apply the adjustment function $\mathbf{n}_{\mathtt{wh}}$ before the aggregation function $\mathbf{g}_{\mathtt{sum}}$ changes the value of the arguments. This is equivalent to using $\langle \mathbf{f}_{\mathtt{frac}}, \mathbf{g}_{\mathtt{sum}}, \mathbf{h}_{\mathtt{prod}} \rangle$ on a different graph. Therefore $\langle \mathbf{f}_{\mathtt{frac}}, \mathbf{g}_{\mathtt{sum}}, \mathbf{h}_{\mathtt{prod}}, \mathbf{n}_{\mathtt{wh}} \rangle$ converge and has a unique semantics. ■

As expected from an adjustment function, the next property states that $\mathbf{n}_{\mathtt{wh}}$ can only reduce the value of an argument.

Proposition 1 *For any* AF, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$, *for all* $a_1, \cdots, a_n \in \mathscr{A}$, *for all* $x_1, \cdots, x_n \in [0,1]$, *if* $\mathbf{n}_{\mathtt{wh}}((x_1,a_1), \cdots, (x_n,a_n)) = (x'_1, \cdots, x'_n)$, *then* $\forall i \in \{1, \cdots, n\}$, $x'_i \leq x_i$.

Proof Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an AF, $a_1, \cdots, a_n \in \mathscr{A}$ and $x_1, \cdots, x_n \in [0,1]$ such that $\mathbf{n}_{\mathtt{wh}}((x_1,a_1), \cdots, (x_n,a_n)) = (\mathtt{Deg}(a_1), \cdots, \mathtt{Deg}(a_n))$. For any $i \in \{1, \cdots, n\}$, from Definition 6, $\mathtt{Deg}(a_i) = \frac{x_i}{1+X}$ such that $X \in [0, +\infty[$ therefore $\mathtt{Deg}(a_i) \leq x_i$. ■

When all the arguments are dissimilar, the adjustment function does not alter the values of the arguments.

Proposition 2 *For any* AF, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$, *for all* $a_1, \cdots, a_n \in \mathscr{A}$, *for all* $x_1, \cdots, x_n \in [0,1]$, *if* $\forall i, j \in \{1, \cdots, n\}$, $i \neq j$, $\mathbf{s}(a_i, a_j) = 0$, *then*

$$\mathbf{n}_{\mathtt{wh}}((x_1,a_1), \cdots, (x_n,a_n)) = (x_1, \cdots, x_n).$$

Proof Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an AF, $a_1, \cdots, a_n \in \mathscr{A}$ and $x_1, \cdots, x_n \in [0,1]$ such that $\forall i, j \in \{1, \cdots, n\}$, $i \neq j$, $\mathbf{s}(a_i, a_j) = 0$. From Definition 6, $\mathbf{n}_{\mathtt{wh}}((x_1,a_1), \cdots, (x_n,a_n)) = (\mathtt{Deg}(a_1), \cdots, \mathtt{Deg}(a_n))$ such that $\mathtt{Deg}(a_1) = \frac{x_1}{1+0}, \cdots, \mathtt{Deg}(a_n) = \frac{x_n}{1+0}$. ■

We show next that increasing the degree of similarity of a pair of arguments leads to the diminution of values of both arguments.

Proposition 3 *For any* AF, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$, *for all* $a_1, a_2, b_1, b_2 \in \mathscr{A}$ *and for any* $x_1, x_2 \in\ ]0,1]$, *if*

- $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),(x_2,a_2)) = (x'_1, x'_2)$,
- $\mathbf{n}_{\mathtt{wh}}((x_1,b_1),(x_2,b_2)) = (x''_1, x''_2)$,
- $\mathbf{s}(b_1,b_2) > \mathbf{s}(a_1,a_2)$,

then $x'_1 > x''_1$ *and* $x'_2 > x''_2$.

Proof Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an $\mathtt{AF}$, $a_1, a_2, b_1, b_2 \in \mathscr{A}$ and $x_1, x_2 \in]0,1]$ such that

- $\mathbf{n}_{\mathtt{wh}}((x_1, a_1), (x_2, a_2)) = (\mathtt{Deg}(a_1), \mathtt{Deg}(a_2))$,
- $\mathbf{s}(b_1, b_2) = \mathbf{s}(a_1, a_2) + \alpha$ such that $\alpha \in]0,1]$ and $\mathbf{n}_{\mathtt{wh}}((x_1, b_1), (x_2, b_2)) = (\mathtt{Deg}(b_1), \mathtt{Deg}(b_2))$.

From the definition 6,

$$\mathtt{Deg}(a_1) = \frac{x_1}{1 + \mathtt{Deg}(a_2) \times \mathbf{s}(a_1, a_2)} \qquad \mathtt{Deg}(a_2) = \frac{x_2}{1 + \mathtt{Deg}(a_1) \times \mathbf{s}(a_1, a_2)}$$

$$\mathtt{Deg}(b_1) = \frac{x_1}{1 + \mathtt{Deg}(b_2) \times (\mathbf{s}(a_1, a_2) + \alpha)} \qquad \mathtt{Deg}(b_2) = \frac{x_2}{1 + \mathtt{Deg}(b_1) \times (\mathbf{s}(a_1, a_2) + \alpha)}$$

Let us develop the equation of $\mathtt{Deg}(a_1)$:

$$\mathtt{Deg}(a_1) = \frac{x_1}{1 + \mathbf{s}(a_1, a_2) \times \frac{x_2}{1 + \mathtt{Deg}(a_1) \times \mathbf{s}(a_1, a_2)}}$$

$$\iff \mathtt{Deg}(a_1) = \frac{x_1}{\frac{1 + \mathbf{s}(a_1, a_2) \times \mathtt{Deg}(a_1) + \mathbf{s}(a_1, a_2) \times x_2}{1 + \mathbf{s}(a_1, a_2) \times \mathtt{Deg}(a_1)}}$$

$$\iff \mathtt{Deg}(a_1) = \frac{x_1 + \mathbf{s}(a_1, a_2) \times (\mathtt{Deg}(a_1) \times x_1)}{1 + \mathbf{s}(a_1, a_2) \times (\mathtt{Deg}(a_1) + x_2)}$$

In a same way we can develop the equation of $\mathtt{Deg}(b_1)$:

$$\mathtt{Deg}(b_1) = \frac{x_1 + (\mathbf{s}(a_1, a_2) + \alpha) \times (\mathtt{Deg}(b_1) \times x_1)}{1 + (\mathbf{s}(a_1, a_2) + \alpha) \times (\mathtt{Deg}(b_1) + x_2)}.$$

Given that $x_1, x_2 \in]0,1]$ then $\alpha \times \mathtt{Deg}(b_1) \times x_1 < \alpha \times (\mathtt{Deg}(b_1) + x_2)$. Therefore $\mathtt{Deg}(b_1) < \mathtt{Deg}(a_1)$. We can do the same reasoning with a_2 and b_2 and we obtain that $\mathtt{Deg}(b_2) < \mathtt{Deg}(a_2)$. ■

When an argument is dissimilar to all other arguments, then we show that if its initial value is 0, then it will not have any impact on the readjusted values of the other arguments. This property is violated by one of the adjustment functions defined in [9] (see the related work section).

Proposition 4 *For any* $\mathtt{AF}$, $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$, *for all* $a_1, \cdots, a_n, b \in \mathscr{A}$, *for all* $x_1, \cdots, x_n, y \in [0,1]$, *if*

- $\forall i \in \{1, \cdots, n\}$, $\mathbf{s}(a_i, b) = 0$,
- $y = 0$,

then $\mathbf{n}_{\mathtt{wh}}((x_1, a_1), \cdots, (x_n, a_n), (y, b)) = (\mathbf{n}_{\mathtt{wh}}((x_1, a_1), \cdots, (x_n, a_n)), 0)$.

More strongly, we show that an argument having an initial value of 0 and for any similarity with other arguments, this arguments doesn't impact the readjusted values of the other arguments.

Proposition 5 *For any* AF*,* $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$*, for all* $a_1, \cdots, a_n, b \in \mathscr{A}$*, for all* $x_1, \cdots, x_n, y \in [0,1]$*, if*

- $y = 0$*,*

then $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n),(y,b)) = (\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n)),0)$.

Proof Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an AF, $a_1, \cdots, a_n, b_1 \in \mathscr{A}$ and $x_1, \cdots, x_n, y \in [0,1]$ such that

- $y = 0$.

From Definition 6 we have $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n)) = (\mathtt{Deg_1}(a_1),\cdots,\mathtt{Deg_1}(a_n)) = \mathtt{Deg_1}_{G'}^{\mathscr{S}_{\mathtt{wh}}}$, where

$$\mathtt{Deg_1}_{G'}^{\mathscr{S}_{\mathtt{wh}}} = \begin{cases} \mathtt{Deg_1}(a_1) = \frac{x_1}{1+\mathtt{Deg_1}(a_2)\times\mathbf{s}(a_1,a_2)+\cdots+\mathtt{Deg_1}(a_n)\times\mathbf{s}(a_1,a_n)} \\ \cdots \\ \mathtt{Deg_1}(a_n) = \frac{x_n}{1+\mathtt{Deg_1}(a_1)\times\mathbf{s}(a_n,a_1)+\cdots+\mathtt{Deg_1}(a_{n-1})\times\mathbf{s}(a_n,a_{n-1})} \end{cases}$$

and $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n),(y,b_1)) = (\mathtt{Deg_2}(a_1),\cdots,\mathtt{Deg_2}(a_n)) = \mathtt{Deg_2}_{G'}^{\mathscr{S}_{\mathtt{wh}}}$, where

$$\mathtt{Deg_2}_{G'}^{\mathscr{S}_{\mathtt{wh}}} = \begin{cases} \mathtt{Deg_2}(a_1) = \frac{x_1}{1+\mathtt{Deg_2}(a_2)\times\mathbf{s}(a_1,a_2)+\cdots+\mathtt{Deg_2}(a_n)\times\mathbf{s}(a_1,a_n)+\mathtt{Deg_2}(b_1)\times\mathbf{s}(a_1,b_1)} \\ \cdots \\ \mathtt{Deg_2}(a_n) = \frac{x_n}{1+\mathtt{Deg_2}(a_1)\times\mathbf{s}(a_n,a_1)+\cdots+\mathtt{Deg_2}(a_{n-1})\times\mathbf{s}(a_n,a_{n-1})+\mathtt{Deg_2}(b_1)\times\mathbf{s}(a_n,b_1)} \\ \mathtt{Deg_2}(b_1) = \frac{y_1}{1+\mathtt{Deg_2}(a_1)\times\mathbf{s}(b_1,a_1)+\cdots+\mathtt{Deg_2}(a_n)\times\mathbf{s}(b_1,a_n)} \end{cases}$$

Given that $y = 0$, $\mathtt{Deg_2}(b_1) = 0$, so for every $i \in \{1,\cdots,n\}$, $\mathtt{Deg_1}(a_i) = \mathtt{Deg_2}(a_i)$. ■

The function $\mathbf{n}_{\mathtt{wh}}$ cannot readjusts a positive value to 0. This means that it does not ignore any attacker when similarities are available. It rather distributes the burden of redundancy among attackers.

Proposition 6 *Let* $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ *be an* AF*,* $a_1, \cdots, a_n \in \mathscr{A}$*,* $x_1, \cdots, x_n \in [0,1]$ *and* $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n)) = (x'_1,\cdots,x'_n)$*. For any* $i \in \{1,\cdots,n\}$*, if* $x_i > 0$*, then* $x'_i > 0$*.*

Proof Let $\mathbf{G} = \langle \mathscr{A}, \mathbf{w}, \mathscr{R}, \sigma, \mathbf{s} \rangle$ be an AF, $a_1, \cdots, a_n \in \mathscr{A}$, $x_1, \cdots, x_n \in [0,1]$ and $\mathbf{n}_{\mathtt{wh}}((x_1,a_1),\cdots,(x_n,a_n)) = (\mathtt{Deg}(a_1),\cdots,\mathtt{Deg}(a_n))$. For any $i \in \{1,\cdots,n\}$, from Definition 6, $\mathtt{Deg}(a_i) = \frac{x_i}{1+X}$ such that $X \in [0,+\infty[$ therefore if $x_i > 0$, then $\mathtt{Deg}(a_i) > 0$. ■

5. Related Work

In [9], the authors proposed three gradual semantics dealing with similarity in argumentation frameworks that are free of weights on attacks. We are interested in comparing the adjustment functions, hence the lack of function **h** is not a problem. However, from the three gradual semantics, one of them (Grouping weighted h-categorizer - GHbs) has not an independent adjustment function, that means this gradual semantics mixed the aggregation function (**g**) with the adjustment function (**n**). That is why we will not compare this method with our own.

The first semantics that we will compare is the Extended weighted h-Categorizer (EHbs) which uses a similarity measure between set of arguments. The principle of its adjustment function used consists in two steps:

1. ordering arguments from the strongest to the weakest ones (depending on their strengths),
2. then after permutation, from each argument the function keeps only the proportion of novelty according to the previous arguments already adjusted.

As described in the background section, there exist different strategies to distribute the similarity. In this function, the similarity is applied on sub-set of arguments according to its rank in the permutation. The first argument of this permutation will for instance keep all its initial weight. Another strategy can be to distribute the diminution on both arguments as done by our $\mathbf{n}_{\mathtt{wh}}$ function (proposition 3). These different strategies of adjustment are relevant for some aggregation functions **g** and not for others. For instance, when **g** is the aggregation function $\mathbf{g}_{\mathtt{max}}$, i.e. returning the maximal value of a set; distributing redundancy will make a significant difference in the evaluation.

Moreover, it can be noted that the way to ordering the attackers is not determinative, i.e. the constraint producing the ranking (only by degree) is not always unique (there may be ties) and these different rankings may produce different adjustments. For instance, if 3 arguments a, b, c have the same degree x but not the same similarity between them then the ordering will change the adjustment.

The second semantics that we will compare is the Readjustment weighted h-Categorizer (RHbs) which uses a binary similarity measure like our semantics. Let us introduce its adjustment function named Readjusted score. This function is based on different averages. We can describe its process in two operations to adjust the degree of an argument a:

1. for each other arguments x, it compute an average adjusted score α between x and a,
2. the final adjusted degree of a is the average of all the average adjusted score α.

We denote by avg the average operator. Formally the definition is the following:

Definition 7 ($\mathbf{n}_{\mathtt{rs}}$) *Let* $a_1, \cdots, a_k \in \mathscr{U}$ *and* $x_1, \cdots, x_k \in [0,1]$. $\mathbf{n}_{\mathtt{rs}}((x_1,a_1), \cdots, (x_k,a_k)) =$

$$\left(\underset{x_i \in \{x_1,\cdots,x_k\} \setminus \{x_1\}}{\mathtt{avg}} \left(\frac{\mathtt{avg}(x_1,x_i) \times (2 - \mathbf{s}(a_1,a_i))}{2} \right), \cdots, \right.$$

$$\left. \underset{x_i \in \{x_1,\cdots,x_k\} \setminus \{x_k\}}{\mathtt{avg}} \left(\frac{\mathtt{avg}(x_k,x_i) \times (2 - \mathbf{s}(a_k,a_i))}{2} \right) \right).$$

$\mathbf{n}_{\mathtt{rs}}() = ()$ *and* $\mathbf{n}_{\mathtt{rs}}((x_1,a_1)) = (x_1)$ *if* $k = 1$.

To compare $\mathbf{n}_{\mathtt{wh}}$ with $\mathbf{n}_{\mathtt{rs}}$ let's come back to the example 1.

Example 1 (Cont) As reminder, $x_1 = 0.04$, $x_2 = 0.3$ and $\mathbf{s}(b_1, b_2) = 1$.
Then $\mathbf{n}_{\mathtt{rs}}((x_1, b_1), (x_2, b_2)) = (0.085, 0.085)$ while $\mathbf{n}_{\mathtt{wh}}((x_1, b_1), (x_2, b_2)) = (0.03, 0.29)$.

Moreover, we propose the new adjustment function $\mathbf{n}_{\mathtt{wh}}$, because the Readjusted score violate some intuitive proposition.

Proposition 7 *The adjustment function $\mathbf{n}_{\mathtt{rs}}$ violates proposition 2, i.e. it does alter the values of the arguments when all the arguments are dissimilar.*

Proof Let $a, b \in \mathscr{U}$ such that $\mathbf{s}(a,b) = 0$ and $x_a = 1, x_b = 0.8$, then $\mathbf{n}_{\mathtt{rs}}((x_a, a), (x_b, b)) = (0.9, 0.9)$. ■

Proposition 8 *The adjustment function $\mathbf{n}_{\mathtt{rs}}$ violates proposition 4, i.e. an argument dissimilar to all other and whose its initial value is 0, can have an impact on the readjusted values of the other arguments.*
In addition, using the aggregation function $\mathbf{g}_{sum}$ there exist $a_1, \cdots, a_n, b \in \mathscr{U}$ and $x_1, \cdots, x_n, y \in [0,1]$ such that:

- $\forall i \in \{1, \cdots, n\}$, $\mathbf{s}(a_i, b) = 0$,
- $y = 0$,
- $\mathbf{g}_{sum}(\mathbf{n}_{\mathtt{rs}}((x_1, a_1), \cdots, (x_n, a_n))) < \mathbf{g}_{sum}(\mathbf{n}_{\mathtt{rs}}((x_1, a_1), \cdots, (x_n, a_n), (y, b)))$.

This means that adding an attacker dissimilar to all other and whose its initial value is 0, can increase the sum of readjusted values of the set of attackers.

Proof Let $a, b, c \in \mathscr{U}$ such that $\mathbf{s}(a,b) = 0.5$, $\mathbf{s}(a,c) = 0$, $\mathbf{s}(b,c) = 0$ and $x_a = 1$, $x_b = 0.8$, $x_c = 0$ then $\mathbf{n}_{\mathtt{rs}}((x_a, a), (x_b, b)) = (0.675, 0.675)$ and $\mathbf{n}_{\mathtt{rs}}((x_a, a), (x_b, b), (x_c, c)) = (0.5875, 0.5375, 0.45)$. Moreover, we have that $0.675 + 0.675 = 1.35 < 1.575 = 0.5875 + 0.5375 + 0.45$. ■

6. Conclusion

The paper extended the general framework for gradual semantics proposed in [11]. The latter defines a gradual semantics with evaluation methods, which are tuples of three aggregation functions. In this paper, we relaxed the constraint that arguments are all dissimilar. We assumed thus the existence of a similarity measure on the set of arguments. We extended the definition of evaluation method by introducing a novel adjustment function. The latter is responsible for taking into account similarity. We also proposed an instance of such function, which is based on the weighted h-Categorizer. Note that the latter is used in the literature for a completely different reason, namely as a gradual semantics. We investigated the properties of the function, and have shown that it can safely be used by a semantics including h-Categorizer itself. This would mean that h-Categorizer can be used as an adjustment function of a semantics and as the semantics itself.

This work can be extended in different directions. One of them is to study adjustment functions more generally in evaluation methods. The objective would be to give the crucial properties of a reasonable adjustment function.

References

[1] Leite J, Martins J. Social Abstract Argumentation. In: IJCAI'11; 2011. p. 2287–2292.

[2] Benferhat S, Dubois D, Prade H. Argumentative inference in uncertain and inconsistent knowledge bases. In: UAI'93; 1993. p. 411–419.

[3] Amgoud L. Evaluation of Analogical Arguments by Choquet Integral. In: Proceedings of the 24th European Conference on Artificial Intelligence, ECAI; 2020. p. (to appear).

[4] Dunne P, Hunter A, McBurney P, Parsons S, Wooldridge M. Weighted argument systems: Basic definitions, algorithms, and complexity results. Artificial Intelligence. 2011;175(2):457–486.

[5] Egilmez S, Martins JG, Leite J. Extending Social Abstract Argumentation with Votes on Attacks. In: Black E, Modgil S, Oren N, editors. Theory and Applications of Formal Argumentation - Second International Workshop, TAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers. vol. 8306 of Lecture Notes in Computer Science. Springer; 2013. p. 16–31.

[6] Amgoud L, David V. Measuring Similarity between Logical Arguments. In: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning KR; 2018. p. 98–107.

[7] Budan P, Martinez V, Budan M, Simari G. Introducing analogy in abstract argumentation. In: Workshop on Weighted Logics for Artificial Intelligence; 2015. .

[8] Amgoud L, Ben-Naim J, Doder D, Vesic S. Acceptability Semantics for Weighted Argumentation Frameworks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI; 2017. p. 56–62.

[9] Amgoud L, Bonzon E, Delobelle J, Doder D, Konieczny S, Maudet N. Gradual Semantics Accounting for Similarity between Arguments. In: Sixteenth International Conference on Principles of Knowledge Representation and Reasoning KR; 2018. p. 88–97.

[10] Besnard P, Hunter A. A logic-based theory of deductive arguments. Artificial Intelligence. 2001;128(1-2):203–235.

[11] Amgoud L, Doder D. Gradual Semantics Accounting for Varied-Strength Attacks. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. AAMAS '19. International Foundation for Autonomous Agents and Multiagent Systems; 2019. p. 1270–1278.

[12] Cayrol C, Lagasquie-Schiex M. Graduality in Argumentation. Journal of Artificial Intelligence Research. 2005;23:245–297.

[13] da Costa Pereira C, Tettamanzi A, Villata S. Changing One's Mind: Erase or Rewind? In: IJCAI'11; 2011. p. 164–171.

[14] Baroni P, Romano M, Toni F, Aurisicchio M, Bertanza G. Automatic evaluation of design alternatives with quantitative argumentation. Argument & Computation. 2015;6(1):24–49.

[15] Rago A, Toni F, Aurisicchio M, Baroni P. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In: KR'16; 2016. p. 63–73.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200495

On Minimality and Consistency Tolerance in Logical Argumentation Frameworks

Ofer ARIELI [a,1] and Christian STRASSER [b]
[a] *School of Computer Science, The Academic College of Tel-Aviv, Israel*
[b] *Institute of Philosophy II, Ruhr University Bochum, Germany*

Abstract. We examine different methods of handling argument consistency and minimality in logical argumentation frameworks, showing that both properties may (and sometimes even should) be omitted from the definition of arguments. In process, we consider the adequacy of attack rules to the underlying logics.

Keywords. logical argumentation, forms of arguments and attacks, consistency preservation, assumption minimization

1. Motivation

A standard way of viewing an argument A in logical (or, deductive) argumentation frameworks is as a pair $A = \langle \mathsf{S}, \psi \rangle$, where ψ (the *conclusion* of A) is a formula that follows, according to the underlying (base) logic, from the set of formulas S (called the *support* of A). Earlier works on the subject concentrated on classical logic (CL) as the base logic, and since the latter is trivialized in the presence of inconsistency, it was usual to assume that S is consistent. In order to keep the support as relevant as possible to the conclusion, S was kept minimal with respect to the subset relation (see [5]). These considerations lead to the following definition of what we call *classical-con-min arguments*.

Definition 1 A CL-*con-min argument* is a pair $A = \langle \mathsf{S}, \psi \rangle$, where S is a CL-consistent and $\subseteq$-minimal finite set of formulas that entails, according to CL, the formula ψ.[2]

Definition 1 is at the heart of many approaches to logic-based argumentation.[3] However, as noted e.g. in [3], the consistency and minimality requirements on the supports of the arguments cause some complications in the construction and the identification of valid arguments, and so they may be lifted. Moreover, in some reasoning contexts non-classical logics may better serve as the underlying logics of the intended argumentation frameworks, and in some cases (e.g., agent-based systems or deontic systems) the standard propositional language should be extended (e.g., with modal operators), which again means that in those cases classical logic is not adequate. Indeed, many approaches

[1] Supported by the Israel Science Foundation, grant No. 550/19.

[2] In order words, if F denotes the falsity operator and $\vdash_{\mathsf{CL}}$ is the consequence relation of classical logic, then S is a finite set of formulas such that $\mathsf{S} \not\vdash_{\mathsf{CL}} \mathsf{F}$, $\mathsf{S} \vdash_{\mathsf{CL}} \psi$, and there is no $\mathsf{S}' \subsetneq \mathsf{S}$ such that $\mathsf{S}' \vdash_{\mathsf{CL}} \psi$.

[3] For more details and references see, e.g., [6,7,13].

to structured argumentation like those that are based on ASPIC systems [16], deductive variation of assumption-based argumentation frameworks [15], sequent-based argumentation [3], and so forth, do not assume anymore that the underlying logic is necessarily classical. Concerning consistency and minimality, in some of these alternatives the handling of these properties is done on the level of the argumentation frameworks themselves, by means of appropriate attack rules, or by posing some restrictions on their applications (see for instance [14]).

In this paper we examine relations between these two approaches for handling minimality and consistency, namely: the one that enforces these properties already on the level of arguments and the other that takes care of them by appropriate attack rules. For the latter we then show how the suitability of the attack rules is affected by the base logic.

2. Preliminaries

For defining logical argumentation frameworks, and arguments in particular, one first has to specify what the underlying logic is.

Definition 2 A (propositional) *logic* is a pair $\mathfrak{L} = \langle \mathscr{L}, \vdash \rangle$, where $\mathscr{L}$ is a propositional language, and $\vdash$ is a (Tarskian, [21]) *consequence relation* for a language $\mathscr{L}$, that is: a binary relation between sets of formulas and formulas in $\mathscr{L}$, satisfying the following conditions: if $\psi \in \mathsf{S}$ then $\mathsf{S} \vdash \psi$ (*reflexivity*); if $\mathsf{S} \vdash \psi$ and $\mathsf{S} \subseteq \mathsf{S}'$ then $\mathsf{S}' \vdash \psi$ (*monotonicity*); and if $\mathsf{S} \vdash \psi$ and $\mathsf{S}', \psi \vdash \phi$ then $\mathsf{S}, \mathsf{S}' \vdash \phi$ (*transitivity*).

In the sequel we shall assume that the language $\mathscr{L}$ contains at least the following (primitive or defined) connectives and constant:

- $\vdash$-*negation* $\neg$, satisfying: $p \not\vdash \neg p$ and $\neg p \not\vdash p$ (for every atomic p),
- $\vdash$-*conjunction* $\wedge$, satisfying: $\mathsf{S} \vdash \psi \wedge \phi$ iff $\mathsf{S} \vdash \psi$ and $\mathsf{S} \vdash \phi$,
- $\vdash$-*disjunction* $\vee$, satisfying: $\mathsf{S}, \phi \vee \psi \vdash \sigma$ iff $\mathsf{S}, \phi \vdash \sigma$ and $\mathsf{S}, \psi \vdash \sigma$,
- $\vdash$-*falsity* F, satisfying: $\mathsf{F} \vdash \psi$ for every formula ψ.[4]

In some cases we shall assume the availability of a *(deductive)* $\vdash$-*implication* satisfying: $\mathsf{S}, \phi \vdash \psi$ iff $\mathsf{S} \vdash \phi \supset \psi$. Then we shall abbreviate $(\phi \supset \psi) \wedge (\psi \supset \phi)$ by $\phi \leftrightarrow \psi$. For a finite set of formulas S we denote by $\bigwedge \mathsf{S}$ (respectively, by $\bigvee \mathsf{S}$) the conjunction (respectively, the disjunction) of all the formulas in S. We shall also denote by $\wp(\mathsf{S})$ (by $\wp_{\mathsf{fin}}(\mathsf{S})$) the set of the (finite) subsets of S. We say that S is $\vdash$-*consistent* if $\mathsf{S} \not\vdash \mathsf{F}$.

The next definition is a generalization of Definition 1 to every propositional logic, and it avoids the consistency and minimality requirements.

Definition 3 Given a logic $\mathfrak{L} = \langle \mathscr{L}, \vdash \rangle$, an $\mathfrak{L}$-*argument* (an *argument* for short) is a pair $A = \langle \mathsf{S}, \psi \rangle$, where S (the support of A) is a finite set of $\mathscr{L}$-formulas and ψ (the conclusion of A) is an $\mathscr{L}$-formula, such that $\mathsf{S} \vdash \psi$. We denote: $\mathsf{Supp}(\langle \mathsf{S}, \psi \rangle) = \mathsf{S}$ and $\mathsf{Conc}(\langle \mathsf{S}, \psi \rangle) = \psi$. Arguments of the form $\langle \emptyset, \psi \rangle$ are called tautological.

Attacks and counter-attacks between arguments are described by the rules in Table 1 (see, e.g., [3,13,20] for further rules).

[4] In particular, F is not a standard atomic formula, since $\mathsf{F} \vdash \neg \mathsf{F}$.

Rule Name	Acronym	Attacking	Attacked	Attack Conditions
Defeat	Def	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2 \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1 \vdash \neg \bigwedge \mathsf{S}_2$
Full Defeat	FullDef	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2, \psi_2 \rangle$	$\psi_1 \vdash \neg \bigwedge \mathsf{S}_2$
Direct Defeat	DirDef	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \{\varphi\} \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1 \vdash \neg \varphi$
Undercut	Ucut	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2 \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1 \vdash \neg \bigwedge \mathsf{S}_2,\ \neg \bigwedge \mathsf{S}_2 \vdash \psi_1$
Full Undercut	FullUcut	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2, \psi_2 \rangle$	$\psi_1 \vdash \neg \bigwedge \mathsf{S}_2,\ \neg \bigwedge \mathsf{S}_2 \vdash \psi_1$
Direct Undercut	DirUcut	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \{\varphi\} \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1 \vdash \neg \varphi, \quad \neg \varphi \vdash \psi_1$
Consistency Undercut	ConUcut	$\langle \emptyset, \neg \bigwedge \mathsf{S}_2 \rangle$	$\langle \mathsf{S}_2 \cup \mathsf{S}_2', \psi_2 \rangle$	
Rebuttal	Reb	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2, \psi_2 \rangle$	$\psi_1 \vdash \neg \psi_2, \quad \neg \psi_2 \vdash \psi_1$
Defeating Rebuttal	DefReb	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2, \psi_2 \rangle$	$\psi_1 \vdash \neg \psi_2$

Table 1. Some attack rules. The support sets of the attacked arguments are assumed to be nonempty (to avoid attacks on tautological arguments).

Logical argumentation frameworks are now defined as follows:

Definition 4 Let $\mathfrak{L} = \langle \mathscr{L}, \vdash \rangle$ be a logic and $\mathscr{A}$ a set of attack rules with respect to $\mathfrak{L}$. Let also S be a set of $\mathscr{L}$-formulas. The *(logical) argumentation framework* for S, induced by $\mathfrak{L}$ and $\mathscr{A}$, is the pair $\mathscr{AF}_{\mathfrak{L},\mathscr{A}}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A}) \rangle$,[5] where $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ is the set of the $\mathfrak{L}$-arguments whose supports are subsets of S, and $\mathit{Attack}(\mathscr{A})$ is a relation on $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \times \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$, defined by $(A_1, A_2) \in \mathit{Attack}(\mathscr{A})$ iff there is some $\mathscr{R} \in \mathscr{A}$ such that A_1 $\mathscr{R}$-attacks A_2 (that is, the pair (A_1, A_2) is an instance of the relation $\mathscr{R}$).

The Dung-style semantics [12] of an argumentation framework and the corresponding entailment relations are defined in the next two definitions.

Definition 5 Let $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A}) \rangle$ be a logical argumentation framework, and let $\mathscr{E} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$. Below, maximality and minimality are taken with respect to the subset relation.

- We say that $\mathscr{E}$ *attacks* an argument A, if there is an argument $B \in \mathscr{E}$ that attacks A (that is, $(B, A) \in \mathit{Attack}$). The set of arguments that are attacked by $\mathscr{E}$ is denoted $\mathscr{E}^+$. We say that $\mathscr{E}$ *defends* A, if $\mathscr{E}$ attacks every argument that attacks A.
- The set $\mathscr{E}$ is called *conflict-free* with respect to $\mathscr{AF}(\mathsf{S})$, if it does not attack any of its elements (i.e., $\mathscr{E}^+ \cap \mathscr{E} = \emptyset$). A set that is maximally conflict-free with respect to $\mathscr{AF}(\mathsf{S})$ is called a *naive extension* of $\mathscr{AF}(\mathsf{S})$. A conflict-free set $\mathscr{E}$ such that $\mathscr{E} \cup \mathscr{E}^+ = \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ is a *stable extension* of $\mathscr{AF}(\mathsf{S})$.
- An *admissible extension* of $\mathscr{AF}(\mathsf{S})$ is a subset of $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ that is conflict-free with respect to $\mathscr{AF}(\mathsf{S})$ and defends all of its elements. A maximally admissible extension of $\mathscr{AF}(\mathsf{S})$ is called a *preferred extension* of $\mathscr{AF}(\mathsf{S})$.
- A *complete extension* of $\mathscr{AF}(\mathsf{S})$ is an admissible extension of $\mathscr{AF}(\mathsf{S})$ that contains all the arguments that it defends. The minimally complete extension of $\mathscr{AF}(\mathsf{S})$ is called the *grounded extension* of $\mathscr{AF}(\mathsf{S})$.[6]

[5] In what follows we shall usually omit the subscripts and write just $\mathscr{AF}(\mathsf{S})$ for $\langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A}) \rangle$.

[6] As is shown in [12, Theorem 25], the grounded extension of $\mathscr{AF}(\mathsf{S})$ is unique. Also, in the same paper it is shown that preferred extensions are maximally complete and that every stable extension is also preferred. For some other facts and definitions of other extensions, see e.g. [4].

We denote by $\mathsf{Adm}(\mathscr{AF}(\mathsf{S}))$ [respectively, by $\mathsf{Cmp}(\mathscr{AF}(\mathsf{S}))$, $\mathsf{Grd}(\mathscr{AF}(\mathsf{S}))$, $\mathsf{Prf}(\mathscr{AF}(\mathsf{S}))$, $\mathsf{Stb}(\mathscr{AF}(\mathsf{S}))$] the set of all the admissible [respectively, the complete, grounded, preferred, stable] extensions of $\mathscr{AF}(\mathsf{S})$.

Definition 6 Let $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$ be a logical argumentation framework, and let $\mathsf{Sem} \in \{\mathsf{Adm}, \mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$. We denote:

- $\mathsf{S} \mid\!\sim^{\mathfrak{L},\mathscr{A}}_{\cup\mathsf{Sem}} \psi$ if there is an argument $\langle \Gamma, \psi\rangle \in \bigcup \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$,
- $\mathsf{S} \mid\!\sim^{\mathfrak{L},\mathscr{A}}_{\cap\mathsf{Sem}} \psi$ if there is an argument $\langle \Gamma, \psi\rangle \in \bigcap \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$,
- $\mathsf{S} \mid\!\sim^{\mathfrak{L},\mathscr{A}}_{\Cap\mathsf{Sem}} \psi$ if for every $\mathscr{E} \in \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$ there is an argument $\langle \Gamma_{\mathscr{E}}, \psi\rangle \in \mathscr{E}$.

In what follows, when the framework is clear from the context, we shall sometimes write $\mathscr{AF}(\mathsf{S}) \mid\!\sim_{\cup\mathsf{Sem}} \psi$ instead of $\mathsf{S} \mid\!\sim^{\mathfrak{L},\mathscr{A}}_{\cup\mathsf{Sem}} \psi$ (and similarly for the other two entailments).

3. Consistency Preservation in Logical Frameworks

In this section we relate the two methods of maintaining inconsistency in logical argumentation frameworks: by posing the consistency restriction of the supports on the arguments (cf. Definition 1) and by using appropriate attack relations between arguments.

Definition 7 Recall from Definition 5, that $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})^{+}$ is the set of arguments that are attacked by some $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$. In what follows we shall also denote this set by S^{+}.

Example 1 The set $\emptyset^{+}$ consists of the arguments that are attacked by tautological arguments (i.e., by those whose support set is empty).

Definition 8 A set of attack is $\emptyset$-*normal* if it excludes attacks on tautological arguments.

Example 2 By their definitions, all the rules in Table 1 are $\emptyset$-normal, since they exclude attacks on arguments with empty support sets (as is indicated in the caption of Table 1). In [direct] undercut and [direct] defeat, this also follows from the attack conditions and in consistency undercut this follows from the form of the attacking and the attacked arguments.

Proposition 1 *Let* $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$ *be a logical argumentation framework for* S*, based on a logic* $\mathfrak{L} = \langle \mathscr{L}, \vdash\rangle$ *and a set* $\mathscr{A}$ *of* $\emptyset$*-normal attack rules. For* $\mathscr{E} \subseteq \emptyset^{+}$ *and* $\mathscr{A}^{\star} \subseteq \mathscr{A}$ *such that* $\mathit{Attack}(\mathscr{A}^{\star}) \subseteq (\mathsf{Arg}_{\mathscr{L}}(\emptyset) \times \mathscr{E})$*, we let* $\mathscr{AF}^{\star}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}, \mathit{Attack}(\mathscr{A} \setminus \mathscr{A}^{\star})\rangle$*. Then* $\mathsf{Sem}(\mathscr{AF}(\mathsf{S})) = \mathsf{Sem}(\mathscr{AF}^{\star}(\mathsf{S}))$ *for every* $\mathsf{Sem} \in \{\mathsf{Adm}, \mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$.

Note 1 Intuitively, the set $\mathscr{E}$ in Proposition 1 consists of the ‘contradictory’ S-based arguments (cf. Example 1) and $\mathscr{A}^{\star}$ consists of the rules that allow to attack the elements in $\mathscr{E}$. What Proposition 1 says, then, is that if ‘contradictory’ arguments are not allowed (as in Definition 1) then attack rules in the style of $\mathscr{A}^{\star}$ may be avoided, and vice-versa: in case that no restrictions are posed on the arguments’ supports (as in Definition 3) then $\mathscr{A}^{\star}$-type attack rules are needed.

Proof. We consider $\mathsf{Sem} \in \{\mathsf{Adm}, \mathsf{Prf}, \mathsf{Stb}\}$, leaving the other cases to the reader.

• Consider $\mathsf{Sem} = \mathsf{Adm}$. Let $\mathscr{H} \in \mathsf{Adm}(\mathscr{AF}(\mathsf{S}))$. We first observe that $\mathscr{H} \subseteq \mathsf{Arg}_{\mathscr{L}}(\mathsf{S}) \setminus \mathscr{E}$. Indeed, if there were an argument $A \in \emptyset^+$ in $\mathscr{H}$, there would be an argument $B \in \mathsf{Arg}_{\mathscr{L}}(\emptyset)$ $\mathscr{A}$-attacking A,[7] and by the $\emptyset$-normality of $\mathscr{A}$ there would not be an attacker of A in $\mathscr{H}$, contradicting the admissibility of $\mathscr{H}$ in $\mathscr{AF}(\mathsf{S})$.

Clearly, $\mathscr{H}$ is conflict-free in $\mathscr{AF}^\star(\mathsf{S})$. Suppose now that there is some $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$ that $(\mathscr{A} \setminus \mathscr{A}^\star)$-attacks some $B \in \mathscr{H}$. Since $\mathscr{H} \in \mathsf{Adm}(\mathscr{AF}(\mathsf{S}))$, there is a $C \in \mathscr{H}$ that $\mathscr{A}$-attacks A. Since $\mathscr{A}$ is $\emptyset$-normal, A has a non-empty support. Since $Attack(\mathscr{A}^\star) \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathscr{E})$ and $A \notin \mathscr{E}$, A also $(\mathscr{A} \setminus \mathscr{A}^\star)$-attacks A. This shows that $\mathscr{H} \in \mathsf{Adm}(\mathscr{AF}^\star(\mathsf{S}))$.

Let now $\mathscr{H} \in \mathsf{Adm}(\mathscr{AF}^\star(\mathsf{S}))$. Clearly, $\mathscr{H} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$. Assume for a contradiction that there are $A, B \in \mathscr{H}$ such that A $\mathscr{A}$-attacks B. By the admissibility of $\mathscr{H}$ in $\mathscr{AF}^\star(\mathsf{S})$, A does not $(\mathscr{A} \setminus \mathscr{A}^\star)$-attack B. Thus, A $\mathscr{A}^\star$-attacks B. However, then $B \in \mathscr{E}$, since $Attack(\mathscr{A}^\star) \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathscr{E})$. This is a contradiction to $\mathscr{H} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$. Thus, $\mathscr{H}$ is conflict-free in $\mathscr{AF}_{\mathfrak{L}}(\mathsf{S})$.

Suppose now that some $B \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ $\mathscr{A}$-attacks some $A \in \mathscr{H}$. If it is an $(\mathscr{A} \setminus \mathscr{A}^\star)$-attack, by the admissibility of $\mathscr{H}$ in $\mathscr{AF}^\star(\mathsf{S})$ there is a $C \in \mathscr{H}$ that $\mathscr{A}$-attacks B. Assume it is an $\mathscr{A}^\star$-attack. Then $A \in \mathscr{E}$, since $\mathrm{Attack}(\mathscr{A}^\star) \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathscr{E})$. This is a contradiction to $\mathscr{H} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$. Thus, $\mathscr{H} \in \mathsf{Adm}(\mathscr{AF}(\mathsf{S}))$.

• Consider $\mathsf{Sem} = \mathsf{Prf}$. This follows immediately from the fact that $\mathsf{Adm}(\mathscr{AF}(\mathsf{S})) = \mathsf{Adm}(\mathscr{AF}^\star(\mathsf{S}))$, since preferred extensions are the maximally admissible ones.

• Consider $\mathsf{Sem} = \mathsf{Stb}$. Let $\mathscr{H} \in \mathsf{Stb}(\mathscr{AF}(\mathsf{S}))$. Assume that $\mathscr{H} \cap \mathscr{E} \neq \emptyset$. Let $A \in \mathscr{H} \cap \mathscr{E}$. Then there is a $B \in \mathsf{Arg}_{\mathfrak{L}}(\emptyset)$ that $(\mathscr{A} \setminus \mathscr{A}^\star)$-attacks A. Since $\mathscr{A}$ is $\emptyset$-normal, there is no $C \in \mathscr{H}$ that $\mathscr{A}$-attacks B. By the stability of $\mathscr{H}$, $B \in \mathscr{H}$, which contradicts the conflict-freeness of $\mathscr{H}$. Thus, $\mathscr{H} \cap \mathscr{E} = \emptyset$ and so $\mathscr{H} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$.

Clearly, $\mathscr{H}$ is $(\mathscr{A} \setminus \mathscr{A}^\star)$-conflict-free since it is $\mathscr{A}$-conflict-free. Suppose that $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus (\mathscr{E} \cup \mathscr{H})$. Then $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{H}$ and so there is a $B \in \mathscr{H}$ that $\mathscr{A}$-attacks A. Since $\mathrm{Attack}(\mathscr{A}^\star) \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathscr{E})$ and $A \notin \mathscr{E}$, B also $(\mathscr{A} \setminus \mathscr{A}^\star)$-attacks A. Thus, $\mathscr{H} \in \mathsf{Stb}(\mathscr{AF}^\star(\mathsf{S}))$.

Suppose now that $\mathscr{H} \in \mathsf{Stb}(\mathscr{AF}^\star(\mathsf{S}))$. Assume for a contradiction that $\mathscr{H}$ is not conflict-free in $\mathscr{AF}(\mathsf{S})$. Thus, there are $A, B \in \mathscr{H}$ such that A $\mathscr{A}$-attacks B. Since $\mathscr{H}$ is conflict-free in $\mathscr{AF}^\star(\mathsf{S})$, A does not $(\mathscr{A} \setminus \mathscr{A}^\star)$-attack B, and so it $\mathscr{A}^\star$-attacks B. Since $\mathrm{Attack}(\mathscr{A}^\star) \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathscr{E})$, $B \in \mathscr{E}$, which contradicts the fact that $\mathscr{H} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$. Thus, $\mathscr{H}$ is conflict-free in $\mathscr{AF}(\mathsf{S})$.

Suppose now that $B \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{H}$. If $B \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$, there is an argument $A \in \mathscr{H}$ that $\mathscr{A}$-attacks B. Otherwise, $B \in \mathscr{E}$, thus there is an $A \in \mathsf{Arg}_{\mathfrak{L}}(\emptyset)$ that $\mathscr{A}$-attacks B. Since $\mathscr{A}$ is $\emptyset$-normal, $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathscr{E}$ and, since $\mathscr{H}$ is stable in $\mathscr{AF}^\star(\mathsf{S})$, $A \in \mathscr{H}$. Altogether, this shows that $\mathscr{H} \in \mathsf{Stb}(\mathscr{AF}(\mathsf{S}))$. □

As a particular case of Proposition 1, we have the following corollary:

Corollary 1 *Let $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), Attack(\mathscr{A}) \rangle$ be a logical argumentation framework for S, based on a logic $\mathfrak{L} = \langle \mathscr{L}, \vdash \rangle$ and a set $\mathscr{A}$ of $\emptyset$-normal attack rules that contains ConUcut. Let also $\mathscr{AF}^{\mathsf{con}}(\mathsf{S}) = \langle \mathsf{Arg}^{\mathsf{con}}_{\mathfrak{L}}(\mathsf{S}), Attack(\mathscr{A}^\star) \rangle$ be a logical argumenta-*

[7] We say that A $\mathscr{A}$-attacks B iff there is some $\mathscr{R} \in \mathscr{A}$ such that A $\mathscr{R}$-attacks B.

tion framework in which $\mathscr{A}^\star = \mathscr{A} - \{\text{ConUcut}\}$ *and* $\mathsf{Arg}^{\mathsf{con}}_{\mathfrak{L}}(\mathsf{S})$ *is the subset of* $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ *that consists only of* $\vdash_{\mathfrak{L}}$*-consistent arguments (i.e, whose supports are* $\vdash_{\mathfrak{L}}$*-consistent). Then* $\mathsf{Sem}(\mathscr{AF}(\mathsf{S})) = \mathsf{Sem}(\mathscr{AF}^{\mathsf{con}}(\mathsf{S}))$ *for every* $\mathsf{Sem} \in \{\mathsf{Adm}, \mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$.

Proof. Follows from Proposition 1 since $\mathit{Attack}(\text{ConUcut}) \subseteq \mathsf{Arg}_{\mathfrak{L}}(\emptyset) \times \mathsf{Arg}^{\mathsf{incon}}_{\mathfrak{L}}(\mathsf{S})$, where $\mathsf{Arg}^{\mathsf{incon}}_{\mathfrak{L}}(\mathsf{S}) = \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}) \setminus \mathsf{Arg}^{\mathsf{con}}_{\mathfrak{L}}(\mathsf{S})$. □

Note 2 The use of ConUcut for attacking arguments that are based on inconsistent supports goes beyond the standard interpretation of inconsistency as in classical logic. For instance, according to logics of formal inconsistency (LFIs, see [9,10]) $\mathsf{S}_1 = \{\psi, \neg\psi\}$ is not considered inconsistent, but rather $\mathsf{S}_2 = \{\psi, \neg\psi, \circ\psi\}$ (where $\circ$ is the consistency operator, thus $\circ\psi$ is intuitively understood as a claim that 'ψ is consistent'). Indeed, when an LFI is the base logic, an argument whose support is S_1 is not ConUcut-attacked, while an argument whose support set contains S_2 *is* ConUcut-attacked (by $\langle \emptyset, \neg(\psi \wedge \neg\psi \wedge \circ\psi)\rangle$). We shall return to this issue in Section 5.

We show that Proposition 1 and Corollary 1 crucially depend on $\mathscr{A}$ being $\emptyset$-normal:

Example 3 We consider classical logic with the premises $\mathsf{S} = \{p \wedge \neg p, s\}$ and with a more radical form of Rebuttal that does not follow the restriction that only arguments with non-empty support may be attacked. Although $\langle p \wedge \neg p, \neg s\rangle$ is ConUcut-attacked by $\langle \emptyset, \neg(p \wedge \neg p)\rangle$, the latter is Rebut-attacked by $\langle p \wedge \neg p, p \wedge \neg p\rangle$ (given our more radical form of Rebuttal). Thus, e.g., the grounded extension will be empty in the presence of Rebuttal, even in the presence of ConUcut. However, after filtering out inconsistent arguments, it is easy to see that $\langle s, s\rangle$ will be an argument in the grounded extension.

Corollary 2 Let $\mathscr{AF}(\mathsf{S})$ and $\mathscr{AF}^\star(\mathsf{S})$ be as in Proposition 1. Then $\mathscr{AF}(\mathsf{S}) \mid\!\sim_{\circ\mathsf{Sem}} \psi$ iff $\mathscr{AF}^\star(\mathsf{S}) \mid\!\sim_{\circ\mathsf{Sem}} \psi$ for every $\circ \in \{\cup, \cap, \Cap\}$ and $\mathsf{Sem} \in \{\mathsf{Adm}, \mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$.[8]

4. Enforcement of Minimal Support

We now turn to the other condition in Definition 1 – subset minimality of the arguments' supports. Our main result is given in Proposition 2. First, some definitions and lemmas.

Definition 9 Given an argumentation framework $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$, a *support ordering* for $\mathscr{AF}(\mathsf{S})$ is a preorder[9] $\preceq$ on the finite subsets of S.[10]

Definition 10 Given a framework $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$, a support ordering $\preceq$ for $\mathscr{AF}(\mathsf{S})$, a set $\mathscr{E} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$, and an argument $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$. We denote:

- $\min_{\preceq}(\mathscr{E}) = \{A \in \mathscr{E} \mid \nexists B \in \mathscr{E} \text{ s.t. } \mathsf{Conc}(B) = \mathsf{Conc}(A) \text{ and } \mathsf{Supp}(B) \prec \mathsf{Supp}(A)\}$,
- $A^{\min}_{\preceq} = \min_{\preceq}(\{B \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{Supp}(A)) \mid \mathsf{Conc}(A) = \mathsf{Conc}(B)\})$, [11]
- $\mathscr{E}^{\min}_{\preceq} = \bigcup\{A^{\min}_{\preceq} \mid A \in \mathscr{E}\}$.

[8] Here we abuse a bit the notations of Definition 6 to emphasize the relations between the frameworks.

[9] I.e., a reflexive and transitive order.

[10] We denote by $\prec$ the strict version of $\preceq$, that is: if $\preceq$ is a preorder on some domain $\mathscr{D}$, then for all $d, d' \in \mathscr{D}$, $d \prec d'$ iff $d \preceq d'$ and $d' \not\preceq d$.

[11] To simplify the notation,we write $A^{\min}_{\preceq}$ instead of $A^{\min}_{\mathfrak{L},\preceq}$.

Thus, $\min_{\preceq}(\mathscr{E})$ filters-out from $\mathscr{E}$ all the arguments whose support is not minimal among the arguments in $\mathscr{E}$, and $\mathscr{E}^{\min}_{\preceq}$ removes from the arguments in $\mathscr{E}$ all the redundant formulas in their supports.

Example 4 Let $\mathfrak{L} = \mathsf{CL}$ (classical logic), $\preceq\, = \,\subseteq$, and $\mathscr{E} = \{p,q \Rightarrow p \vee q\}$. Then $\min_{\preceq}(\mathscr{E}) = \mathscr{E}$ and $\mathscr{E}^{\min}_{\preceq} = \{p \Rightarrow p \vee q,\ q \Rightarrow p \vee q\}$. Also, for $\mathscr{E}' = \mathscr{E} \cup \{p \Rightarrow p \vee q\}$, we have that $\min_{\preceq}(\mathscr{E}') = \{p \Rightarrow p \vee q\}$ and $(\mathscr{E}')^{\min}_{\preceq} = \mathscr{E}^{\min}_{\preceq}$.

Example 5 Obviously, the subset relation $\subseteq$ is the most natural support ordering in our context. However, there are other candidates to be a support ordering $\preceq$, among which are the following:

- For $\Delta, \Gamma \in \wp_{\mathsf{fin}}(\mathsf{S})$ we define $\Delta \preceq_{\vdash} \Gamma$ iff $\Gamma \vdash \bigwedge \Delta$.
- Suppose that S is stratified into a partition $\langle \mathsf{S}_1, \ldots, \mathsf{S}_n \rangle$, where intuitively formulas in S_i are considered more reliable than formulas in S_j when $i > j$.[12] We let $\preceq$ be the lexicographic ordering, i.e., for $\Delta = \langle \Delta_1, \ldots, \Delta_n \rangle$ and $\Gamma = \langle \Gamma_1, \ldots, \Gamma_m \rangle$ (with $\Delta_i, \Gamma_i \in \wp_{\mathsf{fin}}(\mathsf{S}_i)$ for each $1 \leq i \leq \max\{n,m\}$), we define: $\Delta \preceq_{\mathsf{lex}} \Gamma$ iff either $\Delta = \Gamma$, or there is an $1 \leq k \leq \min\{n,m\}$ such that $\Delta_i = \Gamma_i$ for all $i < k$ and $\Delta_k \subsetneq \Gamma_k$.

Note 3 In all the cases of the last example it holds that if $A, B \in \mathsf{Arg}(\mathsf{S})$ have the same conclusion and $\mathsf{Supp}(A) \prec \mathsf{Supp}(B)$, it makes sense to consider B argumentatively more vulnerable, since its support gives more points of attack: Either it contains more formulas ($\preceq\, = \,\subseteq$), or because its support contains stronger logical commitments ($\preceq\, = \,\preceq_{\vdash}$), or because its support contains stronger logical commitments relative to their reliability ($\preceq\, = \,\preceq_{\mathsf{lex}}$). In that sense, the demand of $\preceq$-minimal support from arguments means minimal argumentative vulnerability.

Definition 11 A set of attack rules $\mathscr{A}$ is called $\preceq$-*normal*, if for every $\mathscr{R} \in \mathscr{A}$ the following conditions hold:

1. If A $\mathscr{R}$-attacks B and $\mathsf{Supp}(A') \preceq \mathsf{Supp}(A)$ and $\mathsf{Conc}(A) = \mathsf{Conc}(A')$, then A' $\mathscr{R}$-attacks B.
2. If A $\mathscr{R}$-attacks B and $\mathsf{Supp}(B) \preceq \mathsf{Supp}(B')$ and $\mathsf{Conc}(B) = \mathsf{Conc}(B')$, then A $\mathscr{R}$-attacks B'.

Note 4 The two conditions in Definition 11 resemble rules R_1 and R_2 (respectively) in [1, Definition 12], except that [1] refers only to the supports of the attacking and the attacked arguments, and uses only the subset relation. Also, in R_1 of [1] the condition on the supports are reversed (that is, R_1 refers to attacking super-arguments and R_2 refers to attacked super-arguments). In our case the two conditions assure, respectively, that attacks are closed under $\preceq$-stronger attacking rules and $\preceq$-weaker attacked rules.[13]

The proofs of the next lemmas are omitted due to lack of space.

[12] See [8].

[13] An argument A is a *super-argument* of B, if $\mathsf{Supp}(B) \subseteq \mathsf{Supp}(A)$. If $\mathsf{Supp}(B) \preceq \mathsf{Supp}(A)$ and $\mathsf{Conc}(B) = \mathsf{Conc}(A)$, we say that B is *stronger* than A (or that A is *weaker* than B).

Lemma 1 *Given a logical argumentation framework $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$ and a support ordering $\preceq$ for $\mathscr{AF}(\mathsf{S})$ such that $\mathscr{A}$ satisfies Item 2 of Definition 11. If $\mathscr{E} \in \mathsf{Cmp}(\mathscr{AF}(\mathsf{S}))$ then $\mathscr{E}^{\min}_{\preceq} = \min_{\preceq}(\mathscr{E})$.*

Let $\mathsf{F}_{\mathscr{AF}(\mathsf{S})} : \wp(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})) \to \wp(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}))$ be a function that relates every $\mathscr{E} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ with the set of arguments that are defended by $\mathscr{E}$ in $\mathscr{AF}(\mathsf{S})$. Again, when the context disambiguates we will skip the subscript. The next lemma is easily verified.

Lemma 2 *Given a logical argumentation framework $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$, a support ordering $\preceq$ for $\mathscr{AF}(\mathsf{S})$ for which $\mathscr{A}$ is $\preceq$-normal, and a set $\mathscr{E} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$, it holds that:*

1. $\mathsf{F}(\mathscr{E}) \subseteq \mathsf{F}(\mathscr{E}^{\min}_{\preceq})$,
2. *if* $\mathscr{E}^{\min}_{\preceq} \subseteq \mathscr{E}$ *then* $\mathsf{F}(\mathscr{E}) = \mathsf{F}(\mathscr{E}^{\min}_{\preceq})$, *and*
3. *if* $\mathscr{E} \in \mathsf{Cmp}(\mathscr{AF}(\mathsf{S}))$ *then* $\mathsf{F}(\mathscr{E}) = \mathsf{F}(\mathscr{E}^{\min}_{\preceq}) = \mathscr{E}$.

Lemma 3 *Let $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathrm{Attack}(\mathscr{A})\rangle$ be a logical argumentation framework and $\preceq$ a support ordering for $\mathscr{AF}(\mathsf{S})$. Suppose that $\mathscr{A}$ satisfies Item 2 of Definition 11. If $\mathscr{E} \subseteq \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ defends $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ then $\mathscr{E}$ defends any $B \in A^{\min}_{\preceq}$.*

The next proposition relates the extensions of a logical argumentation framework (with $\preceq$-normal set of attack rules) and the extensions of the corresponding framework, in which the arguments' supports are minimized.

Proposition 2 *Let $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), \mathit{Attack}(\mathscr{A})\rangle$ be a logical argumentation framework for S, $\preceq$ a support ordering for $\mathscr{AF}(\mathsf{S})$, and $\mathscr{A}$ a $\preceq$-normal set of attack rules. We denote:*

$$\mathit{Attack}^{\min}_{\preceq}(\mathscr{A}) = \mathit{Attack}(\mathscr{A}) \cap \big(\min_{\preceq}(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})) \times \min_{\preceq}(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}))\big),$$
$$\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}) = \langle \min_{\preceq}(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})), \mathit{Attack}^{\min}_{\preceq}(\mathscr{A})\rangle.$$

For every $\mathsf{Sem} \in \{\mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$ we have: $\mathscr{E}' \in \mathsf{Sem}(\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}))$ iff there is $\mathscr{E} \in \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$ such that $\mathscr{E}' = \mathscr{E}^{\min}_{\preceq}$, iff there is $\mathscr{E} \in \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$ such that $\mathscr{E}' = \min_{\preceq}(\mathscr{E})$. Moreover, the extensions $\mathscr{E}$ in the second and the third conditions are the same for every $\mathscr{E}'$, namely $\mathscr{E} = \mathsf{F}_{\mathscr{AF}(\mathsf{S})}(\mathscr{E}')$.

Proof. Suppose that $\mathscr{E}' \in \mathsf{Cmp}(\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}))$, and let $\mathscr{E} = \mathsf{F}_{\mathscr{AF}(\mathsf{S})}(\mathscr{E}')$ be the set of all arguments in $\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ that are defended by $\mathscr{E}'$ in $\mathscr{AF}(\mathsf{S})$. We first show that $\mathscr{E}' \subseteq \mathscr{E}$. Suppose that some $A \in \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})$ attacks $B \in \mathscr{E}'$. By the $\preceq$-normality of $\mathscr{A}$, any $A' \in A^{\min}_{\preceq}$ attacks B. Thus, there is a $C \in \mathscr{E}'$ that attacks A' and by the $\preceq$-normality of $\mathscr{A}$ it also attacks A. Thus, $\mathscr{E}'$ defends every $B \in \mathscr{E}'$ and thus $\mathscr{E}' \subseteq \mathscr{E}$. By Lemma 3, $\mathscr{E}' = \mathscr{E}^{\min}_{\preceq}$. By Item 2 of Lemma 2, $\mathsf{F}_{\mathscr{AF}(\mathsf{S})}(\mathscr{E}) = \mathsf{F}_{\mathscr{AF}(\mathsf{S})}(\mathscr{E}') = \mathscr{E}$.

We still have to show that $\mathscr{E}$ is conflict-free. Assume for a contradiction that there are $A, B \in \mathscr{E}$ for which A attacks B. Thus, any $A' \in A^{\min}_{\preceq}$ attacks B by the $\preceq$-normality of $\mathscr{A}$. However, since $\mathscr{E}'$ defends B there is a $C \in \mathscr{E}'$ that attacks A'. Since $A' \in \mathscr{E}'$, this contradicts the conflict-freeness of $\mathscr{E}'$.

Thus, $\mathscr{E} \in \mathsf{Cmp}(\mathscr{AF}(\mathsf{S}))$. By Lemma 1, $\mathscr{E}^{\min}_{\preceq} = \min_{\preceq}(\mathscr{E}) = \mathscr{E}'$.

Suppose now that $\mathscr{E} \in \mathsf{Cmp}(\mathscr{AF}(\mathsf{S}))$. Consider $\mathscr{E}' = \min_{\preceq}(\mathscr{E})$. By Lemma 1, $\mathscr{E}' = \mathscr{E}^{\min}_{\preceq}$. Hence, $\mathscr{E}' \subseteq (\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}))^{\min}_{\preceq}$. Clearly, $\mathscr{E}'$ is conflict-free since $\mathscr{E}$ is conflict-free. By (Item 3 of) Lemma 2, $\mathsf{F}(\mathscr{E}) = \mathsf{F}(\mathscr{E}') = \mathscr{E}'$, and so $\mathsf{F}(\mathscr{E}')^{\min}_{\preceq} = \mathscr{E}'^{\min}_{\preceq} = \mathscr{E}'$. Thus, $\mathscr{E}' \in \mathsf{Cmp}(\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}))$.

Since the grounded (respectively, preferred) semantics concerns $\subseteq$-minimal (respectively, $\subseteq$-maximal) complete extensions, the proof immediately generalizes for these semantics. The proof for the stable semantics is left to the reader. □

By Proposition 2 we get the following corollaries:

Corollary 3 *Let* $\mathscr{AF}(\mathsf{S}) = \langle \mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}), Attack(\mathscr{A})\rangle$ *be a logical argumentation framework for* S*, induced by a logic* $\mathfrak{L}$*,* $\preceq$ *a support ordering for* $\mathscr{AF}(\mathsf{S})$*, and* $\mathscr{A}$ *a set of* $\preceq$*-normal attack rules. Let also* $\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}) = \langle \min_{\preceq}(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S})), Attack^{\min}_{\preceq}(\mathscr{A})\rangle$ *be a logical argumentation framework as defined in Proposition 2. Then for every* $\mathsf{Sem} \in \{\mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$ *it holds that* $\mathsf{Sem}(\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}))$ *consists of the* Sem*-extensions in* $\mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$*, restricted to the elements in* $(\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}))^{\min}_{\preceq}$*, namely:* $\mathscr{E}^{\min}_{\preceq} \in \mathsf{Sem}(\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}))$ *iff there is an extension* $\mathscr{E} \in \mathsf{Sem}(\mathscr{AF}(\mathsf{S}))$ *and* $\mathscr{E}^{\min}_{\preceq} = \mathscr{E} \cap (\mathsf{Arg}_{\mathfrak{L}}(\mathsf{S}))^{\min}_{\preceq}$.

Like the case of consistency preservation (cf. Corollary 2), we have:

Corollary 4 Let $\mathscr{AF}(\mathsf{S})$, and $\mathscr{AF}^{\min}_{\preceq}(\mathsf{S})$ be as in Proposition 2. Then for every $\circ \in \{\cup, \cap, \Cap\}$ and $\mathsf{Sem} \in \{\mathsf{Cmp}, \mathsf{Grd}, \mathsf{Stb}, \mathsf{Prf}\}$ it holds that $\mathscr{AF}(\mathsf{S}) \mathrel{|\!\sim}_{\circ\mathsf{Sem}} \psi$ iff $\mathscr{AF}^{\min}_{\preceq}(\mathsf{S}) \mathrel{|\!\sim}_{\circ\mathsf{Sem}} \psi$.[14]

5. Attack Rules, Revisited

The previous sections show that the handling of inconsistency and minimality in logical argumentation frameworks may be shifted from arguments to the attack rules. Apart of the obvious advantage of a considerable simplification in the construction and the identification of valid arguments (and so, e.g., proof systems may be incorporated for building arguments from simpler arguments, or for searching for counterarguments given a certain argument; See [3]), we believe that representing these consideration is more appropriate in the rule-based level (Indeed, in real-life arguments are not always based on minimal evidence, avoiding inconsistency sometimes means lose of information, etc).

The use of attack rules for maintaining inconsistency and conflicts among arguments should be taken with care, though, especially when non-classical logics are used as the base logic of the framework. In this section we consider some cases in point.

A. Consistency Undercut Corollary 1 indicates that, among others, ConUcut may replace the support consistency requirement. However, in some base logics the use of ConUcut may not be appropriate or even meaningful. This may happen mainly due to the following reasons:

[14] Again, here we abuse a bit the notations in Definition 6 to emphasize how the argumentation frameworks are related.

- *Problems with the attacking arguments*: Consider, for instance, Kleene's 3-valued logic with the connectives $\neg, \wedge, \vee$ (and their usual 3-valued interpretations). This logic has no valid tautological arguments, because in Kleene's logic no formula follows from the emptyset. This means that Consistency Undercut is not applicable in such a logic.
- *Problems with the attacked arguments*: For instance, in Priest's 3-valued logic LP [17,18] with the connectives $\neg, \wedge, \vee$ every set is satisfiable, thus, again. the use of Consistency Undercut is questionable.

Dunn-Belnap's four-valued logic of first-degree entailment (FDE), combining Kleene's logic and LP, suffers from both problems, namely it does not have tautological arguments and every set is satisfiable. However, if the language of $\neg, \wedge, \vee$ is extended with a proper implication connective ($\supset$, see [2]), both tautological and contradictory (unsatisfiable) arguments may be introduced, in which case it makes sense to incorporate consistency undercut.

B. [Direct, Full] Defeat It may happen that certain attack rules need to be adjusted to specific base logics. We demonstrate this with the logics of formal (in)consistency (LFIs), mentioned in Note 2, and the [direct, full] defeat attack rules (see Table 1). According to these rules, the argument $\langle\{\neg\psi\}, \neg\psi\rangle$ should attack $\langle\{\psi\}, \psi\rangle$. However, for frameworks that are based on LFIs such an attack is more problematic, unless ψ is known to be consistent (i.e., $\circ\psi$ can be inferred).

In the presence of a propositional constant F for falsity, a reformulation of the attack condition of [Full] Defeat could be that $\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$, as indicated in Table 2[15]

Rule Name	Acronym	Attacking	Attacked	Attack Condition
Inconsistency Defeat	IncDef	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2 \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$
Inconsistency Full Defeat	IncFullDef	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \mathsf{S}_2, \psi_2 \rangle$	$\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$
Inconsistency Direct Defeat	IncDirDef	$\langle \mathsf{S}_1, \psi_1 \rangle$	$\langle \{\varphi\} \cup \mathsf{S}_2', \psi_2 \rangle$	$\psi_1, \varphi \vdash \mathsf{F}$

Table 2. Attacks by defeat, revisited (again, we assume that supports of the attacked arguments are nonempty).

Note that the revised conditions in the rules of Table 2 avoid the use of conjunction and are suitable for LFI as well: While according to LFI $\langle\{\neg\psi\}, \neg\psi\rangle$ should *not* attack $\langle\{\psi\}, \psi\rangle$ (although $\neg\psi \vdash \neg\psi$), the argument $\langle\{\neg\psi\}, \neg\psi\rangle$ *can* be used for attacking, by inconsistency [full] defeat, the argument $\langle\{\circ\psi, \psi\}, \psi\rangle$, and the latter attack is perfectly justifiable in the context of any LFI, since the attacked argument is based on the assumption that not only its conclusion ψ holds, but it is also consistent.

One may think of several variations the rules in Table 2, following different intuitions. Below are some options:

Intuition 1: Attack based on a consistency assumption of the attacker.

In this case, e.g., $\langle\{\circ p, p\}, p\rangle$ should attack $\langle\neg p, \neg p\rangle$, but not vice versa.

[15] In logics with a conjunction and where the usual contraposition law holds, or when the negation is defined by $\neg\phi = \phi \supset \mathsf{F}$ for a deductive implication $\supset$, this reformulation is even equivalent to the original one.

Intuition 2: Attack based on a consistency conclusion of the attacker.
According to this intuition, $\langle\{\circ p, p\}, \circ p \wedge p\rangle$ attacks $\langle\neg p, \neg p\rangle$, but not vice versa. Here, $\langle\{\circ p, p\}, p\rangle$ should *not* attack $\langle\neg p, \neg p\rangle$.

Intuition 3: Attack based on a consistency assumption of the attacked argument.
This time $\langle\neg p, \neg p\rangle$ attacks $\langle\{\circ p, p\}, p\rangle$, but not vice versa.

The intuitions above may be captured by extending the conditions of the rules of Table 2. For instance, variations of inconsistency full defeat may be the following:

Variation for Intuition 1: $\langle \mathsf{S}_1, \psi_1\rangle$ attacks $\langle \mathsf{S}_2, \psi_2\rangle$ iff $\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$ and $\mathsf{S}_1 \vdash \circ\bigwedge\mathsf{S}_1$.
Variation for Intuition 2: $\langle \mathsf{S}_1, \psi_1\rangle$ attacks $\langle \mathsf{S}_2, \psi_2\rangle$ iff $\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$ and $\psi_1 \vdash \circ\psi_1$.
Variation for Intuition 3: $\langle \mathsf{S}_1, \psi_1\rangle$ attacks $\langle \mathsf{S}_2, \psi_2\rangle$ iff $\psi_1, \mathsf{S}_2 \vdash \mathsf{F}$ and $\mathsf{S}_2 \vdash \circ\bigwedge\mathsf{S}_2$.

The additional condition in each case above just expresses the consistency assumption of the corresponding intuition. In these conditions $\circ\psi$ is intuitively read by 'ψ is $\vdash$-consistent'. In LFI, $\circ$ is a primitive connective, while in other logics it may serve as a defined connective (e.g., $\neg(\psi \wedge \neg\psi)$).

Note 5 (should minimality be enforced?) The examples in this section provide another reason to avoid the minimality requirement in Definition 1: For instance, the support set of $A = \langle\{\psi, \circ\psi\}, \psi\rangle$ is *not* minimal, as indeed $\circ\psi$ is not necessary for the conclusion of the argument, but it *is* necessary for enabling the above attack variation for Intuition 1, of A on $B = \langle\{\neg\psi\}, \neg\psi\rangle$ (thus refuting the latter).[16]

C. [Direct, Full] Undercut and [Defeating] Rebuttal When the conditions in terms of negation are traded by consistency requirements, undercut rules coincide with the corresponding defeat rules. Regarding the rebuttal rules, conditions in the spirit of the previous section could be that the conclusions of the attacking and the attacked arguments are mutually inconsistent, that is: $\psi_1, \psi_2 \vdash \mathsf{F}$. Again, variations of the rules may involve extra conditions, expressing e.g. further consistency assumptions.

6. Conclusion and Further Work

We have shown that logical argumentation frameworks need not be artificially restricted to arguments with minimal supports and that inconsistent arguments may not be filtered out, even in cases that the underlying logic is not trivialized in the presence of inconsistency. Moreover, we have considered some cases in which the attack rules are not faithful to the consistent and/or minimized support assumption, and some reformulations in terms of related conditions are introduced.

The interplay between the nature of the underlying logic and the formulation of the attack rules has already been considered in the literature (see, e,.g., [11] and [19]). The rewriting of the attack rules in Section 5 imply that attacks may reflect considerations that are not encoded by the pure logical consequences depicted by the arguments. For instance, the reason for the attack according to Intuition 1 in Section 5 is not sufficiently

[16] According to this attack rule $\langle\{\neg\psi\}, \neg\psi\rangle$ is also attacked by $\langle\{\psi, \circ\psi\}, \psi \wedge \circ\psi\rangle$, which meets the minimality criterion, but the latter assumes the availability of a conjunction, while $\langle\{\psi, \circ\psi\}, \psi\rangle$ holds only by reflexivity and monotonicity.

explicated by the conclusion of the attacking argument, since the consistency constraint is not contained in it. Thus, a logical condition only in terms of entailments by the latter (as expressed by the defeat rules) won't be enough in this case. This brings up a new bunch of questions, such as if (and how) it is possible to reformulate specific attack rules to preserve basic properties, such as support minimization, without violating the intended argumentation semantics. This remains a topic for future work.

References

[1] Leila Amgoud. Postulates for logic-based argumentation systems. *International Journal of Approximate Reasoning*, 55(9):2028–2048, 2014.

[2] Ofer Arieli and Arnon Avron. The value of the four values. *Artificial Intelligence*, 102(1):97–141, 1998.

[3] Ofer Arieli and Christian Straßer. Sequent-based logical argumentation. *Argument & Computation*, 6(1):73–99, 2015.

[4] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[5] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1–2):203–235, 2001.

[6] Philippe Besnard and Anthony Hunter. Argumentation based on classical logic. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 133–152. Springer, 2009.

[7] Philippe Besnard and Anthony Hunter. A review of argumentation based on deductive arguments. In *Handbook of Formal Argumentation*, pages 437–484. College Publications, 2018.

[8] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In Natesa Sridharan, editor, *Proceedings of IJCAI'89*, pages 1043–1048. Morgan Kaufmann, 1989.

[9] Walter Carnielli, Marcelo Coniglio, and Joao Marcos. Logics of formal inconsistency. In *Handbook of Philosophical Logic*, volume 14, pages 1–95. Springer, 2007. Second edition.

[10] Walter Carnielli and Joao Marcos. A taxonomy of C-systems. In *Paraconsistency: The Logical Way to the Inconsistent*, Lecture Notes in Pure and Applied Mathematics, pages 1–94. Marcel Dekker, 2002.

[11] Esther Anna Corsi and Christian Fermüller. Logical argumentation principles, sequents, and nondeterministic matrices. In *Proceedings of LORI'17*, pages 422–437. Springer, 2017.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[13] Nikos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9–10):1479–1497, 2011.

[14] Diana Grooters and Henry Prakken. Two aspects of relevance in structured argumentation: Minimality and paraconsistency. *Journal of Artificial Inteligence Research*, 56:197–245, 2016.

[15] Jesse Heyninck and Ofer Arieli. Simple contrapositive assumption-based frameworks. *Journal of Approximate Reasoning*, 121:103–124, 2020.

[16] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[17] Graham Priest. Logic of paradox. *Journal of Philosophical Logic*, 8(1):219–241, 1979.

[18] Graham Priest. Reasoning about truth. *Artificial Intelligence*, 39(2):231–244, 1989.

[19] Chenwei Shi, Sonja Smets, and Fernando R. Velázquez-Quesada. Beliefs supported by binary arguments. *Journal of Applied Non-Classical Logics*, 28(2–3):165–188, 2018.

[20] Christian Straßer and Ofer Arieli. Normative reasoning by sequent-based argumentation. *Journal of Logic and Computation*, 29(3):387–415, 2019.

[21] Alfred Tarski. *Introduction to Logic*. Oxford University Press, 1941.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200496

Timed Abstract Dialectical Frameworks: A Simple Translation-Based Approach

Ringo BAUMANN [a], Maximilian HEINRICH [a]
[a] *{baumann, mheinrich}@informatik.uni-leipzig.de*

Abstract. Abstract dialectical frameworks (ADFs) are one of the most powerful generalization of classical Dung-style AFs. In this paper we show how to use ADFs if we want to deal with acceptance conditions changing over time. We therefore introduce so-called *timed abstract dialectical frameworks (tADFs)* which are essentially ADFs equipped with time states. Beside a precise formal definition of tADFs and an illustrating example we prove that Kleene's three-valued logic $\mathcal{K}_3$ facilitate the evaluation of acceptance functions if we do not allow multiple occurrences of atoms.

Keywords. Abstract Dialectical Frameworks, Time, Three-valued Logics

Introduction

Argumentation has become one of the major fields within AI over the last two decades [1,2]. In particular, Dung's abstract argumentation frameworks (AFs) are a by now widely used formalism [3]. Main reasons for this success story are the simplicity of AFs and the plethora of existing semantics [4], the ability to reconstruct mainstream nonmonotonic formalisms [3] as well as their potential to be used as core method in advanced argumentation formalisms [5,6]. However, through the years the community realized that the limited expressive capability of AFs, namely the option of single attacks only, reduce their suitability as right target systems for more complex applications [7]. Therefore a number of additional functionality were introduced encompassing preferences, values, collective attacks, attacks on attacks as well as support relations between arguments [8,9,10,11,12]. One of the most powerful generalizations of Dung AFs, yet staying on the abstract layer, are so-called *abstract dialectical frameworks* (ADFs) [13]. The additional expressive power is achieved by adding acceptance conditions to the arguments which allow for the specification of arbitrary relationships between arguments and their parents in the argument graph.

In this paper we show how to use classical ADFs if we are faced with conditions changing over time. We therefore introduce so-called *timed abstract dialectical frameworks* (tADFs) which are essentially classical ADFs plus time states. In this way we are able to speak about the same statement s at different time points t. For instance, an acceptance condition like $\phi_{s_4} = a_1 \vee a_2 \vee a_3$ encodes that s should be accepted at time point 4 if statement a is at least ones accepted between time points 1 and 3. If the numbers are interpreted as the first months of the year and if s and a are standing for “I am on vacation in France” or “I have a salary increase”, respectively, then ϕ_{s_4} expresses “I will be vacationing in France in April, if I get a salary increase between January and March.”

The paper is organized as follows: Section 1 reviews necessary background regarding ADFs. In Section 2 we proceed with the formal introduction of tADFs and a presentation of useful timed acceptance conditions. Moreover, we give an illustrating example. Section 3 provides two theoretical insights regarding the evaluation of acceptance functions with the help of three-valued logics. Finally, Section 4 discusses related work and give pointers for future work.

1. Background

1.1. Classical ADFs, Information Order and Consensus

The definition of ADFs [14] was motivated by the effort to obtain more expressive power than classical AFs. This is achieved by equipping each argument with a so-called acceptance condition which can be given as a logical formula [15].

Definition 1. *An* abstract dialectical framework *is a tuple $D = (S, \Phi)$ where S is a set of statements and $\Phi = \{\varphi_s \mid s \in S\}$ is a set of propositional formulae.*

The formal definitions of the different semantics are based on three-valued operators which handle two-valued interpretations.

Definition 2. *Let $D = (S, \Phi)$ be an* ADF*. A two-valued resp. three-valued interpretation v for D is a total function $v : S \mapsto \{\mathbf{t}, \mathbf{f}\}$ or $v : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. We use $\mathcal{V}_2^D$ and $\mathcal{V}_3^D$ for the set of all two resp. three valued interpretations for D.*

Next we define the so-called *information order*. It orders the three values $\mathbf{u}$ (undecided), $\mathbf{t}$ (true) and $\mathbf{f}$ (false) based on their information content.

Definition 3. *Let $D = (S, \Phi)$ be an ADF. The* information order *$\leq_i$ over $\{\mathbf{t},\mathbf{f},\mathbf{u}\}$ is the reflexive closure of $<_i$, where $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$. This is generalised for three-valued interpretations for D in a point-wise fashion:*

$$v_1 \leq_i v_2 \text{ if and only if } \forall s \in S : v_1(s) \in \{\mathbf{t}, \mathbf{f}\} \implies v_1(s) = v_2(s).$$

The consensus operator *$\sqcap_i$ assigns $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and $\mathbf{u}$ otherwise.*

Let $\mathrm{u} \in \mathcal{V}_3^D$, s.t. $\mathrm{u}(s) = \mathbf{u}$ for any $s \in S$. Note that for any $v \in \mathcal{V}_3^D, \mathrm{u} \leq_i v$. This means, u is the $\leq_i$-least element in $\mathcal{V}_3^D$. We will call u the *least information interpretation.* Moreover, for $v \in \mathcal{V}_3^D$ we define $[v]_2^D = \{w \in \mathcal{V}_2^D \mid v \leq_i w\}$. This means, $[v]_2^D$ contains all two-valued completions of v.

1.2. Semantics

To define the semantics the approximation fixpoint theory of Denecker, Marek, and Truszczyński [16] has been used.

Definition 4. *Given an ADF $D = (S, \Phi)$. We define $\Gamma_D : \mathcal{V}_3^D \mapsto \mathcal{V}_3^D$ as*

$$\Gamma_D(v) : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\} \text{ with } s \mapsto \sqcap_i\{w(\varphi_s) \mid w \in [v]_2^D\}.$$

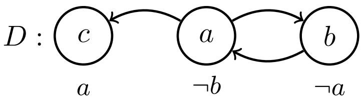
The idea behind the operator is, that based on a given three-valued interpretation, it is checked for every two-valued interpretation with at least as much information whether a consensus on the valuation of the acceptance conditions can be found. If all two valued interpretations consent on either $\mathbf{t}$ or $\mathbf{f}$, then the respective truth value can be assigned by the operator, otherwise it will be evaluated with $\mathbf{u}$. In the following we introduce so-called admissible, complete, preferred and grounded interpretation (abbr. by *adm*, *cmp*, *prf*, *grd*).

Definition 5. *Given an ADF $D = (S, \Phi)$ and $v \in \mathcal{V}_3^D$.*

1. *$v \in adm(D)$ if and only if $v \leq_i \Gamma_D(v)$,*
2. *$v \in cmp(D)$ if and only if $v = \Gamma_D(v)$,*
3. *$v \in prf(D)$ if and only if v is $\leq_i$-maximal in $cmp(D)$,*
4. *$v \in grd(D)$ if and only if v is $\leq_i$-least in $cmp(D)$.*

The definitions above justify the following two subset chains for any ADF D, namely $prf(D) \subseteq cmp(D) \subseteq adm(D)$ as well as $grd(D) \subseteq cmp(D) \subseteq adm(D)$.

Example 1. *Consider the ADF $D = (\{a, b, c\}, \{\phi_a = \neg b, \phi_b = \neg a, \phi_c = a\})$. Let*



us verify that $\{\boldsymbol{u}\} = grd(D)$. It suffices to show that $\boldsymbol{u}$ satisfies $\boldsymbol{u} = \Gamma_D(\boldsymbol{u})$. Note that $\leq_i$-leastness is immediately apparent since $\boldsymbol{u}$ is even $\leq_i$-least in $\mathcal{V}_3^D$. Consider the two-valued interpretation I_1, I_2, s.t. $I_1(a) = I_1(b) = I_1(c) = \mathbf{t}$ and $I_2(a) = I_2(b) = I_2(c) = \mathbf{f}$. We obtain $I_1(\phi_a) \sqcap_i I_2(\phi_a) = \mathbf{u}$ since $I_1(\phi_a) = I_1(\neg b) = \mathbf{f}$ and $I_2(\phi_a) = I_2(\neg b) = \mathbf{t}$. Analogously, one may easily check that $I_1(\phi_b) \sqcap_i I_2(\phi_b) = \mathbf{u}$ and $I_1(\phi_c) \sqcap_i I_2(\phi_c) = \mathbf{u}$ justifying $\boldsymbol{u} = \Gamma_D(\boldsymbol{u})$. The other semantics are given as $adm(D) = \{v_1, v_2, v_3, v_4, \boldsymbol{u}\}$, $cmp(D) = \{v_1, v_3, \boldsymbol{u}\}$, $prf(D) = \{v_1, v_3\}$ with $v_1 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{t}\}$, $v_2 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{u}\}$, $v_3 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{f}\}$ and $v_4 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{u}\}$.

2. Temporal Aspects and Timed ADFs

2.1. Timed Abstract Dialectical Framework

The classical definition of ADFs does not provide one with temporal notions. However, in daily life we are often faced with statements/laws which are valid for a certain time only or depend on the past development, e.g. "You can continue working in the company as long as the Brexit is not delivered." or "From the beginning of next year it will be not allowed to build a nightclub near a residential area.". In order to encode statements like the ones before we need to be able to distinguish between different time states related via a certain ordering. In this very first paper we decided to keep things as simple as possible. Nevertheless, we will see that this approach is powerful enough to model many frequently occuring temporal restrictions. More precisely, a *timed abstract dialectical framework* (tADF) is a classical ADF equipped with a countable set T of time states. We assume that this set is totally ordered, i.e. there is a binary relation $\leq$ over T which is antisymmetric, transitive and connex. Many times T will simply be a subset of the first natural numbers with the inherited standard ordering. Hereby, a certain time state n might stand for an hour, a day, a week or a month or whatever granularity is needed. In this way we are able to speak about the same statement s at different time points t in the future, denoted as s_t. Accordingly, we will have timed acceptance conditions ϕ_{s_t} for any statement s at any time point t.

Definition 6. *A* timed abstract dialectical framework *(for short,* tADF*) is a tuple $D = (S, T, \Phi)$ where S is a set of statements, T total ordered set of time states and $\Phi = \{\varphi_{s_t} \mid s \in S, t \in T\}$ is a set of propositional formulae, one for each statement $s \in S$ and time state $t \in T$.*

In tADFs we treat each argument at each time step as one single classical statement. This means, a tADF with n statements and m time states corresponds to a classical ADF with $n \cdot m$ statements. Moreover, the definition of tADFs allows us to apply the standard semantics of classical ADFs (cf. Example 2).

2.2. Temporal Acceptance Functions

To facilitate the use of tADFs we introduce additional temporal shorthands, which can be used for the corresponding acceptance conditions. Note that any shorthand can be retranslated to classical propositional logic. Given $D = (S, T, \Phi)$ and statements $a, c \in S$ as well as a time interval $[i, j] \subseteq T$.

1. $\varphi_{c_t} = a^{[i,j]}_{\geq 1} := \bigvee\limits_{i \leq k \leq j} a_k.$
   This formula expresses that c should be accepted at time state t, if a is **at least ones accepted** in $[i, j]$. Hence, a supports c at least ones inbetween time states i and j.
2. $\varphi_{c_t} = a^{[i,j]}_{\geq n} := \bigvee\limits_{\substack{\{k_1,\ldots,k_n\} \subseteq [i,j] \\ |\{k_1,\ldots,k_n\}|=n}} a_{k_1} \wedge \ldots \wedge a_{k_n}.$
   This formula expresses that c should be accepted at time state t, if a is **at least n-times accepted** in $[i, j]$. This means, a supports c at least n-times during the time interval $[i, j]$.
3. $\varphi_{c_t} = a^{[i,j]}_{\leq n} := \neg(a^{[i,j]}_{\geq n+1}) = \bigwedge\limits_{\substack{\{k_1,\ldots,k_{n+1}\} \subseteq [i,j] \\ |\{k_1,\ldots,k_{n+1}\}|=n+1}} \neg a_{k_1} \vee \ldots \vee \neg a_{k_{n+1}}.$
   This formula expresses that c should be accepted at time state t, if a is **at most n-times accepted** in $[i, j]$. This means, an n-fold acceptance of a during the time interval $[i, j]$ prevents the acceptance of c.
4. $\varphi_{c_t} = a^{[i,j]}_{\leq 1} := \neg(a^{[i,j]}_{\geq 2}) = \bigwedge\limits_{\substack{\{k_1,k_2\} \subseteq [i,j] \\ |\{k_1,k_2\}|=2}} \neg a_{k_1} \vee \neg a_{k_2}.$
   For the sake of completeness we also present an important instantiation of the timed acceptance formula above, namely $a^{[i,j]}_{\leq 1}$ expressing that c should be accepted at time state t, if a is **at most ones accepted** in $[i, j]$.
5. $\varphi_{c_t} = a^{[i,j]}_{=n} := \varphi_{c_t} = a^{[i,j]}_{\leq n} \wedge a^{[i,j]}_{\geq n}$
   This formula expresses that c should be accepted at time state t, if a is **exactly n-times accepted** in $[i, j]$.

A timed ADF as well as the above introduced shorthands are illustrated in the following example.

Example 2. *Suppose that Charles is making plans for the first months of the new year. He will spend his vacation (v) in France in April if he gets a salary raise (s) in the months before the vacation. In order to get to the desired location he would like to take a plane (p). Unfortunately, such a flight line (l) is currently only planned but Charles knows that it will be introduced between March and May. If no flight is available, he will take the train (t).*

This example can be therefore represented as a tADF $D = (S, T, \Phi)$ where $S = \{v, s, t, p, l\}$ and $T = \{1, 2, 3, 4, 5\}$ (cf. Figure 1). Here, any time state $n \in T$ corresponds to the n^{th} month of the year as expected. The acceptance functions are listed in Table 2. For instance, the formula φ_{l_4} expresses that the flight line will be set up in April, if it is neither introduced in March, nor in May. φ_{l_4} supports vacation in France provided that Charles received at least one raise in the first three months of the year and if a train or plane goes there. Moreover, the condition φ_{t_4} encodes that Charles will take the train if there is no plane available in April and finally, φ_{p_4} expresses that Charles will take an airplane if the flight connection has been established previously and if he is not traveling by train. Salary increases are possible for any month and do not depend on other events. Consequently, $\varphi_{s_i} = s_i$ for any $i \in \{1, 2, 3, 4, 5\}$.

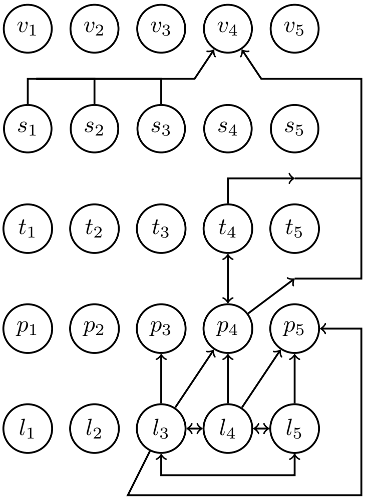


Figure 1. The tADF D

a_t	φ_{a_t}
v_1, v_2, v_3, v_5	$\bot$
v_4	$s_{\geq 1}^{[1,3]} \wedge (p_4 \vee t_4)$
t_1, t_2, t_3, t_5	$\top$
t_4	$\top \wedge \neg p_4 \equiv \neg p_4$
p_1, p_2	$\bot$
p_3	l_3
p_4	$l_{\geq 1}^{[3,4]} \wedge \neg t_4$
p_5	$l_{\geq 1}^{[3,5]}$
l_1, l_2	$\bot$
l_3	$\neg \left(l_{\geq 1}^{[4,5]} \right)$
l_4	$\neg l_3 \wedge \neg l_5$
l_5	$\neg \left(l_{\geq 1}^{[3,4]} \right)$
s_i	s_i

Table 1. Acceptance functions of D

For the evaluation of the tADFs D we use classical ADF semantics. In the following we stick to preferred interpretations as they maximize the information content which appears desirable for the planning context. Table 2 shows 8 out of forty preferred interpretations[1] *of the tADF D. Any interpretation describes a possible scenario. The selected interpretations agree on the availability of the plane in May since for any considered scenario the flight line was only introduced in*

[1] All preferred interpretation can be found under `https://github.com/kmax-tech/ADF`.

May meaning that Charles has to take the train in April in order to get to France. The first interpretation v_1 expresses that the vacation cannot take place since no salary increase happened in the months before. In any other interpretations one or more salary increases happened implying that Charles can take his vacation.

$prf(D)$	l_3	l_4	l_5	p_3	p_4	p_5	s_1	s_2	s_3	t_4	v_4
v_1	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**f**	**f**	**t**	**f**
v_2	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**
v_3	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**t**	**f**	**t**	**t**
v_4	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**t**	**t**	**t**	**t**
v_5	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**f**	**f**	**t**	**t**
v_6	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**f**	**t**	**t**	**t**
v_7	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**	**f**	**t**	**t**
v_8	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**	**t**	**t**	**t**

Table 2. Selected preferred interpretations of D.

3. Evaluation of Acceptance Functions and Three-Valued Logics

In order to facilitate the use of (t)ADFs, we developed a Python script[2], which enables an easy calculation of the desired semantics. During creation of the script the questions occurred, whether the computational expensive calculation of the gamma operator can be somehow simplified. According to Definition 4 the operator takes a three-valued interpretation v and outputs a three-valued one v'. More precisely, for any statement s we have to evaluate the corresponding acceptance function φ_s w.r.t. all two-valued completions of v. Now, applying the consensus operator on these two-valued outputs leaves us with the assignment to s under v'. The idea was to use a three-valued logic $\mathcal{L}_3$, s.t. the evaluation of φ_s can be done directly in $\mathcal{L}_3$ without any computation of two-valued completions and the use of the consensus operator. The following theorem shows that this endeavour is doomed to failure.

Theorem 1. *There is no truth-functional three-valued logic $\mathcal{L}_3$, s.t. for any propositional formula φ and any three-valued interpretation v:*

$$v^{\mathcal{L}_3}(\varphi) = \sqcap_i \{w(\varphi) \mid w \in [v]_2\}.$$

The decisive point for the impossibility of using a three-valued logic in general is that two-valued completions of parts of a composed formula cannot be considered independently. However, such behaviour can be enforced if considering acceptance conditions where each atom appears at most ones. We therefore define the following fragment of classical propositional logic. Let $\mathcal{A} = \{a, b, c, ...\}$ be the set of atomic formulas and $\sigma(\varphi)$ the set of all atoms occuring in φ, e.g. for $\varphi = a \vee \neg a$ we have $\sigma(\varphi) = \{a\}$.

Definition 7. *The set $\mathcal{F}$ is defined inductively as:*

1. $\mathcal{A} \subseteq \mathcal{F}$,
2. *If* $\varphi \in \mathcal{F}$, *then* $\neg\varphi \in \mathcal{F}$,
3. *If* $\varphi, \psi \in \mathcal{F}$ *and* $\sigma(\varphi) \cap \sigma(\psi) = \emptyset$, *then* $\varphi \vee \psi, \varphi \wedge \psi \in \mathcal{F}$.

[2] Submitted to SAFA 2020. `http://safa2020.argumentationcompetition.org/`

a	b	$a \vee b$	$a \wedge b$	$\neg a$
t	**t**	**t**	**t**	**f**
t	**f**	**t**	**f**	**f**
t	**u**	**t**	**u**	**f**
f	**t**	**t**	**f**	**t**
f	**f**	**f**	**f**	**t**
f	**u**	**u**	**f**	**t**
u	**t**	**t**	**u**	**u**
u	**f**	**u**	**f**	**u**
u	**u**	**u**	**u**	**u**

Table 3. Kleene's three-valued logic $\mathcal{K}_3$

It is easy to see that any formula $\varphi \in \mathcal{F}$ does not have multiple occurrences of atoms. The following theorem shows that if restricting acceptance functions to $\mathcal{F}$ the use of Kleenes strong three valued logic $\mathcal{K}_3$ [17] is enabled. The thruth tables regarding disjunction, conjunction and negation are given in Table 3.

Theorem 2. *For any $\varphi \in \mathcal{F}$ and any three-valued interpretation v we have:*

$$v^{\mathcal{K}_3}(\varphi) = \sqcap_i \{w(\varphi) \mid w \in [v]_2\}.$$

4. Discussion and Conclusion

The concept of time in regard to argumentation is not new. In [18] a timed argumentation framework (TAFs) is considered, which can be used for classical AFs and bipolar AFs [12]. In comparison tADFs are offering a more fine-grained approach, because not only pure attack and support relations between nodes can be considered but also mixed forms. In addition tADFs are offering the possibility to make statements about events which depends on other timesteps in the past or the near future. Therefore it is not required to consider a specific time-interval as in TAFs. An other approach to the time topic is the LARS-framework [19] which uses a logic-based framework and a window operator for modeling datatstreams at given time-intervals. Here the focus is on a continous stream of input and evaluation of possible actions. Timed ADFs are designed to consider all time points through the defined acceptance conditions. Therefore there is no narrowing to a current time step with information available at that moment, through this could be considered with specifc semantics. The definition of tADFs allows us to use all theoretical results about ADFs. In order to facilitate the calculation of ADFs semantics, we introduced a special subclass of formulas, where the value of the gammaoperator can be calculated directly with Kleenes strong-three valued logic. Also it could be shown that no three-valued logic in general can exist in order to model the gammaoperator. In further research we want to evaluate, whether there exist further subclasses of ADFs, which can be calculated with a pure logic approach. Also it appears feasible to look for specific time semantics, e.g. where the truth-value of an argument has the least changes over a given time period.

Acknowledgement

We thank DFG (Deutsche Forschungsgemeinschaft) and BMBF (Bundesministerium für Bildung und Forschung) for funding this work (project numbers 406289255 and 01IS18026B).

References

[1] Bench-Capon TJM, Dunne PE. Argumentation in Artificial Intelligence. Artificial Intelligence. 2007;171(10-15):619–641.

[2] Baroni P, Gabbay D, Giacomin M, van der Torre L. Handbook of Formal Argumentation. College Publications; 2018.

[3] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321–358.

[4] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. .

[5] Toni F. A tutorial on assumption-based argumentation. Argument & Computation. 2014;5(1):89–117.

[6] Modgil S, Prakken H. The $ASPIC^+$ framework for structured argumentation: a tutorial. Argument & Computation. 2014;5(1):31–62.

[7] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards Artificial Argumentation. AI Magazine. 2017;38(3):25–36.

[8] Amgoud L, Vesic S. A new approach for preference-based argumentation frameworks. Ann Math Artif Intell. 2011;63(2):149–183.

[9] Bench-Capon TJM, Atkinson K. Abstract Argumentation and Values. In: Argumentation in Artificial Intelligence; 2009. p. 45–64.

[10] Nielsen SH, Parsons S. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In: Workshop on Argumentation in Multi-Agent Systems; 2006. p. 54–73.

[11] Baroni P, Cerutti F, Giacomin M, Guida G. Encompassing Attacks to Attacks in Abstract Argumentation Frameworks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty; 2009. p. 83–94.

[12] Cayrol C, Lagasquie-Schiex M. Bipolar abstract argumentation systems. In: Argumentation in Artificial Intelligence; 2009. p. 65–84.

[13] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks Revisited. In: Rossi F, editor. Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013). IJCAI/AAAI; 2013. .

[14] Brewka G, Woltran S. Abstract Dialectical Frameworks. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010; 2010. .

[15] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks. An Overview. IfCoLog Journal of Logics and their Applications Volume 4, number 8 Formal Argumentation. 2017 10;4(8):2263–2317.

[16] Denecker M, Marek VW, Truszczyński M. Ultimate approximation and its application in nonmonotonic knowledge representation systems. Inf Comput. 2004;192(1):84–121.

[17] Wintein S. On All Strong Kleene Generalizations of Classical Logic. Studia Logica. 2016;104(3):503–545.

[18] Budán MCD, Cobo ML, Martinez DC, Simari GR. Bipolarity in temporal argumentation frameworks. International Journal of Approximate Reasoning. 2017;84:1–22.

[19] Beck H, Dao-Tran M, Eiter T, Fink M. LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI'15. AAAI Press; 2015. p. 1431–1438.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200497

Ranking-Based Semantics from the Perspective of Claims

Stefano BISTARELLI [a], Wolfgang DVOŘÁK [b], Carlo TATICCHI [c], and Stefan WOLTRAN [b]

[a] *Department of Mathematics and Computer Science, Univ. of Perugia, Italy*
[b] *Institute of Logic and Computation, TU Wien, Austria*
[c] *Gran Sasso Science Institute, L'Aquila, Italy*

Abstract. The paper provides an initial study on how ranking semantics in argumentation have to be handled when leaving the purely abstract setting. We employ claim-augmented frameworks where each argument is associated to a claim it stands for. We propose liftings from argument- to claim-level in two veins: for desired properties and for actual rankings. Our main contribution is to investigate whether the satisfaction of properties by argument-based ranking semantics carries over to the lifted, claim-based, variants of the corresponding properties and semantics.

Keywords. Ranking semantics, Claim-augmented frameworks.

1. Introduction

In abstract argumentation, the concept of ranking semantics has received increasing attention over the last years. In contrast to the traditional extension- and labelling-based semantics which assign only a few different levels of acceptance to arguments, ranking semantics order arguments from the most to the least acceptable ones. Several such semantics have been introduced and numerous properties that characterise the rankings have been proposed; see e.g. [1,2] for an overview.

However, there is common agreement [3] that abstract argumentation should not be treated as an isolated formalism but be embedded in an instantiation procedure which generates the arguments and the relation between them. Arguments in the abstract setting should be seen as placeholders for a more complex structure which at least contain a claim the argument stands for. The effect for traditional abstract argumentation semantics in this context has been thoroughly investigated [4,5]. Prominently, studies concerning rationality postulates [6] revealed certain undesired effects. To the best of our knowledge, a systematic analysis of ranking-based semantics in this context has not been undertaken yet.

In this work, we provide an initial study towards an understanding of the functioning of ranking semantics when arguments are not considered to be purely abstract but where each argument stands for a particular claim. In such a setting, standard ranking semantics over arguments implicitly provide an order over claims; however, given the common situation that different arguments can stand

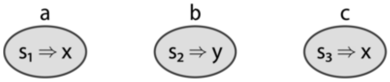


Figure 1. Arguments a, b and c with claims (x, y) and supports to the claims (s_1, s_2, s_3).

for the same claim, it is evident that certain ambiguities arise: consider a framework with three arguments a, b, c, where arguments a and c stand for claim x and argument b stands for claim y (see Figure 1), and a ranking of arguments of the form $a \succ b \succ c$ is determined by some ranking semantics (note that such a ranking is not implausible: it might be the case that the support of x in argument a is more plausible than the support of x in argument c). What does the ranking $a \succ b \succ c$ of arguments then tell us when we are interested in a ranking of their claims? Is claim x more acceptable than claim y?

The main objectives in this paper are to propose and investigate a "lifting" of an argument-ranking to a claim-ranking. The idea of lifting relies on the intuition that a claim x is more acceptable than a claim y if there is at least one argument for claim x that is more acceptable than all arguments for claim y. In order to investigate the behaviour of such a lifting, we will reformulate several properties (originally proposed to classify argument-based ranking semantics) in a claim-centric perspective. The main insights of our paper are to show that statements in the spirit of "for every argument-ranking semantics that satisfies property P, its lifted version satisfies the claim-centric variant of P" hold or are violated. As a vehicle for these investigations we use claim-augmented argumentation frameworks (CAFs) as introduced in the complexity-study [4]. CAFs provide a natural intermediate layer between structured and purely abstract argumentation, as they carry the necessary information to first compute the extensions and then re-interpret them in terms of the instantiated problem.

Indeed, there would be different ways to come up with a ranking on the claim-level. One would be to avoid the situation where different arguments are related to the same claim (since then, the ranking of claims is immediate from a ranking of arguments). Recently, translations towards such unique-claim frameworks have been investigated [5] and could be coupled with ranking semantics for frameworks with collective attacks [7]. Another option would be to define new ranking semantics on claims from scratch, for instance, by taking the logical structure of claims also into account. In this preliminary study, we have opted for the lifting approach outlined above, since (a) it naturally builds on ranking semantics on the argument level which are well understood and (b) it provides first immediate insights on the relationship between rankings on argument- and claim-level which might be useful towards more special-tailored claim-ranking semantics.

2. Background

We recall the fundamental definitions from [8,4] and provide the notion of ranking-based semantics and some of their logical properties proposed in the literature.

Definition 1. An *abstract argumentation framework* (AF) is a pair (A, R) where A is a set of arguments and R is a binary attack relation on A. Given two arguments

$a, b \in A$, an attack from a to b is denoted with $(a, b) \in R$. Given an AF F, we use A_F to refer to the arguments of F and R_F to refer to the attack relation of F. For AF $F = (A, R)$, $a \in A$ and $S \subseteq A$, we define $a_F^+ = \{b \in A \mid (a, b) \in R\}$, $a_F^- = \{b \in A \mid (b, a) \in R\}$, $S_F^+ = \bigcup_{a \in S} a_F^+$ and $S_F^- = \bigcup_{a \in S} a_F^-$ (we will omit the subscript F when it is clear from the context).

While standard semantics for AFs [8] are defined on top of the notion of acceptability (an argument a is acceptable with respect to a set S in an AF F, if $a^- \subseteq S^+$) and select subsets of arguments, *ranking-based* semantics [1] can be used for sorting the arguments from the most to the least preferred.

Definition 2. A *ranking semantics* associates with any AF $F = (A, R)$ a total pre-order (i.e., a reflexive transitive relation) $\succcurlyeq_F$ on A, called the ranking on F. $a \succcurlyeq_F b$ means that a is at least as acceptable as b in F; $a \simeq_F b$ is a shortcut for $a \succcurlyeq_F b$ and $b \succcurlyeq_F a$; $a \succ_F b$ is a shortcut for $a \succcurlyeq_F b$ and $b \not\succcurlyeq_F a$. Again, when clear from the context, we will just write $\succcurlyeq$ to denote the ranking.

Ranking-based semantics can be characterised through the set of logical properties they satisfy, among those proposed in the literature. For this work, we refer to the properties introduced in [9]. We recall them in Table 1 with the abbreviations standing for abstraction, independence, void precedence, self-contradiction, cardinality precedence, quality precedence, counter-transitivity, strict counter-transitivity, and defence precedence. A few auxiliary concepts are required. First, an isomorphism between two AFs $F = (A, R)$ and $F' = (A', R')$ is given by a bijective function $\gamma : A \to A'$ such that $\forall a, b \in A$, $(a, b) \in R$ if and only if $(\gamma(a), \gamma(b)) \in R'$. We use $\gamma(F)$ to identify the AF F' obtained by applying γ to the AF F and call F and F' to be γ-isomorph. Second, a weakly connected component of an AF F is a maximal subgraph of F in which any two nodes are connected to each other by a path (ignoring the direction of the edges); $cc(F)$ represents the set of weakly connected components of F.

Finally, we require the concept of group comparison [9]. Let $\succcurlyeq$ be a ranking on the elements of a set S. The associated group comparison $\succcurlyeq^G$ on 2^S is defined as follows, for $S_1, S_2 \subseteq S$: $S_1 \succcurlyeq^G S_2$ if and only if there exists an injective mapping $f : S_2 \to S_1$ such that $\forall a \in S_2$, $f(a) \succcurlyeq a$. Moreover, $S_1 \succ^G S_2$ if and only if $S_1 \succcurlyeq^G S_2$ and either $|S_2| < |S_1|$ or f additionally satisfies $f(a) \succ a$ for some $a \in S_2$. For an AF $F = (A, R)$ and ranking $\preceq$ on A, $\preceq^G$ denotes the associated group comparison over subsets of A, as used, for instance, in (**CT**) and (**SCT**).

Table 1. Properties for argument-based ranking semantics.

(**Abs**)	$\forall$ γ-isomorph AFs F, F', $a, b \in A_F$: $a \succcurlyeq_F b \iff \gamma(a) \succcurlyeq_{F'} \gamma(b)$
(**Ind**)	$\forall$ AFs F, $F' \in cc(F)$, $a, b \in A_{F'}$: $a \succcurlyeq_F b \iff a \succcurlyeq_{F'} b$
(**VP**)	$\forall$ AFs F, $a, b \in A_F$: $\left(a^- = \emptyset \wedge b^- \neq \emptyset\right) \implies a \succ b$
(**SC**)	$\forall$ AFs F, $a, b \in A_F$: $\left(a \notin a^+ \wedge b \in b^+\right) \implies a \succ b$
(**CP**)	$\forall$ AFs F, $a, b \in A_F$: $\lvert a^- \rvert < \lvert b^- \rvert \implies a \succ b$
(**QP**)	$\forall$ AFs F, $a, b \in A_F$: $\left(\exists c \in b^- : \forall d \in a^- : c \succ d\right) \implies a \succ b$
(**CT**)	$\forall$ AFs F, $a, b \in A_F$: $b^- \succcurlyeq^G a^- \implies a \succcurlyeq b$
(**SCT**)	$\forall$ AFs F, $a, b \in A_F$: $b^- \succ^G a^- \implies a \succ b$
(**DP**)	$\forall$ AFs F, $a, b \in A_F$: $\left(\lvert a^- \rvert = \lvert b^- \rvert, (a^-)^- \neq \emptyset = (b^-)^-\right) \implies a \succ b$

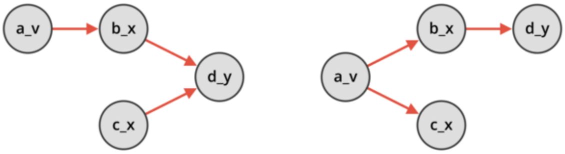


Figure 2. Examples of a well-formed CAF (left) and an att-unitary CAF (right).

We conclude the this section with the formalism we will base our studies on. The main idea is to provide additional information by associating a claim to each argument, while a claim can be associated with more than one argument.

Definition 3. A *claim-augmented argumentation framework* (CAF) is a triple (A, R, claim) where (A, R) is an AF and $\mathit{claim}\colon A \to X$ assigns a claim (from a given universe of possible claims X) to each argument of A. If $\mathit{claim}(a) = x$ we also say that a supports x. For a CAF $CF = (A, R, \mathit{claim})$, we use AF_{CF} to refer to AF (A, R) and X_{CF} to denote the set of claims in CF, i.e. $X_{CF} = \{\mathit{claim}(a) \mid a \in A\}$.

Given a CAF $CF = (A, R, \mathit{claim})$, and claim $x \in X$, we use $A_{CF,x}$ to denote the set of arguments a in CF with $\mathit{claim}(a) = x$. For claims $x, y \in X$, we say that x attacks y in CF if there are arguments $a \in A_{CF,x}, b \in A_{CF,y}$, such that $(a, b) \in R$; we further use $x^+_{CF} = \{y \in X \mid x \text{ attacks } y \text{ in } CF\}$ and $x^-_{CF} = \{y \in X \mid y \text{ attacks } x \text{ in } CF\}$. If clear from the context, we will drop the subscript CF.

Extension-based semantics for CAFs are defined by re-interpreting extensions of the standard semantics of the underlying AF via the *claim*-function. Since we focus here on ranking semantics, we refer to [4] for details. We finally introduce two central subclasses of CAFs [5].

Definition 4. Let $CF = (A, R, \mathit{claim})$ be a CAF with $F = (A, R)$ the underlying AF. CF is called (i) *well-formed* if $a^+ = b^+$ for all $a, b \in A$ with $\mathit{claim}(a) = \mathit{claim}(b)$; (ii) *att-unitary* if $a^- = b^-$ for all $a, b \in A$ with $\mathit{claim}(a) = \mathit{claim}(b)$.

In other words, a CAF is well-formed when arguments with the same claim attack the same arguments, while it is att-unitary when arguments with the same claim are attacked by the same arguments. Examples are given in Figure 2 (we use the label a_x to denote an argument a supporting a claim x).

3. CAFs Ranking and Properties

Our goal is to transfer the notion of ranking from classical AFs to CAFs. We call this mapping a *lifting* from arguments to claims. In particular, we are interested in checking whether the properties satisfied by a ranking-based semantics on AFs are preserved (on the level of claims) after the lifting. To conduct this study, we need the concept of ranking-based semantics for CAFs in the first place and then discuss how properties are lifted to the claim level.

Definition 5. A *claim-based ranking semantics* associates with any CAF CF a total pre-order $\succcurlyeq_{CF}$ on X_{CF}, called the ranking on CF. $x \succcurlyeq_{CF} y$ means that claim x is at least as acceptable as claim y in CF. Shortcuts $x \simeq_{CF} y$ and $x \succ_{CF} y$ are analogous to Definition 2. Again, we will occasionally drop the subscript.

Table 2. Basic properties for lifted claim-based ranking semantics.

Support Dependency (**SD**)	$\forall$ CAFs CF, $x, y \in X_{CF}$: $A_x \succcurlyeq^G A_y \implies x \succcurlyeq y$
Strict SD (**SSD**)	$\forall$ CAFs CF, $x, y \in X_{CF}$: $A_x \succ^G A_y \implies x \succ y$
Generalized SD (**GSD**)	$\forall$ CAFs CF, $x \in X_{CF}$: $x \succcurlyeq_{CF} y \implies x \succ_{CF^{+x}} y$

Table 3. Properties for claim-based ranking semantics.

(**C-Abs**)	$\forall$ γ_X-isomorph CAFs CF, CF', $x, y \in X_{CF}$: $x \succcurlyeq_{CF} y \iff \gamma_X(x) \succcurlyeq_{CF'} \gamma_X(y)$
(**C-Ind**)	$\forall$ CAFs CF, $CF' \in cc_c(CF)$, $x, y \in X_{CF'}$: $x \succcurlyeq_{CF} y \iff x \succcurlyeq_{CF'} y$
(**C-VP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(x^- = \emptyset \wedge y^- \neq \emptyset) \implies x \succ y$
(**C-SC**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(x \notin x^+ \wedge y \in y^+) \implies x \succ y$
(**C-CP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\lvert x^- \rvert < \lvert y^- \rvert \implies x \succ y$
(**C-QP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(\exists z \in y^- : \forall u \in x^- : z \succ u) \implies x \succ y$
(**C-CT**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $y^- \succcurlyeq^G x^- \implies x \succcurlyeq y$
(**C-SCT**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $y^- \succ^G x^- \implies x \succ y$
(**C-DP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(\lvert x^- \rvert = \lvert y^- \rvert, (x^-)^- \neq \emptyset = (y^-)^-) \implies x \succ y$

In what follows, we define three sets of properties for claim-based ranking semantics to be satisfied. The first group (see Table 2) of such properties are concerned with the fundamental properties one expects from a lifting from the ranking of arguments of an AF to a ranking of claims of a CAF. First we require that if the support sets of two claims are comparable (via group comparison), we want that claims with (strictly) stronger support to be (strictly) stronger, see properties **SD** and **SSD**. Second we require that the ranking of a claim is strengthened by additional support. To this end, we define for a CAF $CF = (A, R, \mathit{claim})$ and claim $x \in X_{CF}$, the CAF $CF^{+x} = (A, R, \mathit{claim}')$ where some argument $a \in A$ with $\mathit{claim}(a) \neq x$ gets x as its claim, i.e. $\mathit{claim}'(a) = x$ and $\mathit{claim}'(b) = \mathit{claim}(b)$ for $b \in A \setminus \{a\}$, see property **GSD**. Notice, that (A, R) is unchanged in CF^{+x} and thus the ranking of arguments is not affected.

The second group basically rephrases the properties from Table 1 in such a way that the notion of attack on argument-level is replaced by the notion of attack on the claim-level (cf. Definition 3).[1] The resulting properties are called claim-oriented and collected in Table 3. We also need adaption of γ-isomorphism and weakly connected components. First, an isomorphism between claims is a bijective function $\gamma_X : X \to X$. Given CAF $CF = (A, R, \mathit{claim})$, we also use $\gamma_X(CF)$ to denote the CAF (A, R, claim') where $\mathit{claim}'(a) = \gamma_X(\mathit{claim}(a))$ for all $a \in A$, and call CF and CF' to be γ_X-isomorph. Second, the notion of weakly connected components is extended to CAFs in the following way: The claim-connected components $cc_c(CF)$ of a CAF $CF = (A, R, \mathit{claim})$ are the subset maximal sub-frameworks such that the involved claims are weakly connected via attacks between claims (note that each claim-connected component is thus the union of one or more connected components of (A, R)). Finally, group comparison is used in the same way as in Section 2: For a CAF CF and a ranking $\preceq$ on X_{CF}, $\preceq^G$ denotes the associated group comparison over subsets of X_{CF}.

[1] These properties do not address the different natures of arguments and claims and thus not all of them are expected properties of claim-rankings. However, they are perfectly suited to study which properties are maintained by lifting argument rankings to the claim level.

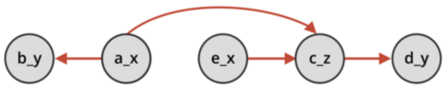


Figure 3. Example of a CAF CF where we have $(A_y)^- = \{a, c\}$ and $A_{y^-} = \{a, c, e\}$.

Table 4. Refined properties for claim-based ranking semantics.

(**AC-Abs**)	$\forall$ γ-isomorph CAFs CF, CF', $x, y \in X_{CF}$: $x \succcurlyeq_{CF} y \iff \gamma_X(x) \succcurlyeq_{CF'} \gamma_X(y)$
(**AC-Ind**)	$\forall$ CAFs CF, $CF' \in cc^*(F)$, $x, y \in X_{CF'}$: $x \succcurlyeq_{CF} y \iff x \succcurlyeq_{CF'} y$
(**AC-VP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\big(\exists a \in A_x : a^- = \emptyset \wedge \forall b \in A_y : b^- \neq \emptyset\big) \implies x \succ y$
(**AC-SC**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\big(\exists a \in A_x : a \notin a^+ \wedge \forall b \in A_y : b \in b^+\big) \implies x \succ y$
(**AC-CP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\lvert(A_x)^-\rvert < \lvert(A_y)^-\rvert \implies x \succ y$
(**AC-QP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\big(\exists a \in (A_y)^- : \forall b \in (A_x)^- : a \succ b\big) \implies x \succ y$
(**AC-CT**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(A_y)^- \succcurlyeq^G (A_x)^- \implies x \succcurlyeq y$
(**AC-SCT**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $(A_y)^- \succ^G (A_x)^- \implies x \succ y$
(**AC-DP**)	$\forall$ CAF CF, $x, y \in X_{CF}$: $\big(\lvert(A_x)^-\rvert = \lvert(A_y)^-\rvert, ((A_x)^-)^- \neq \emptyset \wedge$ $((A_y)^-)^- = \emptyset\big) \implies x \succ y$

The final group of properties provides an alternative to ones in Table 3. The intuition is that the simple replacement of attack between arguments by attack between claims might be too general. Take for instance, property **C-VP**: it applies only when each supporter of claim x in CAF has no attacker ($x^- = \emptyset$) and claim y has at least one supporter being attacked ($y^- \neq \emptyset$). However, it also appears reasonable to apply this property when at least one supporter of x has no attacker, but all supporters of y are attacked. The resulting property is **AC-VP**. Likewise, we occasionally replace x^- (i.e. the set of claims attacking x) by $(A_x)^-$ (i.e. the set of arguments attacking the supporters of x). It is important to note that $(A_x)^-$ is different to the supporters of x^-, i.e. the set $A_{x^-} := \bigcup_{y \in x^-} A_y$; see Figure 3.

Table 4 presents those refinements where the ranking of claims is obtained from the arguments supporting the claim, and for certain properties (i.e., **AC-QP**, **AC-CT**, and **AC-SCT**) we even take into account that the claim ranking is obtained by lifting an argument ranking. Two things remain to be clarified: For **AC-Abs**, given CAF $CF = (A, R, claim)$, we use a pair of bijective functions $\gamma = (\gamma_A, \gamma_X)$ with $\gamma_A : A \to A$ and $\gamma_X : X \to X$. We also use $\gamma(CF)$ to denote the CAF $(A', R', claim')$ where $A' = \{\gamma_A(a) \mid a \in A\}$ $R' = \{(\gamma_A(a), \gamma_A(b)) \mid (a, b) \in R\}$ and $claim'(\gamma_A(a)) = \gamma_X(claim(a))$ for all $a \in A$, and call CF and CF' to be γ-isomorph. Second, for **AC-Ind**, the CAFs in $cc^*(CF)$ are obtained by the weakly connected components of AF $F = AF_{CF}$ together with the claim function from CF restricted to the arguments in that component.

4. Lifting via Lexicographic Order

In this section, we first propose a method for lifting a ranking on arguments to a ranking on claims based on a certain lexicographic order, and then investigate how this claim-based ranking relates to the properties introduced in the previous section. Using a lexicographic order is based on the following intuition: if a claim

x has one supporter that is better than all the supporters of another claim y, then $x \succ y$. In case the best supporters of x and y are equally acceptable, we look at the second best supporter and so on. Formally, this is captured as follows.

Definition 6. Given a set S ordered through a preference relation $\succcurlyeq$, let $\max(S)$ return an element $s \in S$ such that $\nexists t \in S, t \succ s$.[2] We define the *lexicographic order* relation $\succcurlyeq^L$ (based on $\succcurlyeq$) between subsets of S as follows ($A, B \subseteq S$):

- $A \succcurlyeq^L \emptyset$, $\emptyset \not\succcurlyeq^L A$, for $A \neq \emptyset$
- $A \succcurlyeq^L B \iff i)$ $\max(A) \succ \max(B)$, or
  $ii)$ $\max(A) \succcurlyeq \max(B)$ and $A \setminus \max(A) \succcurlyeq^L B \setminus \max(B)$

As before, we write $A \succ^L B$ in case $A \succcurlyeq^L B$ and $B \not\succcurlyeq^L A$ jointly hold.

As we will show, $\succcurlyeq^L$ is always a refinement of the group-comparison $\succcurlyeq^S$ as defined in Section 2. However, the concepts are clearly different: Let $S = \{a, b, c\}$ and $a \succ b \succ c$. For the sets $A = \{a\}$ and $B = \{b, c\}$ we have that $A \succ^L B$ as $\max(A) = a \succ \max(B) = b$, but even $A \succcurlyeq^G B$ cannot hold since $|B| > |A|$. Note that also $B \succcurlyeq^G A$ cannot hold since each element in B is worse than a in $\succcurlyeq$. In fact, when $\succcurlyeq$ is a total (pre-)order then also $\succcurlyeq^L$ is a total (pre-)order.

We now define our central notion of lifting a ranking semantics.

Definition 7. Given a ranking semantics σ that assigns to any AF F a ranking $\succcurlyeq_F$ on A_F, we call a claim-based ranking semantics σ' a *lex-lifting* of σ if for each CAF CF the ranking $\succcurlyeq_{CF}$ assigned by σ' satisfies:

$$\textbf{LO} : \text{for all claims } x, y \in X_{CF} \ : x \succcurlyeq_{CF} y \iff A_x \succcurlyeq^L_{AF_{CF}} A_y$$

As argument rankings only provide the order of arguments according to their strength but no quantitative measure on the difference of their acceptance degrees there are strong limitations on how the strengths of the arguments supporting a claim and the number of arguments supporting a claim can be traded against each other when computing the claim ranking. We consider the strength to be more important, e.g., a claim supported by an unattacked argument should be ranked higher than an argument solely supported by (a large number of) self-attacking arguments. **LO** implements this intuition by first going for the strongest supporting arguments and the number of arguments only becomes relevant when the strongest supporting arguments tie. Alternatives to the lex-lifting would be to order claims by considering the minimum, maximum, or median strength argument supporting the claim. We expect that these approaches (when compared to the lex-lifting approach) will satisfy a smaller number of the analysed properties; a detailed comparison is subject of future work.

We now show which properties are satisfied for lex-liftings, in particular under the assumption that the underlying ranking semantics satisfies the corresponding property on the argument level. As we will see, satisfaction is sometimes conditioned by subclasses of CAFs (cf. Def 4). We start with properties from Table 2.

Proposition 1. *Every lex-lifting of a ranking semantics satisfies* **SD***,* **SSD***, and* **GSD***.*

[2] If there are several such elements $s \in S$ then $\max(S)$ picks an arbitrary of these elements.

Proof. For **SD**, it suffices to show that for every ranking $\succcurlyeq$ on some set S, it holds that $\succcurlyeq^G \subseteq \succcurlyeq^L$. Hence suppose $A \succcurlyeq^G B$, for $A, B \subseteq S$. By definition, there is an injective mapping $f : B \to A$ such that $\forall b \in B$, $f(b) \succcurlyeq b$. W.l.o.g. we can assume that f is monotone (since, if $f(x) \succ f(y)$ for some $y \succ x$, we can swap the values of $f(x)$ and $f(y)$), and that $f(\max(B)) = \max(A)$ by the same argument. Thus $\max(A) \succcurlyeq \max(B)$. In case $\max(A) \succ \max(B)$ we obtain $A \succ^L B$; otherwise, let $A' = A \setminus \max(A)$, $B' = B \setminus \max(B)$ and consider $f' : B' \to A'$ with $f'(x) = f(x)$ for all $x \in B'$. Since f is injective, f' is injective too, and by definition $\forall b \in B'$, $f(b) \succcurlyeq b$. We thus can continue this argument until the recursion comes to a halt or $B' = \emptyset$. Finally, the other two properties can be proven in a similar way. □

We now continue with the properties from Table 3 and their refined versions from Table 4. In each proposition, we will oppose a property and its refined version. In case a property does not hold (or only for some subclass of CAFs), corresponding counterexamples are given right after the proposition.

Proposition 2. *For every lex-lifting σ' of a ranking semantics σ it holds that: (1) σ' satisfies* **C-Abs***; and (2) σ' satisfies* **AC-Abs***, whenever σ satisfies* **Abs***.*

Proof. 1) holds since no matter how γ_X is chosen, we have $A_x = A_{\gamma_X(x)}$, and thus the property **LO** is not affected. 2) Consider a CAF $CF = (A, R, \mathit{claim})$, let $\succcurlyeq_{CF}$ be the ranking assigned by σ, $\gamma = (\gamma_A, \gamma_X)$ be a pair of bijective functions $\gamma_A : A \to A$ and $\gamma_X : X_{CF} \to X_{CF}$, and $CF' = (A', R', \mathit{claim}')$ be a CAF γ-isomorphic to CF. Let $x, y \in X_{CF}$ and suppose $x \succcurlyeq_{CF} y$. By **LO**, we know that $A_x \succcurlyeq^L_{(A,R)} A_y$. Now as σ satisfies **Abs**, we also get $\gamma_A(A_x) \succcurlyeq^L_{(A',R')} \gamma_A(A_y)$, where $\gamma_A(S) = \{\gamma_A(a) \mid a \in S\}$. Further as γ is an isomorphism between CF and CF' we have $\mathit{claim}'(\gamma_A(a)) = \gamma_X(\mathit{claim}(a))$ for all $a \in A$. It follows that $\gamma_A(A_x) = A_{CF',\gamma_X(x)}$ and $\gamma_A(A_y) = A_{CF',\gamma_X(y)}$. Thus $\gamma_X(x) \succcurlyeq^L_{AF_{CF'}} \gamma_X(y)$, and by the **LO** property of σ' we arrive at $\gamma_X(x) \succcurlyeq_{CF'} \gamma_X(y)$. The reverse direction can be shown in essentially the same way and we obtain that σ' satisfies **AC-Abs**. □

Obviously, (2) cannot be satisfied in general. Just consider $\gamma = (\gamma_A, \gamma_X)$ with γ_X the identity function. If σ does not satisfy **Abs**, σ' cannot satisfy **AC-Abs** then, since **Abs** and **AC-Abs** coincide in this setting.

Proposition 3. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **Ind** *it holds that: (1) σ' satisfies* **C-Ind***; and (2) σ' satisfies* **AC-Ind** *for CAFs being well-formed or att-unitary.*

Proof. 1) basically holds since the supporters of a claim in a CAF CF are guaranteed to occur in exactly one component $CF' \in cc_c(CF)$. From that **C-Ind** for σ' carries over from **Ind** for σ. 2) Since CF is well-formed or att-unitary, it is guaranteed that all supporters of a claim are contained in the same weakly connected component of AF_{CF} or all supporters of that claim are isolated (no incoming or outgoing attack, no self-attacks). In the former case, the argument of 1) applies; for the latter, **AC-Ind** does not pose any restriction on ordering those claims. □

If the CAF is neither well-formed nor att-unitary, we have no guarantee that **AC-Ind** is satisfied when lifting a semantics satisfying **Ind**. A counterexample

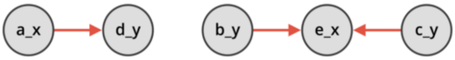


Figure 4. A CAF with two weakly connected components where lifting violates **AC-Ind**.

is provided in Figure 4. Note that this CAF CF is neither well-formed (since, for instance, e does not attack d) nor att-unitary (since b is not attacked by a). Suppose to have the following ranking over the arguments: $a \simeq b \simeq c \succ d \succ e$, and let CF' the sub-CAF on the left-hand side (with arguments a, d). By **LO**, we have $x \succ_{CF'} y$ (observe that $a \succ d$ carries over to that sub-AF due to **Ind**) but $y \succ_{CF} x$ (since $max(A_x) = a \simeq b = max(A_y)$, following **LO**, it remains to compare d with $\{c, e\}$, where c is preferred over d).

Proposition 4. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **VP** *it holds that: (1) σ' satisfies* **C-VP** *for att-unitary CAFs; (2) σ' satisfies* **AC-VP**.

Proof. We start with 2). Consider a CAF CF and two claims $x, y \in X_{CF}$, such that $\exists a \in A_x : a^- = \emptyset$ and $\forall b \in A_y : b^- \neq \emptyset$. We have to show that $x \succ y$. Since σ satisfies **VP** we know that $a \succ b$ for all $b \in A_y$. Hence, $\max(A_x) \succ \max(A_y)$ and thus $A_x \succ^L A_y$. By **LO**, $x \succ y$. Continuing with 1), for an att-unitary CAF CF, we know that for any claim y, with $y^- \neq \emptyset$, $b^- \neq \emptyset$ for all $b \in A_y$. We thus can apply the same reasoning as in 2). □

Now consider the well-formed CAF CF with arguments a, b, c, a attacking b and a supports claim x while b, c both support claim y. Assume further a ranking semantics σ that assigns $a \simeq c \succ b$ and thus satisfying **VP**. Via **LO** this is lifted to $y \succ x$ and, by definition, $A_y \setminus \{b\} = \{c\}$ while $A_x \setminus \{a\} = \emptyset$, yields $A_y \succcurlyeq^L A_x$. On the other hand, $x^- = \emptyset$ and $y^- \neq \emptyset$, i.e., **C-VP** is violated.

Proposition 5. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **SC** *it holds that (1) σ' satisfies* **C-SC** *for CAFs that are well-formed and att-unitary; and (2) σ' satisfies* **AC-SC**.

Proof. 2) Consider a CAF CF and two claims $x, y \in X_{CF}$, such that $\exists a \in A_x : a \notin a^+$ and $\forall b \in A_y : b \in b^+$. We have to show that $x \succ y$. Since σ satisfies **SC** we know that $a \succ b$ for all $b \in A_y$. Hence, $\max(A_x) \succ \max(A_y)$ and thus $A_x \succ^L A_y$; by **LO**, $x \succ y$. 1) it is sufficient to see that for CAFs that are both well-formed and att-unitary, an argument a with claim x is self-attacking iff all arguments with claim x are self-attacking. The argument from 2) then applies here as well. □

We show that for CAFs that are either well-formed or att-unitary (but not both), we have no guarantee that **C-SC** is satisfied when lifting a semantics satisfying **SC**. Consider the CAF CF with arguments a, b, c, such that b attacks b. For the case of well-formed CAFs, consider additional attack (c, b); for att-unitary CAFs, consider instead (b, c). Moreover, a supports claim x and b, c both support claim y. Assume further that a ranking semantics σ assigns $a \simeq c \succ b$ to the AF thus satisfying **SC**. Via **LO** this is lifted to $y \succ x$ (since $\max(A_x) = a \simeq b = \max(A_y)$ and, by definition, $A_y \setminus \{b\} = \{c\}$ while $A_x \setminus \{a\} = \emptyset$, yields $A_y \succcurlyeq^L A_x$). On the other hand, $x \notin x^+$ but $y \in y^+$. Hence, **C-SC** is violated.

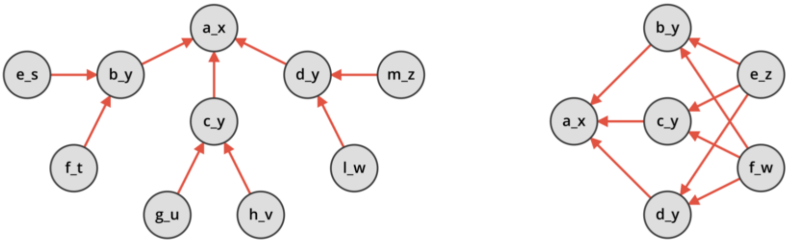


Figure 5. Examples of a well-formed CAF where lifting violates **AC-CP** (left) and a well-formed and att-unitary CAF where lifting violates **C-CP** (right).

Proposition 6. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **CP** *it holds that σ' satisfies* **AC-CP** *for att-unitary CAFs.*

Proof. Consider an att-unitary CAF CF and claims $x, y \in X_{CF}$, such that $|(A_x)^-| < |(A_y)^-|$. We have to show that $x \succ y$. Note that each $a \in A_x$ has the same attackers, and the same is true for A_y. Hence, for each $a \in A_x$ and each $b \in A_y$, $|a^-| < |b^-|$. Since σ satisfies **CP** we know that $a \succ b$ for all $a \in A_x, b \in A_y$. Hence, $\max(A_x) \succ \max(A_y)$ and thus $A_x \succ^L A_y$, and by **LO**, $x \succ y$. □

We next show that **AC-CP** is not guaranteed for well-formed CAFs and **C-CP** is not guaranteed for CAFs that are both well-formed and att-unitary. First, consider the first CAF CF in Figure 5 and the ranking $b \simeq c \simeq d \succ a$ satisfying **CP**. Lifting this ranking yields $y \succ x$. On the other hand, we have $|(A_x)^-| = 3$ and $|(A_y)^-| = 6$. Thus, **AC-CP** requires $x \succ y$ and is thus violated. Second, consider the second CAF in Figure 5 which is both well-formed and att-unitary. Consider the ranking $e \simeq f \succ b \simeq c \simeq d \succ a$. This ranking satisfies **CP** on the underlying AF and **LO** yields $y \succ x$. However, on the level of claims we have $|x^-| = 1$ (the only claim attacking x is y) and $|y^-| = 2$. Thus, **C-CP** requires $x \succ y$.

Proposition 7. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **QP** *it holds that σ' satisfies* **AC-QP** *for att-unitary CAFs.*

We skip the proof of this result and the forthcoming Propositions 8 and 9 for space reasons. They follow the same pattern as the proofs of the previous results.

We show that lifting to **AC-QP** does not work for well-formed CAFs. Consider the first CAF CF depicted in Figure 6 and the ranking $a \simeq e \succ d \succ c \succ b$ that satisfies **QP**. Moreover, with $a \in (A_y)^-$ we have an argument such that for all $b \in (A_x)^-$, $a \succ b$. Hence, **AC-QP** would require $x \succ y$. However, the lifting of the ranking via **LO** yields $y \succ x$ (since $e \succ d$).

Lifting to **C-QP** does not hold, even for CAFs that are both well-formed and att-unitary. Consider the CAF of Figure 7 (right-hand side) and the ranking $a \simeq b \simeq d \succ f \succ c \simeq e$ that satisfies **QP**. Its lifting yields $z \succ u \succ y \succ x \simeq v$ (note that $|A_z| > |A_u|$, thus $z \succ u$). Now we have $z \in x^-$, $v^- = \{u\}$ and $z \succ u$ but $v \not\succ x$ and thus **C-QP** is violated.

Proposition 8. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **CT**, **SCT** *resp., it holds that σ' satisfies* **AC-CT**, **AC-SCT** *resp., for att-unitary CAFs.*

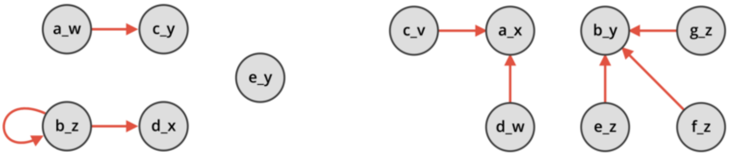


Figure 6. Examples of a well-formed CAF where lifting violates **AC-QP**, **AC-SCT** and **AC-CT** (left) and a well-formed, att-unitary CAF where lifting violates **C-CT** and **C-SCT** (right).

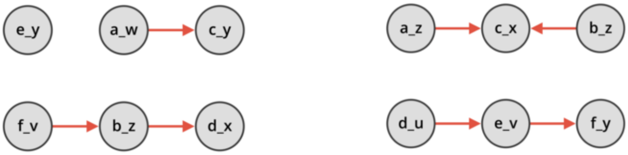


Figure 7. Examples of a well-formed CAF where lifting violates **AC-DP** (left) and a well-formed, att-unitary CAF where lifting violates **C-DP** and **C-QP** (right).

To show that lifting to **AC-SCT** does not hold for well-formed CAFs, we reuse the first CAF CF from Figure 6 and the ranking $a \simeq e \succ d \succ c \succ b$ satisfying **SCT** ($a \succ b \implies d \succ c$). Since, $(A_y)^- = \{a\}$ and $(A_x)^- = \{b\}$, **AC-SCT** requires $x \succ y$. However, $y \succ x$ via **LO**. Note that the example applies also to **AC-CT**.

It remains to illustrate the problems with lifting to **C-CT**. Consider the second CAF CF depicted in Figure 6 which is well-formed and att-unitary and a ranking of arguments $c \simeq d \simeq e \simeq f \simeq g \succ a \succ b$ (satisfying **SCT** and **CT**). Its lifting yields, in particular, $x \succ y$. However, $x^- = \{v, w\}$ and $y^- = \{z\}$ and thus $x^- \succ^G y^-$ which requires $y \succ x$ (to satisfy **C-SCT**) or $y \succcurlyeq x$ (to satisfy **C-CT**).

Proposition 9. *For every lex-lifting σ' of a ranking semantics σ that satisfies* **DP** *it holds that σ' satisfies* **AC-DP** *for att-unitary CAFs.*

Lifting to **AC-DP** does not hold for well-formed CAFs: let CF be the CAF given in Figure 7 (left-hand side) with ranking $a \simeq e \simeq f \succ d \succ b \simeq c$. This ranking satisfies **DP** and its lifting implies, in particular, $y \succ x$ since e is preferred over d. On the other hand, $|(A_x)^-| = |(A_y)^-| = 1$ and $((A_x)^-)^- = \{f\}$ while $((A_y)^-)^- = \emptyset$. **AC-DP** (which would thus require $x \succ y$) is therefore violated.

We finally show that lifting to **C-DP** does not hold, even for CAFs that are both well-formed and att-unitary. Consider the CAF of Figure 7 (right-hand side) and ranking $a \simeq b \simeq c \simeq d \succ f \succ e$ on AF that satisfies **DP**. In particular, we have that $z \simeq x \succ y$. However, $|x^-| = |y^-| = 1$, and moreover, $(y^-)^- \neq \emptyset$ and $(x^-)^- = \emptyset$. By **C-DP**, we would need $y \succ x$; the property is thus violated.

Table 5. Lex-lifting properties. WF and AU stands for well-formed and att-unitary, respectively.

	Abs	**Ind**	**VP**	**SC**	**CP**	**QP**	**CT**	**SCT**	**DP**
C-	All	All	AU	WF $\wedge$ AU	None	None	None	None	None
AC-	All	WF $\vee$ AU	All	All	AU	AU	AU	AU	AU

5. Conclusion and Future Work

We investigated ranking-based semantics in the context of CAFs, where claims constitute an extension to the abstract structure of the arguments, and devised a method for lifting an argument-ranking to the level of the claims. To characterise such a lifting, we reformulated some of the classical properties for ranking-based semantics so as to make them suitable also for CAFs. In detail, we provide two interpretations for each property: one relies solely on claims, while the other takes into account arguments with the same claim. Table 5 summarises our results, i.e. which properties hold for which classes of CAFs. We suppose for each property of the claim-ranking, the corresponding property on the argument-ranking to hold.

This work has to be seen as a first approach towards ranking semantics on the claim level, hence it can be extended in many directions. First, instead of relying on the lifting of the ranking from arguments to claims, we could also devise a ranking-based semantics directly on claims (e.g. by exploiting the logical structure of arguments). Second, ranking-based semantics for CAFs might be induced from scores assigned to arguments (and claims); existing ranking-based semantics (e.g. [2]) could be used for this purpose. This requires a method for aggregating the values and assigning a score to coalitions of arguments (in this context, the notion of robustness [10] is of interest). Further avenues for research include a complexity analysis, fuzzy approaches (see, e.g. [11]), relations to the axioms from [12], and the connection of lex-liftings to the concept of Galois connections [13].

References

[1] Elise Bonzon, Jérôme Delobelle, Sébastien Konieczny, and Nicolas Maudet. A Comparative Study of Ranking-Based Semantics for Abstract Argumentation. In *AAAI 2016*, pages 914–920.

[2] Stefano Bistarelli and Carlo Taticchi. Power index-based semantics for ranking arguments in abstract argumentation frameworks. *Intelligenza Artificiale*, 13(2):137–154, 2019.

[3] Henry Prakken and Michiel De Winter. Abstraction in argumentation: Necessary but dangerous. In *COMMA 2018*, pages 85–96.

[4] Wolfgang Dvořák and Stefan Woltran. Complexity of abstract argumentation under a claim-centric view. In *AAAI 2019*, pages 2801–2808.

[5] Wolfgang Dvořák, Anna Rapberger, and Stefan Woltran. On the relation between claim-augmented argumentation frameworks and collective attacks. In *ECAI 2020*.

[6] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *AIJ*, 171(5–6):286–310, 2007.

[7] Bruno Yun, Srdjan Vesic, and Madalina Croitoru. Ranking-Based Semantics for Sets of Attacking Arguments. In *AAAI 2020*, pages 3033-3040.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *AIJ*, 77(2):321–357, 1995.

[9] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In *SUM 2013*, pages 134–147.

[10] Stefano Bistarelli, Francesco Santini, and Carlo Taticchi. Local expansion invariant operators in argumentation semantics. In *CLAR 2018*, pages 45–62.

[11] Esther Anna Corsi and Christian G. Fermüller. Connecting fuzzy logic and argumentation frames via logical attack principles. *Soft Comput.*, 23(7):2255–2270, 2019.

[12] Leila Amgoud and Jonathan Ben-Naim. Argumentation-based ranking logics. In AAMAS 2015, pages 1511–1519.

[13] Ferdinand Börner. Basics of Galois connections. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, pages 38–67.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200498

Basic Beliefs and Argument-Based Beliefs in Awareness Epistemic Logic with Structured Arguments

Alfredo BURRIEZA [a], Antonio YUSTE-GINEL [a,1]
[a] *Department of Philosophy, University of Malaga, Spain*

Abstract. There are two intuitive principles governing belief formation and argument evaluation that can potentially clash. After arguing that adopting them unrestrictedly leads to an infinite regress, we propose a formal framework in which qualified versions of both principles can be subscribed without falling into such a regress. The proposal integrates tools from two different traditions: structured argumentation and awareness epistemic logic. We show that our formalism satisfies certain rationality postulates and argue that the rest of them can be seen as too ideal when modelling resource-bounded agents.

Keywords. epistemic logic, structured argumentation, awareness logic, beliefs

1. Introduction

There exists certain tension between the formation of some epistemic attitudes of an agent and the way she assesses her available arguments. For the sake of simplicity, we will restrict our attention to the case of beliefs in what follows. The mentioned tension arises when one tries to embrace two principles that, when taken separately, seem to be intuitively acceptable:

P1 The beliefs of an agent should be partially determined by the evaluation she performs of her available arguments. To be more precise, if an agent is considering her doxastic attitude towards a sentence φ, she should first assess her available arguments about φ and then form her belief consequently (for instance, by believing φ if she owns an accepted argument in favour of φ).[2] In short: belief formation is conditioned by argument evaluation.

P2 When an agent assesses her available arguments, she should take into account her beliefs with respect to the premises. In this sense, arguments with believed premises should be taken to be stronger by the agent than arguments whose premises are not believed. In short: argument evaluation is conditioned by belief formation.

[1] Corresponding Author: Office 522, Department of Philosophy, Faculty of Humanities, University of Málaga, 29010, Spain; E-mail: antonioyusteginel@gmail.com.

[2] The term *accepted* is extremely vague at this point, but it will be discussed and clarified later on.

Adopting P1 and P2 unrestrictedly leads to an infinite regress. To see this, let us examine the following fictional dialogue with an agent embracing P1 and P2. We start the conversation by asking: "why do you believe φ?". By applying P1, she would reply something like: "because I own an accepted argument α that concludes φ". We could ask her, in turn: "why do you accept argument α?". The agent might reply, applying P2: "because I believe that its premises $\mathsf{Prem}(\alpha) = \varphi_1, ..., \varphi_n$ are true". Then we would ask: "why do you believe so?" and she would invoke P1 again to say that she owns accepted arguments $\alpha_1, ..., \alpha_n$ concluding $\varphi_1, ..., \varphi_n$. It is easy to see that this conversation could go on indefinitely.

It is worth saying that an analogous form of regression is found in the epistemological literature about the foundation of epistemic justification. Concretely, it is used as a classical argument for foundationalist theories of epistemic justification [1], in which we found inspiration for the present work. Besides, it is interesting to note that different works from the fields of formal argumentation and epistemic logic have separately subscribed different versions of P1 or P2. Let us just mention and briefly comment some of them.

Regarding P1 within formal argumentation, the idea of founding the beliefs (or knowledge) of agents on the evaluation they perform of their available arguments is already present in the seminal work of Dung [2]. This idea is recovered and further developed by frameworks of structured argumentation (e.g. [3,4]), where the sentences believed by the agent can be explicitly stated. Concurrently, epistemic logic has recently focused on the problem of including the –heretofore ignored– justification component into its formal models of knowledge and belief. This has been done in multiple manners, among which we can distinguish between syntactic and semantic approaches –where the adjectives *syntactic* and *semantic* refer to the choices for modelling justification. As for the first group of approaches, it is customary to employ justification logic (e.g. [5,6,7]). As for the second one, they have focused on how to ground the beliefs and knowledge of an agent in (possibly conflicting) pieces of evidence [8,9]. Additionally, some works (among others [10,11]) have mixed tools from formal argumentation and epistemic logic in order to develop their particular view of P1.

Regarding P2, we could say that its explicit acceptance is less spread throughout the literature. Nevertheless, in formal argumentation the idea of ordering sets of premises according to their reliability (see Section 1.2 of [12] and the references given there) can be understood as a version of P2. Besides, some works in justification logic [6,7] define the acceptance of a complex piece of evidence as the agent having a (modal) belief of its premises being true.

The main aim of this paper is to present a simple formalism (Section 2) that allows embracing explicitly qualified versions of P1 and P2 without falling into the mentioned regress (Section 3). We do so by integrating tools from awareness epistemic logic and formal argumentation. Moreover, and locating our work in the field of epistemic logic, we are interested in resource-bounded agents. This implies overcoming at least two problems: i) the classical problem of logical omniscience and ii) certain idealizations that underlie structured argumentation formalisms and that have recently been examined critically [13]. In particular, we drop the extended assumption that agents generate *all* well-shaped arguments from a given knowledge base and analyse the (negative) effects of this choice on the satisfiability of [3]'s rationality postulates (Section 4).

2. An Awareness Logic for Belief and Argumentation

The main ingredients of our logic for belief and argumentation are: (i) epistemic logic [14,15,16], a well-known tool for modelling qualitatively beliefs and knowledge of several agents; (ii) its extension with awareness operators [17] to model explicit beliefs, which allows overcoming the problem of logical omniscience (see e.g. Section 9 of [15]) and (iii) ideas taken from ASPIC$^+$ [4] to model structured arguments. [3] Among the most relevant features of ASPIC$^+$, we highlight the following ones: a) it deals with both deductive and non-deductive (defeasible) arguments; capturing also different kinds of attacks among arguments (attacking the premises, the conclusion or the inference link) and b) it has been shown to be comprehensive, in the sense that many other proposals in structured argumentation and non-monotonic logic can be seen as special cases of it (see [4]).

Definition 1 (Language)**.** *Let $\mathbb{P}$ be a fixed and denumerable set of* atoms*; the language $\mathscr{L}_{\mathsf{BA}}$ is defined as the the pair $(\mathscr{F},\mathscr{A})$ of* formulas *and* arguments *which are respectively generated by the following grammars:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box\varphi \mid \mathsf{aware}(\alpha) \mid \mathsf{conc}(\alpha) = \varphi \mid$$

$$\mid \mathsf{strict}(\alpha) \mid \mathsf{undercuts}(\alpha,\alpha) \mid \mathsf{wellshap}(\alpha) \qquad p \in \mathbb{P}, \alpha \in \mathscr{A}$$

$$\alpha ::= \langle\varphi\rangle \mid \langle\alpha_1,...,\alpha_n \twoheadrightarrow \varphi\rangle \mid \langle\alpha_1,...,\alpha_n \Rightarrow \varphi\rangle \qquad \varphi \in \mathscr{L}_{\mathsf{BA}}$$

Elements of $\mathbb{P}$ represent *factual atomic sentences*, i.e. sentences about states of affairs whose truth value is agent-independent. The rest of boolean connectives are defined and read as usual. Let us adopt the following intuitive reading for the remaining formulas and arguments: $\langle\varphi\rangle$ is an atomic argument, whose only premise and conclusion is φ. $\langle\alpha_1,...,\alpha_n \twoheadrightarrow \varphi\rangle$ represents an argument whose last inference link strictly (deductively) concludes φ. $\langle\alpha_1,...,\alpha_n \Rightarrow \varphi\rangle$ represents an argument whose last inference link defeasibly concludes φ. $\Box\varphi$ means that the agent has a basic-implicit belief that φ. Basic-implicit beliefs accept different intuitive readings, both positive (reasonable assumptions, sound observations, etc) and negative (prejudices, biases, etc). The adjective *basic* underlines the idea that their source is not inferential, while *implicit* points out that they are closed under logical consequence. $\mathsf{aware}(\alpha)$ means that the agent is aware of argument α. As usual in awareness logic [17], the operator aware admits several informal readings. For the special case of atomic arguments ($\mathsf{aware}(\langle\varphi\rangle)$), we propose to read them as follows: “the agent recognizes her doxastic attitude toward φ through non-inferential methods ”. $\mathsf{wellshap}(\alpha)$ means that argument α is well-shaped, i.e. it has been constructed properly for the sentence it says it argues for. In more detail, every subargument of α using a strict inference link has been produced by the application of a valid deductive rule and every subargument of α using a defeasible inference link has been produced using an accepted defeasible rule. $\mathsf{conc}(\alpha) = \varphi$ means that φ is the conclusion of α. $\mathsf{undercuts}(\alpha,\beta)$ means that α undercuts β (i.e. α attacks β’s inference link). Finally, $\mathsf{strict}(\alpha)$ means that α does not make use of any defeasible rule, i.e. α only contains atomic arguments and arguments formed with $\twoheadrightarrow$.

[3] We remark that the formalism below does not intend to be an alternative to ASPIC$^+$, but rather an application of it to solve the conceptual problem presented in the introduction.

Definition 2 (Argument structure [4]). *Let us define the following meta-syntactic functions for analysing an argument's structure:*

$\mathsf{Prem}(\alpha)$ returns the *premises* of α and it is defined as follows: $\mathsf{Prem}(\langle\varphi\rangle)=\{\varphi\}$, $\mathsf{Prem}(\langle\alpha_1,...,\alpha_n\hookrightarrow\varphi\rangle)=\mathsf{Prem}(\alpha_1)\cup...\cup\mathsf{Prem}(\alpha_n)$ where $\hookrightarrow\in\{\twoheadrightarrow,\Rightarrow\}$. **Example:** $\mathsf{Prem}(\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle)=\{p,q\}$.

$\mathsf{Conc}(\alpha)$ returns the *conclusion* of α and it is defined as follows $\mathsf{Conc}(\langle\varphi\rangle)=\{\varphi\}$ and $\mathsf{Conc}(\langle\alpha_1,...,\alpha_n\hookrightarrow\varphi\rangle)=\{\varphi\}$ where $\hookrightarrow\in\{\twoheadrightarrow,\Rightarrow\}$. Note that arguments of ASPIC$^+$ have unique conclusions (differently to what happens, for instance, in justification logic [6] where the $+$ operator allows for arguments with multiple conclusions). **Example:** $\mathsf{Conc}(\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle)=s\vee t$.

$\mathsf{sub}_{\mathsf{A}}(\alpha)$ returns the *subarguments* of α and it is defined as follows: $\mathsf{sub}_{\mathsf{A}}(\langle\varphi\rangle)=\{\langle\varphi\rangle\}$ and $\mathsf{sub}_{\mathsf{A}}(\langle\alpha_1,...,\alpha_n\hookrightarrow\varphi\rangle)=\{\langle\alpha_1,...,\alpha_n\hookrightarrow\varphi\rangle\}\cup\mathsf{sub}_{\mathsf{A}}(\alpha_1)\cup...\cup\mathsf{sub}_{\mathsf{A}}(\alpha_n)$ where $\hookrightarrow\in\{\twoheadrightarrow,\Rightarrow\}$. **Example:** $\mathsf{sub}_{\mathsf{A}}(\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle)=\{\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle,\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle,\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle,\langle p\rangle,\langle q\rangle\}$.

$\mathsf{TopRule}(\alpha)$ returns the *top rule* of α, i.e. the last one applied in the formation of α. It is defined as follows: $\mathsf{TopRule}(\langle\varphi\rangle)$ is left undefined, $\mathsf{TopRule}(\langle\alpha_1,...,\alpha_n\twoheadrightarrow\varphi\rangle)=\mathsf{TopRule}(\langle\alpha_1,...,\alpha_n\Rightarrow\varphi\rangle)=((\mathsf{Conc}(\alpha_1),...,\mathsf{Conc}(\alpha_n)),\varphi)$. **Example:** $\mathsf{TopRule}(\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle)=(s,s\vee t)$.

$\mathsf{DefRule}(\alpha)$ returns the set of *defeasible rules* of α and it is defined as $\mathsf{DefRule}(\langle\varphi\rangle)=\emptyset$, $\mathsf{DefRule}(\langle\alpha_1,...,\alpha_n\twoheadrightarrow\varphi\rangle)=\mathsf{DefRule}(\alpha_1)\cup...\cup\mathsf{DefRule}(\alpha_n)$ and $\mathsf{DefRule}(\langle\alpha_1,...,\alpha_n\Rightarrow\varphi\rangle)=\{((\mathsf{Conc}(\alpha_1),...,\mathsf{Conc}(\alpha_n)),\varphi)\}\cup\mathsf{DefRule}(\alpha_1)\cup...\cup\mathsf{DefRule}(\alpha_n)$. **Example:** $\mathsf{DefRule}(\langle\langle\langle\langle p\rangle,\langle q\rangle\Rightarrow r\rangle\Rightarrow s\rangle\twoheadrightarrow s\vee t\rangle)=\{((p,q),r),((r),s)\}$.

Let us also define *single negations*, for any $\varphi\in\mathscr{L}_{\mathsf{BA}}$: $\sim\varphi:=\psi$ if φ is of the form $\neg\psi$; else $\sim\varphi:=\neg\varphi$.

Definition 3 (Model). *A model for $\mathscr{L}_{\mathsf{BA}}$ is a tuple $M=(W,\mathscr{B},\mathscr{O},\mathscr{D},\mathfrak{n},||\cdot||)$ where:*

- $W\neq\emptyset$ *is a set of* possible worlds
- $\mathscr{B}\subseteq W$ *and* $\mathscr{B}\neq\emptyset$ *is the set of* doxastically indistinguishable worlds
- $\mathscr{O}\subseteq\mathscr{A}$ *is the (finite) set of* available arguments *or the* awareness set of the agent
- $\mathscr{D}\subseteq\mathscr{L}^n_{\mathsf{BA}}\times\mathscr{L}_{\mathsf{BA}}$ *(with $n\in\mathbb{N}$) is a finite set of* accepted defeasible rules *s.t. if $((\varphi_1,...,\varphi_n),\varphi)\in\mathscr{D}$, then $\{\varphi_1,...,\varphi_n,\varphi\}\nvdash_0\bot$; where $\vdash_0$ is the consequence relation of classical propositional logic*
- $\mathfrak{n}:\mathscr{D}\rightarrow\mathbb{P}$ *is a (possibly partial)* naming function *for defeasible rules, where $\mathfrak{n}(R)$ informally means "the defeasible rule R is applicable"*
- $||\cdot||:\mathbb{P}\rightarrow\wp(W)$ *is an* atomic valuation

Definition 4 (Truth). *Formulas of $\mathscr{L}_{\mathsf{BA}}$ are interpreted in pointed models (M,w) where $w\in W$. $M,w\vDash\varphi$ means that φ is true in (M,w). $\vDash$ is defined for every kind of formulas as follows (we omit the clauses for propositional variables and boolean connectives):*

- $M,w\vDash\Box\varphi$ iff for all $w'\in W$: $w'\in\mathscr{B}$ implies $M,w'\vDash\varphi$
- $M,w\vDash\mathsf{aware}(\alpha)$ iff $\alpha\in\mathscr{O}$
- $M,w\vDash\mathsf{conc}(\alpha)=\varphi$ iff $\mathsf{Conc}(\alpha)=\varphi$
- $M,w\vDash\mathsf{strict}(\alpha)$ iff $\mathsf{DefRule}(\alpha)=\emptyset$

- $M,w \vDash \mathsf{undercuts}(\alpha,\beta)$ iff $\mathsf{Conc}(\alpha) = \sim\mathfrak{n}(\mathsf{TopRule}(\beta))$[4]
- $M,w \vDash \mathsf{wellshap}(\langle\varphi\rangle)$
- $M,w \vDash \mathsf{wellshap}(\langle\alpha_1,\ldots,\alpha_n \twoheadrightarrow \varphi\rangle)$ iff $M,w \vDash \mathsf{wellshap}(\alpha_i)$ for every $1 \leq i \leq n$ and $\{\mathsf{Conc}(\alpha_1),\ldots,\mathsf{Conc}(\alpha_n)\} \vdash_0 \varphi$
- $M,w \vDash \mathsf{wellshap}(\langle\alpha_1,\ldots,\alpha_n \Rightarrow \varphi\rangle)$ iff $M,w \vDash \mathsf{wellshap}(\alpha_i)$ for every $1 \leq i \leq n$ and $((\mathsf{Conc}(\alpha_1),\ldots,\mathsf{Conc}(\alpha_n)),\varphi) \in \mathscr{D}$

Validity ($\vDash \varphi$) and local logical consequence ($\Gamma \vDash \varphi$) are defined as usual [18]. Note that our way of representing basic-implicit beliefs is equivalent (in the single-agent case) to have a Kripke model where the accessibility relation is serial, transitive and euclidean; therefore $\Box$ satisfies $KD45$ axioms (see [10,16]). Regarding the truth clauses for $\mathsf{conc}(\alpha) = \varphi$ and $\mathsf{strict}(\alpha)$; it is easy to show that these kinds of formulas are model independent (since they are based on argument structure, see Definition 2). This implies that, for these kinds of formulas they are true in a pointed model iff they are valid. Furthermore, note that the clause for $\Box\varphi$, $\mathsf{undercuts}(\alpha,\beta)$, $\mathsf{aware}(\alpha)$ and $\mathsf{wellshap}(\alpha)$ makes the satisfiability of these kinds of formulas world-independent, i.e. they are true in a pointed model if they are globally true in the model. Consequently, we have that $\star \to \Box\star$ and $\neg\star \to \Box\neg\star$ are valid schemata, where $\star \in \{\mathsf{aware}(\alpha), \mathsf{conc}(\alpha) = \varphi, \mathsf{undercuts}(\alpha,\beta), \mathsf{wellshap}(\alpha), \mathsf{strict}(\alpha)\}$. Informally, this amounts to assume that: i) awareness of arguments is fully introspective w.r.t. basic-implicit beliefs and ii) the agent is logically competent w.r.t. the arguments she is aware of. However, and unlike what is usual in structured argumentation [4]; our agent will not work with the whole set of all well-shaped arguments (which is by definition infinite), but rather with the (finite) set of arguments that she is aware of.

3. Basic Beliefs and AB-Beliefs in $\mathscr{L}_{\mathsf{BA}}$

In order to solve the tension between P1 and P2, we distinguish between basic-explicit beliefs (Definition 5) and argument-based beliefs (AB-Beliefs, for short; Definition 9). While the notion of basic belief (both its implicit and explicit versions) only needs some informal clarification (Section 3.1); AB-beliefs force us to import some concepts from formal argumentation (sections 3.2, 3.3 and 3.4), especially from ASPIC$^+$. Most of the central concepts used in ASPIC$^+$(or our adaptations) are definable in $\mathscr{L}_{\mathsf{BA}}$.

3.1. Basic Beliefs

Recall that basic-implicit beliefs are represented through the primitive, normal modal operator $\Box$, hence they suffer from logical omniscience: ($M,w \vDash \Box\psi$ for all $\psi \in \Gamma$ and $\Gamma \vDash \varphi$) implies $M,w \vDash \Box\varphi$. This property has been extensively discussed in the epistemic logic literature, and it has been argued to be problematic when dealing with resource-bounded agents (see e.g. [17,15, Chapter 9]). The pitfall can be overcome by distinguishing between basic-implicit beliefs ($\Box\varphi$) and basic-explicit beliefs ($\Box^e\varphi$) following the awareness approach [17]:

Definition 5 (Basic-explicit beliefs)**.** $\Box^e\varphi := \Box\varphi \wedge \mathsf{aware}(\langle\varphi\rangle)$

[4] Note that we do not need to consider $\mathsf{undercuts}$ as a primitive operator, since it could be defined through a (simpler) operator that captures the meaning of $\mathfrak{n}$. We make this choice for the sake of succinctness.

Informally, basic-explicit beliefs can be generally understood as actual beliefs of the agent whose justification is not inferential (it comes from other epistemic phenomena, such as observations or reliable communications). In fact, we can think of many beliefs that a (reasonable) epistemic agent may have that need no arguments to be justified. Imagine, for instance, that you walk into your classroom and you see three students in there. Consequently, you form the belief that “there are three student in the classroom”. Do you need any complex argument to justify such a belief? Our claim is that, in principle, you do not. Indeed, you can form arguments supporting the proposition if someone would question your belief. But, for the agent herself (you, in this case), mere observation is a good enough reason to believe that there are three students in the classroom.

3.2. Doxastic Preference

Premises are usually understood as a source of argument strength [12], regarding the *support dimension* (see [12] for the distinction between the three dimensions or tiers of argument strength). In structured argumentation [4,12], this is often modelled by stratifying a given set of formulas into different preference classes. Such a hierarchy is usually assumed to be primitive and its nature is abstracted away from the modelling process. Let us now show how basic beliefs induce a meaningful hierarchy of this kind. Let (M,w) be a pointed model and let $\alpha \in \mathscr{A}$, we can distinguish between three types of premises of α: $\mathsf{Prem}(\alpha) = \mathsf{Prem}^{+}(\alpha) \cup \mathsf{Prem}^{?}(\alpha) \cup \mathsf{Prem}^{-}(\alpha)$ where each component is defined as follows $\mathsf{Prem}^{+}(\alpha) := \{\varphi \in \mathsf{Prem}(\alpha) \mid M,w \vDash \Box\varphi\}$ (the set of trusted or believed premises); $\mathsf{Prem}^{?}(\alpha) := \{\varphi \in \mathsf{Prem}(\alpha) \mid \neg\Box\varphi \wedge \neg\Box\neg\varphi\}$ (the set of premises considered contingent by the agent) and $\mathsf{Prem}^{-}(\alpha) := \{\varphi \in \mathsf{Prem}(\alpha) \mid M,w \vDash \Box\neg\varphi\}$ (the set of disbelieved premises). The three kind of premises are pairwise disjoint (due to the consistency of basic beliefs) and possibly empty. Furthermore, this distinction induces another one within the set of all arguments $\mathscr{A} = \mathscr{A}^{+} \cup \mathscr{A}^{?} \cup \mathscr{A}^{-}$ where each component is defined as follows: $\mathscr{A}^{+} := \{\alpha \in \mathscr{A} \mid \mathsf{Prem}(\alpha) = \mathsf{Prem}^{+}(\alpha)\}$; $\mathscr{A}^{?} := \{\alpha \in \mathscr{A} \mid \mathsf{Prem}(\alpha) = \mathsf{Prem}^{+}(\alpha) \cup \mathsf{Prem}^{?}(\alpha), \mathsf{Prem}^{?} \neq \emptyset\}$ and $\mathscr{A}^{-} = \{\alpha \in \mathscr{A} \mid \mathsf{Prem}^{-}(\alpha) \neq \emptyset\}$.[5] It seems natural to assume the following preference ordering between the three classes of arguments $\mathscr{A}^{+} \sqsupset_p \mathscr{A}^{?} \sqsupset_p \mathscr{A}^{-}$, that can be lowered to arguments straightforwardly: $\alpha >_p \beta$ iff $\alpha \in \mathscr{A}^{'}$, $\beta \in \mathscr{A}^{''}$ and $\mathscr{A}^{'} \sqsupset_p \mathscr{A}^{''}$ with $','' \in \{+,?,-\}$. The relation $>_p$ is precisely our qualified version of P2: *argument evaluation is conditioned by **basic** belief formation*. Interestingly enough, this relation can be captured in $\mathscr{L}_{\mathsf{BA}}$, as shown in [19], using the following shorthands: $\mathsf{accept}(\alpha) := \bigwedge_{\varphi\in\mathsf{Prem}(\alpha)} \Box\varphi$[6] (*basic acceptance*); $\mathsf{reject}(\alpha) := \bigvee_{\varphi\in\mathsf{Prem}(\alpha)} \Box\neg\varphi$ (*basic rejection*); $\mathsf{prem}^{>}(\alpha,\beta) := (\mathsf{accept}(\alpha) \wedge \neg\mathsf{accept}(\beta)) \vee (\neg\mathsf{reject}(\alpha) \wedge \mathsf{reject}(\beta))$; $\mathsf{prem}^{\approx}(\alpha,\beta) := \neg\mathsf{prem}^{>}(\alpha,\beta) \wedge \neg\mathsf{prem}^{>}(\beta,\alpha)$:

Proposition 1. *Let (M,w) be a pointed model, we have that $M,w \vDash \mathsf{prem}^{>}(\alpha,\beta)$ iff $\alpha >_p \beta$.*

[5] The *lifting principle* applied in order to go from preferences between premises to preference between arguments is the so-called *min-min* principle [12]. Note that basic beliefs permit more fine-grained distinctions regarding the relative strength of arguments. For instance, we could distinguish within $\mathscr{A}^{?}$ between arguments whose premises are jointly considered a doxastic possibility $\mathscr{A}^{?+} := \{\alpha \in \mathscr{A} \mid \Diamond \bigwedge_{\varphi\in\alpha} \varphi\}$ and arguments that do not enjoy this property $\mathscr{A}^{?-} := \mathscr{A}^{?}/\mathscr{A}^{?+}$. Nonetheless, we adopt the current division for simplicity.

[6] This definition is inspired by [6].

Premises are not the only source of argument strength regarding the *support dimension*. The other main source are inference links. In order to keep things simple, we adopt a minimal (yet intuitively acceptable) principle to assess inference links: *ceteris paribus*, strict arguments should be preferred to defeasible arguments. This principle can be captured as follows: let $\mathscr{A}^{st} := \{\alpha \in \mathscr{A} \mid \mathsf{DefRule}(\alpha) = \emptyset\}$ and let $\mathscr{A}^{df} := \mathscr{A}/\mathscr{A}^{st}$, we can define new preference classes by intersecting separately both sets with the previous hierarchy. Furthermore, we assume the following natural preference ordering:

$$\mathscr{A}^{+} \cap \mathscr{A}^{st} \sqsupset_{il} \mathscr{A}^{+} \cap \mathscr{A}^{df} \sqsupset_{il} \mathscr{A}^{?} \cap \mathscr{A}^{st} \sqsupset_{il} \mathscr{A}^{?} \cap \mathscr{A}^{df} \sqsupset_{il} \mathscr{A}^{-} \cap \mathscr{A}^{st} \sqsupset_{il} \mathscr{A}^{-} \cap \mathscr{A}^{df}$$

The new preference ordering can be lowered to arguments as follows: $\alpha >_{il} \beta$ iff $\alpha \in \mathscr{A}'$, $\beta \in \mathscr{A}''$ and $\mathscr{A}' \sqsupset_{il} \mathscr{A}''$ with $\mathscr{A}', \mathscr{A}'' \in \{\mathscr{A}^{+} \cap \mathscr{A}^{st}, \mathscr{A}^{+} \cap \mathscr{A}^{df}, \mathscr{A}^{?} \cap \mathscr{A}^{st}, \mathscr{A}^{?} \cap \mathscr{A}^{df}, \mathscr{A}^{-} \cap \mathscr{A}^{st}, \mathscr{A}^{-} \cap \mathscr{A}^{df}\}$. Note that the relation satisfies $>_{p} \subset >_{il}$. Besides, it can be captured in $\mathscr{L}_{\mathsf{BA}}$ through the following schemes: $\mathsf{strict}^{>}(\alpha,\beta) := \mathsf{strict}(\alpha) \wedge \neg\mathsf{strict}(\beta)$; $\alpha > \beta := \mathsf{prem}^{>}(\alpha,\beta) \vee \big(\mathsf{prem}^{\approx}(\alpha,\beta) \wedge \mathsf{strict}^{>}(\alpha,\beta)\big)$; $\alpha \geq \beta := \neg(\beta > \alpha)$; $\alpha \approx \beta := \alpha \geq \beta \wedge \beta \geq \alpha$.

Proposition 2. *Let (M,w) be a pointed model, we have that $M,w \vDash \alpha \geq \beta$ iff $\alpha \geq_{il} \beta$.*

Let us stress two points regarding the preference ordering $\geq_{il}$ which are important for the study of [3]'s rationality postulates. First, it is *reasonable* in the sense of [4]. Second, $\geq_{il}$ is a total preorder on $\mathscr{A}$. This fact, expressed in the object language has the form of the following valid schemas, for every $\alpha,\beta,\delta \in \mathscr{A}$: $\vDash (\alpha \geq \beta \wedge \beta \geq \delta) \rightarrow \alpha \geq \delta$ (transitivity) and $\vDash \alpha \geq \beta \vee \beta \geq \alpha$ (connectedness).

3.3. Attack and Defeat

Agents do not assess arguments in isolation, or merely pairwise, checking if certain features of the involved premises and inference links are good enough to support the conclusion. Another important dimension of argument strength is called the *dialectical tier* which, following [12], is "mainly represented by relations of argumentative attack and defeat between arguments". $\mathscr{L}_{\mathsf{BA}}$ is rich enough to capture the three customary kinds of attacks discussed in structured argumentation:

Definition 6 (Argument attack). *Given a pointed model (M,w) and $\alpha,\beta \in \mathscr{A}$: we say that α undermines β iff $M,w \vDash \mathsf{undermines}(\alpha,\beta)$, where $\mathsf{undermines}(\alpha,\beta) := \bigvee_{\varphi \in \mathsf{Prem}(\beta)} \mathsf{conc}(\alpha) = {\sim}\varphi$; α rebuts β iff $M,w \vDash \mathsf{rebuts}(\alpha,\beta)$, where $\mathsf{rebuts}(\alpha,\beta) := \bigvee_{\langle \beta_1,\ldots,\beta_n \hookrightarrow \varphi\rangle \in \mathsf{sub}_{\mathsf{A}}(\beta)} \mathsf{conc}(\alpha) = {\sim}\varphi$ where $\hookrightarrow \in \{\twoheadrightarrow, \Rightarrow\}$; and α undercuts β iff $M,w \vDash \mathsf{undercuts}^{*}(\alpha,\beta)$, where $\mathsf{undercuts}^{*}(\alpha,\beta) := \bigvee_{\beta' \in \mathsf{sub}_{\mathsf{A}}(\beta)} \mathsf{undercuts}(\alpha,\beta')$.*

Our definition of attack integrates a notion of *unrestricted rebuttal*, in the sense that rebuttals are permitted on any kind of complex argument. This is indeed polemic. While the creators of ASPIC^{+}, amongst others, only allow rebuttals on the application of defeasible rules; others have argued that the unrestricted notion seems natural in dialectical contexts [20,21]. Moreover, [21] requires the rebutted argument to be defeasible (non-strict) while [20] does not require it but, in turn, this feature is implied by their setting. We permit *completely unrestricted rebuttals* for a simple reason: since awareness sets do not exhibit any closure property, in absence of completely unrestricted rebuttal direct consistency fails (see Section 4 for more details).

From an agent perspective, some attacks must be disregarded. Imagine, for instance, that an agent is aware of $\langle\langle p\rangle \twoheadrightarrow p \vee q\rangle$ and that she accepts it in a doxastic sense ($\Box p$). She then receives an undermining argument $\langle\langle r\rangle \Rightarrow \neg p\rangle$ but she does not accept it (she does not believe that r). It seems that such an attack must not be considered a threat for the agent. Consequently, the notion of *defeat* should take into account the preference relation defined above. We import the definition of defeat from ASPIC$^+$to our object language, introducing two essential differences. First, preferences do play a role when determining the success of undercutting attacks (the reason for doing so is offered below). Second, the agent only considers defeats among the well-shaped arguments that she is aware of, capturing that although her resources are bounded (w.r.t. argument generation) they are locally well applied. We proceed in two steps: defining a successful counterpart for each type of attack and adding the awareness/well-shapedness requirement.

Definition 7 (Successful attack, defeat)**.** *Given a pointed model* (M,w) *and two arguments* $\alpha,\beta \in \mathscr{A}$ *we say that:* α successfully undermines β *iff* $M,w \vDash \mathsf{SuUndermines}(\alpha,\beta)$*, where* $\mathsf{SuUndermines}(\alpha,\beta) := \bigvee_{\varphi\in\mathsf{Prem}(\beta)}(\mathsf{conc}(\alpha) = {\sim}\varphi \wedge \alpha \geq \langle\varphi\rangle)$*;* α successfully rebuts β *iff* $M,w \vDash \mathsf{SuRebuts}(\alpha,\beta)$ *where* $\mathsf{SuRebuts}(\alpha,\beta) := \bigvee_{\langle\beta_1,\ldots,\beta_n\hookrightarrow\varphi\rangle\in\mathsf{sub_A}(\beta)}(\mathsf{conc}(\alpha) = {\sim}\varphi \wedge \alpha \geq \langle\beta_1,\ldots,\beta_n \hookrightarrow \varphi\rangle)$*;* α successfully undercuts β *iff* $M,w \vDash \mathsf{SuUndercuts}(\alpha,\beta)$*, where* $\mathsf{SuUndercuts}(\alpha,\beta) := \bigvee_{\beta'\in\mathsf{sub_A}(\beta)}(\mathsf{undercuts}(\alpha,\beta') \wedge \alpha \geq \beta')$ *and, finally, we say that* α defeats β *iff* $M,w \vDash \mathsf{defeat}(\alpha,\beta)$*, where* $\mathsf{defeat}(\alpha,\beta) := (\mathsf{SuUndermines}(\alpha,\beta) \vee \mathsf{SuRebuts}(\alpha,\beta) \vee \mathsf{SuUndercuts}(\alpha,\beta)) \wedge \mathsf{aware}(\alpha) \wedge \mathsf{aware}(\beta) \wedge \mathsf{wellshap}(\alpha) \wedge \mathsf{wellshap}(\beta)$.

As mentioned above, it has been argued that undercutting attacks always succeed (independently from what the preferences are) [4]. This may lead to counter-intuitive cases in the current setting. Taking the same example that [4], due to Pollock, suppose that an agent considers that an object is red because she sees that it is red (she is aware of an argument $\langle\langle\mathsf{SeeRed}\rangle \Rightarrow \mathsf{IsRed}\rangle$). Suppose that someone suggests her to consider the undercutting "there might be a red shining, therefore the inference rule you are applying does not hold". This can be modelled by putting into her awareness set an argument $\langle\langle\mathsf{RedLight}\rangle \Rightarrow \neg\mathsf{D}\rangle$ where D is an atomic proposition saying that the defeasible inference rule $((\mathsf{SeeRed}),\mathsf{IsRed})$ is applicable. Suppose however that she believes that there is no such light in the room, $M,w \vDash \Box\neg\mathsf{RedLight}$. It looks that, under this assumption, $\langle\langle\mathsf{RedLight}\rangle \Rightarrow \neg\mathsf{D}\rangle$ is not a good reason to prevent the agent from drawing her initial conclusion that IsRed holds.

3.4. AB-Beliefs

Given a set of well-shaped and owned arguments B, the agent is already able to determine the defeat relation among them. Nevertheless, the question of how to decide which subset(s) of B should be considered *justified* remains still open. This question has been called the *evaluation tier* of argument strength in [12] and it is notoriously solved by applying different semantics to an *argumentation framework* (first introduced by Dung in [2]). Note that each pointed model (M,w) naturally induces a Dung-style argumentation framework [2] (AF, for short), which will be the main construct to define AB-beliefs.

Definition 8 (Associated argumentation framework)**.** *Let* (M,w) *be a pointed model where* $M = (W,\mathscr{B},\mathscr{O},\mathscr{D},\mathfrak{n},||\cdot||)$*. The argumentation framework associated to* (M,w)*,* *denoted by* AF^M *is the pair* $(\mathscr{O}^{ws}, \rightsquigarrow)$ *where* $\mathscr{O}^{ws} := \{\alpha \in \mathscr{O} \mid M, w \vDash \mathsf{wellshap}(\alpha)\}$ *and* $\rightsquigarrow \subseteq \mathscr{O}^{ws} \times \mathscr{O}^{ws}$ *is defined as* $\alpha \rightsquigarrow \beta$ *iff* $M, w \vDash \mathsf{defeat}(\alpha, \beta)$.[7]

The semantics of an AF is usually given in terms of *extensions*, i.e. subsets of $\mathscr{O}^{ws}$ satisfying certain intuitive constraints to be an acceptable set [2]. Given a set of arguments $B \subseteq \mathscr{O}^{ws}$, typical minimal requirements are *conflict-freeness* (there are no $\alpha, \beta \in B$ s.t. $\alpha \rightsquigarrow \beta$) and self-defence (every defeater of members of B is in turn defeated by some member of B). A set of arguments B is a *complete extension* iff it contains precisely the arguments it defends. Finally, the *grounded extension* of AF^M, denoted by $GE(AF^M)$ is the minimal (w.r.t. set inclusion) complete extension. For a more precise definition of these notions and an extensive discussion about the existing semantics, the reader is referred to [22]. Our choice of grounded semantics for defining AB-beliefs is rooted on the arguments presented in [23] for such a decision regarding epistemic reasoning.

Definition 9 (AB-Beliefs)**.** *Let* (M, w) *be a pointed model for* $\mathscr{L}_{\mathsf{BA}}$*, and let* $\varphi \in \mathscr{L}_{\mathsf{BA}}$*, we say that* φ *is AB-believed in* (M, w)*, denoted by* $M, w \vDash \mathsf{B}^{\mathsf{AB}}\varphi$*, iff* $\exists \alpha \in GE(AF^M) : \mathsf{Conc}(\alpha) = \varphi$.

This definition captures our qualified version of P1: ***AB**-belief formation is conditioned by argument evaluation*. Moreover, note that the following schema is valid $\vDash \Box^e \varphi \to \mathsf{B}^{\mathsf{AB}}\varphi$ (i.e. basic-explicit beliefs are a special case of AB-beliefs). AB-beliefs cannot be captured in $\mathscr{L}_{\mathsf{BA}}$. The reason for this is that its definition quantifies over arguments (and sets of arguments, since the grounded extension requires subset-minimality). This inconvenience could be circumvented in several ways that are out of the scope of this paper. Instead, let us just increase $\mathscr{L}_{\mathsf{BA}}$ with a new clause $\mathsf{B}^{\mathsf{AB}}\varphi$, where $\varphi \in \mathscr{L}_{\mathsf{BA}}$, and adopt the truth clause of Definition 9 for the new kind of formulas. In the following example, we illustrate the difference between both kinds of beliefs and how our qualified versions of P1 and P2 work.

Example 1 (Assessing a survey)**.** *A researcher in charge of a survey (in what follows, the agent) is assessing the last report of her team. In particular, the agent is wondering whether a* Claim *follows from some* Data *gathered by her team, as suggested in the report, i.e. she is determining the acceptability of* $\langle\langle \mathsf{Data}\rangle \Rightarrow \mathsf{Claim}\rangle$. *Model* M*, depicted in the top part of Figure 1 shows her implicit doxastic attitudes towards the involved propositions. The bottom-part of the same figure shows the associated AF,* AF^M*, where black arrows represent defeats and dashed arrows represent unsuccessful attacks. Some elements of the model are omitted in the representation (*$\mathscr{O}$*,* $\mathscr{D}$ *and* $\mathfrak{n}$*), but they can be completed by observing the associated AF.*
The head of the laboratory has told the agent to consider the undercutting attack $\langle\langle \neg\mathsf{Honest}\rangle \Rightarrow \neg\mathsf{Reliab}\rangle$*, according to which if her team is not behaving honestly, the defeasible rule* $((\mathsf{Data}), \mathsf{Claim})$ *should be considered suspicious (we fix* $\mathfrak{n}(((\mathsf{Data}), \mathsf{Claim})) = \mathsf{Reliab}$*). Nevertheless, the agent holds a basic-explicit belief that her team is behaving honestly,* $M, w \vDash \Box^e \mathsf{Honest}$*; so she disregards the mentioned undercutting,* $M, w \vDash \neg\mathsf{SuUndercuts}(\langle\langle \neg\mathsf{Honest}\rangle \Rightarrow \neg\mathsf{Reliab}\rangle, \langle\langle \mathsf{Data}\rangle \Rightarrow \mathsf{Claim}\rangle)$*. Moreover, she also considers the strict argument* $\langle\langle \mathsf{Defective} \to \neg\mathsf{Data}\rangle, \langle \mathsf{Defective}\rangle \twoheadrightarrow \neg\mathsf{Data}\rangle$ *ac-*

[7] Given the simplification of the modal semantics we have assumed, it can be shown that for every model M, with domain W, it holds that $AF^{M,w} = AF^{M,w'}$ for every $w, w' \in W$. This remark permits us to refer to $AF^{M,w}$ just as AF^M.

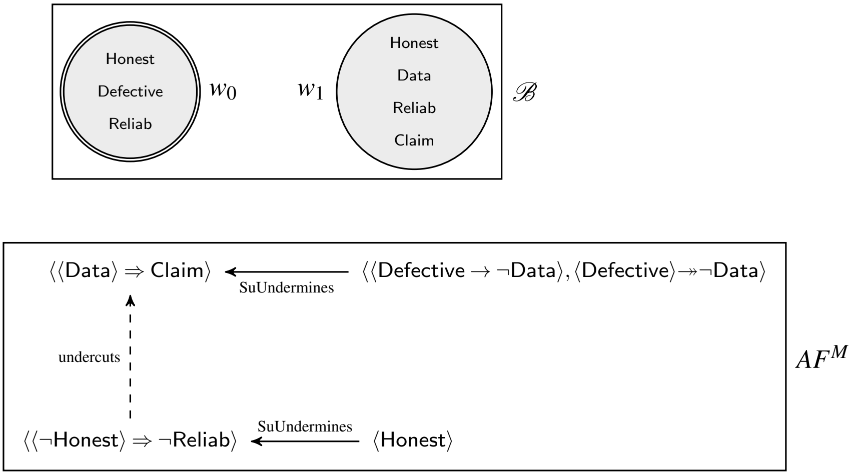

Honest
Defective
Reliab
w_0
w_1
Honest
Data
Reliab
Claim
$\mathscr{B}$
$\langle\langle \mathsf{Data}\rangle \Rightarrow \mathsf{Claim}\rangle$
SuUndermines
$\langle\langle \mathsf{Defective} \rightarrow \neg\mathsf{Data}\rangle, \langle \mathsf{Defective}\rangle \twoheadrightarrow \neg\mathsf{Data}\rangle$
undercuts
$\langle\langle \neg\mathsf{Honest}\rangle \Rightarrow \neg\mathsf{Reliab}\rangle$
SuUndermines
$\langle \mathsf{Honest}\rangle$
AF^M

Figure 1. Pointed model (M, w_0) (top part) and its associated AF, AF^M (bottom part).

cording to which if one of the measure devices used in the study is defective, then the gathered data is not true. Note that this argument undermines $\langle\langle \mathsf{Data}\rangle \Rightarrow \mathsf{Claim}\rangle$. *Due to previous problems with the mentioned device, she considers as doxastically possible a situation where it does not work properly (w_0), hence the undermining succeeds. Consequently, she keeps sceptic about the value of* Claim*:* $M,w \models \neg\mathsf{B}^{\mathsf{AB}}\mathsf{Claim} \wedge \neg\mathsf{B}^{\mathsf{AB}}\neg\mathsf{Claim}$.

4. Rationality Postulates

In [3], Caminada and Amgoud provide a list of rationality postulates that a good argumentation formalism must satisfy. In [4], Modgil and Prakken discuss these postulates in relation to ASPIC$^+$. In this section, we offer sufficient conditions for two of them to be satisfied and argue that the other two are too idealistic in an epistemic logic for resource-bounded agents. First of all, let us formulate the postulates in the current setting. Let AF^M be an associated AF, we say that AF^M satisfies:

- $\mathsf{RP}_{\mathsf{SUB}}$ (*sub-argument closure*) iff for any $\alpha \in GE(AF^M)$, $\mathsf{sub}_{\mathsf{A}}(\alpha) \subseteq GE(AF^M)$
- $\mathsf{RP}_{\mathsf{DC}}$ (*direct consistency*) iff $\nexists \varphi \in \mathscr{L}_{\mathsf{BA}}$: $\varphi, \sim\varphi \in \mathsf{Conc}(GE(AF^M))$[8]
- $\mathsf{RP}_{\mathsf{CL}}$ (*closure under strict rules*) iff for all $\varphi \in \mathscr{L}_{\mathsf{BA}}$ s.t. $\mathsf{Conc}(GE(AF^M)) \vdash_0 \varphi$ it holds that $\varphi \in \mathsf{Conc}(GE(AF^M))$
- $\mathsf{RP}_{\mathsf{IC}}$ (*indirect consistency*) iff $\mathsf{Conc}(GE(AF^M)) \nvdash_0 \bot$

The following propositions establish sufficient conditions for $\mathsf{RP}_{\mathsf{SUB}}$ (resp. $\mathsf{RP}_{\mathsf{DC}}$) to be satisfied by an associated AF:

Proposition 3. *Let (M,w) be a pointed model, where $M = (W, \mathscr{B}, \mathscr{O}, \mathscr{D}, \mathfrak{n}, ||\cdot||)$. If $\mathscr{O}$ is closed under subarguments (i.e. $\alpha \in \mathscr{O}$ implies* $\mathsf{sub}_{\mathsf{A}}(\alpha) \subseteq \mathscr{O}$*), then $GE(AF^M)$ is closed under subarguments.*

[8] We lift the domain of the function Conc from arguments to sets of arguments as follows: $\mathsf{Conc}(\mathscr{S}) := \{\mathsf{Conc}(\alpha) \mid \alpha \in \mathscr{S}\}$ for any $\mathscr{S} \subseteq \mathscr{A}$.

Proposition 4. *Let (M,w) be a pointed model, then AF^M satisfies* direct consistency.

Remark. *In the current setting, it is crucial for Proposition 4 to hold that we allow completely unrestricted rebuttals (see Definition 6 and the subsequent discussion).*

$\mathsf{RP}_{\mathsf{CL}}$ and $\mathsf{RP}_{\mathsf{IC}}$ are violated by the current framework. Let us show why this happens and why it is not an unavoidable inconvenience for our purposes. First of all, note that $\mathsf{RP}_{\mathsf{CL}}$ cannot be satisfied by *any* associated AF. Note that $\mathsf{Conc}(GE(AF^M)) \vdash_0 \{\varphi \in \mathscr{L}_{\mathsf{BA}} \mid \vdash_0 \varphi\}$ for any model M. Therefore, for $\mathsf{RP}_{\mathsf{CL}}$ to be true, it should hold that $\{\varphi \in \mathscr{L}_{\mathsf{BA}} \mid \vdash_0 \varphi\} \subseteq \mathsf{Conc}(GE(AF^M))$. But this is impossible since $\{\varphi \in \mathscr{L}_{\mathsf{BA}} \mid \vdash_0 \varphi\}$ is infinite and $\mathsf{Conc}(GE(AF^M))$ is finite by assumption (because awareness sets are finite by assumption). Nevertheless, $\mathsf{RP}_{\mathsf{CL}}$ is just a special case of logical omniscience (propositional logical omniscience); so its satisfiability should not be pursued when modelling resource-bounded agents. As pointed out in [3], this problem can be avoided using query-based implementations for computing the grounded extension. This strategy does not seem appropriate in the current context, since it still would require to generate the whole set of well-shaped arguments.

As for $\mathsf{RP}_{\mathsf{IC}}$, its failure is more threatening. Moreover, our agent fails to have the following forms of consistency (that fall between direct and indirect consistency): (i) there is no $\varphi \in \mathsf{Conc}(GE(AF^M))$ such that $\{\varphi\} \vdash_0 \bot$ and (ii) there are no $\varphi, \psi \in \mathsf{Conc}(GE(AF^M))$ such that $\{\varphi, \psi\} \vdash_0 \bot$. These facts revel the minimal character of our formalism. Note however, that the first case can be avoided by closing $\mathscr{O}$ under conclusions and single negations. The second case can in turn be overcome by defining $\sim\varphi := \{\psi \mid \{\varphi, \psi\} \vdash_0 \bot\}$. Be as it may, failure of different forms of consistency are understood as pitfalls in many different contexts. However, at the same time, it seems plausible to claim that reasonable (yet not fully rational) agents can have indirectly inconsistent AB-beliefs; as far as they keep their AB-beliefs being directly consistent (see e.g. [24, §2] for a defence of this kind of inconsistencies). Note that although AB-beliefs might be indirectly inconsistent, they are not trivial (agents never end up believing *everything*). Moreover, if one wants to strengthen the reasoning skills of the modelled agent, two interesting questions arise. First, is there any set of sufficient conditions that guarantees the satisfaction of $\mathsf{RP}_{\mathsf{IC}}$ in $\mathscr{L}_{\mathsf{BA}}$ while keeping awareness sets finite? A positive answer might not be trivial, since the satisfaction of $\mathsf{RP}_{\mathsf{IC}}$ is usually proved as a corollary of $\mathsf{RP}_{\mathsf{DC}}$ and $\mathsf{RP}_{\mathsf{CL}}$ [3,4]. Second, given an indirectly inconsistent associated AF, is there an action (or sequence of actions) such that indirect consistency is recovered?

5. Future Work

Besides the open problems mentioned in the last section, there are several questions that require further study. We highlight the following ones. First, examining $\mathscr{L}_{\mathsf{BA}}$ on the view of additional postulates (see [25]). Second, it would also be interesting to study whether it is possible to characterize axiomatically the behaviour of B^{AB}, when treated as a primitive operator.

Acknowledgements We thank the anonymous reviewers for valuable and insightful comments, some of which had to be left out due to lack of space unfortunately. The research activity of A. Yuste-Ginel is supported by the predoctoral grant MECD-FPU 2016/04113.

References

[1] Hasan A, Fumerton R. Foundationalist Theories of Epistemic Justification. In: Zalta EN, editor. The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University; 2018.

[2] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321–357.

[3] Caminada M, Amgoud L. On the evaluation of argumentation formalisms. Artificial Intelligence. 2007;171(5-6):286–310.

[4] Modgil S, Prakken H. A general account of argumentation with preferences. Artificial Intelligence. 2013;195:361–397.

[5] Artemov S. The logic of justification. The Review of Symbolic Logic. 2008;1(4):477–513.

[6] Baltag A, Renne B, Smets S. The Logic of Justified Belief Change, Soft Evidence and Defeasible Knowledge. In: Ong L, de Queiroz R, editors. Logic, Language, Information and Computation. WoLLIC 2012. LNCS. vol. 7456. Springer; 2012. p. 168–190.

[7] Baltag A, Renne B, Smets S. The logic of justified belief, explicit knowledge, and conclusive evidence. Annals of Pure and Applied Logic. 2014;165(1):49–81.

[8] van Benthem J, Pacuit E. Dynamic logics of evidence-based beliefs. Studia Logica. 2011;99(1-3):61.

[9] Baltag A, Bezhanishvili N, Özgün A, Smets S. Justified Belief and the Topology of Evidence. In: Väänänen J, Hirvonen Å, de Queiroz R, editors. Logic, Language, Information, and Computation. Springer; 2016. p. 83–103.

[10] Grossi D, van der Hoek W. Justified Beliefs by Justified Arguments. In: Baral C, Giacomo GD, Eiter T, editors. Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference. AAAI Press; 2014.

[11] Shi C, Smets S, Velázquez-Quesada FR. Beliefs supported by binary arguments. Journal of Applied Non-Classical Logics. 2018;28:2-3:165–188.

[12] Beirlaen M, Heyninck J, Pardo P, Straßer C. Argument strength in formal argumentation. IfCoLog Journal of Logics and their Applications. 2018;5(3):629–675.

[13] D'Agostino M, Modgil S. A Rational Account of Classical Logic Argumentation for Real-World Agents. In: Kaminka G, et al., editors. Proceedings of the Twenty-Second European Conference on Artificial Intelligence. ECAI'16. IOS Press; 2016. p. 141–149.

[14] Hintikka J. Knowledge and belief: an introduction to the logic of the two notions. Cornell University Press; 1962.

[15] Fagin R, Halpern JY, Moses Y, Vardi M. Reasoning about knowledge. MIT press; 2004.

[16] Meyer JJC, van der Hoek W. Epistemic logic for AI and computer science. vol. 41. Cambridge University Press; 1995.

[17] Fagin R, Halpern JY. Belief, awareness, and limited reasoning. Artificial intelligence. 1987;34(1):39–76.

[18] Blackburn P, De Rijke M, Venema Y. Modal Logic. Cambridge University Press; 2002.

[19] Burrieza A, Yuste-Ginel A. Argument evaluation in multi-agent justification logics. Logic Journal of the IGPL. 2019;DOI:10.1093/jigpal/jzz046.

[20] Caminada MWA, Modgil S, Oren N. Preferences and Unrestricted Rebut. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Computational Models of Argument: Proceedings of COMMA 2014. IOS Press; 2014. p. 209–220.

[21] Heyninck J, Straßer C. Revisiting Unrestricted Rebut and Preferences in Structured Argumentation. In: Sierra C, editor. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17; 2017. p. 1088–1092.

[22] Baroni P, Caminada M, Giacomin M. Abstract argumentation frameworks and their semantics. In: Baroni P, Gabbay DM, Giacomin M, editors. Handbook of formal argumentation. London: College Publications; 2018. p. 159–236.

[23] Caminada M. On the issue of reinstatement in argumentation. In: Fisher M, van der Hoek W, Konev B, Lisitsa A, editors. Logics in Artificial Intelligence. JELIA 2006. vol. 4160 of LNCS. Springer; 2006. p. 111–123.

[24] Parikh R. Sentences, belief and logical omniscience, or what does deduction tell us? The Review of Symbolic Logic. 2008;1(4):459–476.

[25] Caminada M. Rationality Postulates: applying argumentation theory for non-monotonic reasoning. Journal of Applied Logics. 2017;4(8):2707–2734.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200499

Minimal Strong Admissibility: A Complexity Analysis

Martin CAMINADA [a] and Paul E. DUNNE [b]
[a] *Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 2LQ, UK*
(CaminadaM@cardiff.ac.uk)
[b] *University of Liverpool, Ashton Building, Liverpool L69 7ZF, UK*
(P.E.Dunne@liverpool.ac.uk)

Abstract. The concept of strong admissibility plays an important role in some of the dialectical proof procedures that have been stated for grounded semantics. As the grounded extension is the (unique) biggest strongly admissible set, to show that an argument is in the grounded extension it suffices to show that it is in a strongly admissible set. We are interested in identifying a strongly admissible set that minimizes the number of steps needed in the associated dialectical proof procedure. In the current work, we look at the computational complexity of doing so.

Keywords. strong admissibility, computational complexity, explainable AI

1. Introduction

The concept of strong admissibility was first introduced in the work of Baroni and Giacomin [1] and has subsequently been studied by Caminada and Dunne [6,4]. Strong admissibility is particularly useful for showing that a particular argument is part of the grounded extension. As the grounded extension is the (unique) biggest strongly admissible set, showing membership of any strongly admissible set is sufficient to prove that the argument is in the grounded extension.

Alternatively, one could apply the concept of a strongly admissible labelling [6,4]. As the grounded labelling is the (unique) biggest strongly admissible labelling,[1] showing that an argument is labelled `in` by any strongly admissible labelling is sufficient to prove that the argument is labelled `in` by the grounded labelling (and therefore is an element of the grounded extension [5,11]).

As an argument can be labelled `in` by more than one strongly admissible labelling (or be an element of more than one strongly admissible set) the question then becomes which particular strongly admissible labelling to show in order to prove membership of the grounded extension. Although in principle any strongly admissible labelling that labels the argument `in` will do, it can have advantages to select a strongly admissible labelling that is *minimal*, especially when the aim is explainability.

[1] Biggest w.r.t. "$\sqsubseteq$" [12,4].

The concept of a strongly admissible labelling matters because it is at the basis of some of the proof procedures for grounded semantics [4], in particular of the Grounded Discussion Game [7]. The Grounded Discussion Game is a dialectical proof procedure with two players: the proponent and the opponent. The game is such that an argument is in the grounded extension iff it is possible for the proponent to win the discussion. The idea is that this discussion can serve as an explanation of why a particular argument should be accepted as being in the grounded extension. In such a case, the computer will assume the role of proponent and a human user will assume the role of opponent [3]. If an argument is in the grounded extension, the proponent can win the discussion by using a strongly admissible labelling as a roadmap [6]. In order to minimize the number of discussion steps (and hence save the user's time during the discussion) the strongly admissible labelling that is to be applied as a roadmap should have a minimal size[2] among all strongly admissible labellings that label the argument `in`. In the current paper we examine the computational complexity of verifying such a minimal strongly admissible labelling. In addition, we study the computational complexity of determining whether there is a strongly admissible labelling that labels a particular argument `in` and has a size of at most k.

This paper is structured as follows. First, in Section 2 we present some formal preliminaries regarding abstract argumentation and strong admissibility. Then, in Section 3 we present some results regarding the computational complexity of identifying strongly admissible labellings with bounded or minimal size. We round off in Section 4 with a discussion of the obtained results.

2. Preliminaries

For current purposes, we restrict ourselves to finite argumentation frameworks.

Definition 1. *An* argumentation framework *is a pair* (Ar, att) *where* Ar *is a finite set of entities, called arguments, whose internal structure can be left unspecified, and att is a binary relation on* Ar*. For any* $A, B \in Ar$ *we say that* A attacks B *iff* $(A, B) \in att$*.*

Definition 2. *Let* (Ar, att) *be an argumentation framework,* $A \in Ar$ *and* $\mathcal{A}rgs \subseteq Ar$*. We define* A^+ *as* $\{B \in Ar \mid A \text{ attacks } B\}$*,* A^- *as* $\{B \in Ar \mid B \text{ attacks } A\}$*,* $\mathcal{A}rgs^+$ *as* $\cup\{A^+ \mid A \in \mathcal{A}rgs\}$*, and* $\mathcal{A}rgs^-$ *as* $\cup\{A^- \mid A \in \mathcal{A}rgs\}$*.* $\mathcal{A}rgs$ *is said to be* conflict-free *iff* $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$*.* $\mathcal{A}rgs$ *is said to* defend A *iff* $A^- \subseteq \mathcal{A}rgs^+$*. The characteristic function* $F : 2^{Ar} \rightarrow 2^{Ar}$ *is defined as* $F(\mathcal{A}rgs) = \{A \mid \mathcal{A}rgs \text{ defends } A\}$*.*

Definition 3. *Let* (Ar, att) *be an argumentation framework.* $\mathcal{A}rgs \subseteq Ar$ *is*

- *an admissible set iff* $\mathcal{A}rgs$ *is conflict-free and* $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$
- *a complete extension iff* $\mathcal{A}rgs$ *is conflict-free and* $\mathcal{A}rgs = F(\mathcal{A}rgs)$
- *a grounded extension iff* $\mathcal{A}rgs$ *is the smallest (w.r.t.* $\subseteq$*) complete extension*
- *a preferred extension iff* $\mathcal{A}rgs$ *is a maximal (w.r.t.* $\subseteq$*) complete extension*

[2] We recall that the size of a labelling $\mathcal{L}ab$ is $|\mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab)|$.

The concept of strong admissibility was introduced by Baroni and Giacomin [1]. For current purposes we will apply the equivalent definition of Caminada [6,4].

Definition 4. *Let (Ar, att) be an argumentation framework. $\mathcal{A}rgs \subseteq Ar$ is* strongly admissible *iff every $A \in \mathcal{A}rgs$ is defended by some $\mathcal{A}rgs' \subseteq \mathcal{A}rgs \setminus \{A\}$ which in its turn is again strongly admissible.*

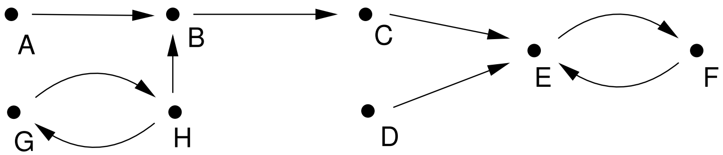


Figure 1. An example of an argumentation framework.

As an example (taken from [4]), in the argumentation framework of Figure 1 the strongly admissible sets are $\emptyset$, $\{A\}$, $\{A, C\}$, $\{A, C, F\}$, $\{D\}$, $\{A, D\}$, $\{A, C, D\}$, $\{D, F\}$, $\{A, D, F\}$ and $\{A, C, D, F\}$, the latter also being the grounded extension. The set $\{A, C, F\}$ is strongly admissible as A is defended by $\emptyset$, C is defended by $\{A\}$ and F is defended by $\{A, C\}$, each of which is a strongly admissible subset of $\{A, C, F\}$ not containing the argument it defends. Please notice that although the set $\{A, F\}$ defends argument C in $\{A, C, F\}$, it is in its turn not strongly admissible (unlike $\{A\}$). Hence the requirement in Definition 4 for $\mathcal{A}rgs'$ to be a *subset* of $\mathcal{A}rgs \setminus \{A\}$. We also observe that although $\{C, H\}$ is an admissible set, it is not a *strongly* admissible set, since no subset of $\{C, H\} \setminus \{H\}$ defends H.

It can be shown that each strongly admissible set is conflict-free and admissible [4]. The strongly admissible sets form a lattice, of which the empty set is the bottom element and the grounded extension is the top element [4].

The above definitions essentially follow the extension based approach as described in [13]. It is also possible to define the key argumentation concepts in terms of argument labellings [5,11].

Definition 5. *Let (Ar, att) be an argumentation framework. An* argument labelling *is a function $\mathcal{L}ab : Ar \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. An argument labelling is called an* admissible labelling *iff for each $A \in Ar$ it holds that:*

- *if $\mathcal{L}ab(A) = \texttt{in}$ then for each B that attacks A it holds that $\mathcal{L}ab(B) = \texttt{out}$*
- *if $\mathcal{L}ab(A) = \texttt{out}$ then there exists a B that attacks A such that $\mathcal{L}ab(B) = \texttt{in}$*

$\mathcal{L}ab$ is called a complete labelling *iff it is an admissible labelling and for each $A \in Ar$ it also holds that:*

- *if $\mathcal{L}ab(A) = \texttt{undec}$ then there is a B that attacks A such that $\mathcal{L}ab(B) = \texttt{undec}$, and for each B that attacks A such that $\mathcal{L}ab(B) \neq \texttt{undec}$ it holds that $\mathcal{L}ab(B) = \texttt{out}$*

As a labelling is essentially a function, we sometimes write it as a set of pairs. Also, if $\mathcal{L}ab$ is a labelling, we write $\texttt{in}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{out}\}$ and $\texttt{undec}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{undec}\}$. As a labelling is also a partition of the arguments into sets of $\mathtt{in}$-labelled arguments, $\mathtt{out}$-labelled arguments and $\mathtt{undec}$-labelled arguments, we sometimes write it as a triplet $(\mathtt{in}(\mathcal{L}ab), \mathtt{out}(\mathcal{L}ab), \mathtt{undec}(\mathcal{L}ab))$.

Definition 6 ([12])**.** *Let $\mathcal{L}ab$ and $\mathcal{L}ab'$ be argument labellings of argumentation framework (Ar, att). We say that $\mathcal{L}ab \sqsubseteq \mathcal{L}ab'$ iff $\mathtt{in}(\mathcal{L}ab) \subseteq \mathtt{in}(\mathcal{L}ab')$ and $\mathtt{out}(\mathcal{L}ab) \subseteq \mathtt{out}(\mathcal{L}ab')$.*

Definition 7. *Let $\mathcal{L}ab$ be a complete labelling of argumentation framework (Ar, att). $\mathcal{L}ab$ is said to be*

- *a grounded labelling iff $\mathcal{L}ab$ is the (unique) smallest (w.r.t. $\sqsubseteq$) complete labelling*
- *a preferred labelling iff $\mathcal{L}ab$ is a maximal (w.r.t. $\sqsubseteq$) complete labelling*

The next step is to define a strongly admissible labelling. In order to do so, we first need to introduce the concept of a min-max numbering [4].

Definition 8. *Let $\mathcal{L}ab$ be an admissible labelling of argumentation framework (Ar, att). A* min-max numbering *is a total function $\mathcal{MM}_{\mathcal{L}ab} : \mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab) \to \mathbb{N} \cup \{\infty\}$ such that for each $A \in \mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab)$ it holds that:*

- *if $\mathcal{L}ab(A) = \mathtt{in}$ then $\mathcal{MM}_{\mathcal{L}ab}(A) = max(\{\mathcal{MM}_{\mathcal{L}ab}(B) \mid B$ attacks A and $\mathcal{L}ab(B) = \mathtt{out}\}) + 1$ (with $max(\emptyset)$ defined as 0)*
- *if $\mathcal{L}ab(A) = \mathtt{out}$ then $\mathcal{MM}_{\mathcal{L}ab}(A) = min(\{\mathcal{MM}_{\mathcal{L}ab}(B) \mid B$ attacks A and $\mathcal{L}ab(B) = \mathtt{in}\}) + 1$ (with $min(\emptyset)$ defined as ∞)*

It has been proved that every admissible labelling has a unique min-max numbering [4]. A strongly admissible labelling can then be defined as follows [4].

Definition 9. *A* strongly admissible labelling *is an admissible labelling whose min-max numbering yields natural numbers only (so no argument is numbered ∞).*

As an example (taken from [4]), consider again the argumentation framework of Figure 1. Here, the admissible labelling $\mathcal{L}ab_1 = (\{A, C, F, G\}, \{B, E, H\}, \{D\})$ has min-max numbering $\{(A : 1), (B : 2), (C : 3), (E : 4), (F : 5), (G : \infty), (H : \infty)\}$, which means that it is not strongly admissible. The admissible labelling $\mathcal{L}ab_2 = (\{A, C, D, F\}, \{B, E\}, \{G, H\})$ has min-max numbering $\{(A : 1), (B : 2), (C : 3), (D : 1), (E : 2), (F : 3)\}$, which means that it is strongly admissible.

The strongly admissible labellings also form a lattice, of which the all-$\mathtt{undec}$ labelling is the bottom element and the grounded labelling is the top element [4].

A strongly admissible set is at the basis of the Grounded Discussion Game [7], which is a sound and complete dialectical proof procedure for proving that an argument is in the grounded extension. The game is played by two parties, called the proponent and the opponent, who each utter moves that contain arguments. The proponent starts by uttering what is called the main argument.The rules of the game are such that the main argument is in the grounded extension iff the proponent has a winning strategy for the game. The proponent is able to play such a winning strategy by basing his moves on a strongly admissible labelling and its associated min-max numbering. As the main argument can be labelled $\mathtt{in}$

by several strongly admissible labellings, this raises the question of which strongly admissible labelling to choose. If the aim is to use the Grounded Discussion Game for purposes of explanation and human-computer interaction (as is suggested in [8]) one would like to choose a strongly admissible labelling that minimizes the required number of steps in the associated discussion. It has been observed [7] that such a strongly admissible labelling $\mathcal{L}ab$ should have a minimal size (that is, $|\mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab)|$ should be minimal) among all strongly admissible labellings that label the main argument $\mathtt{in}$.

3. Computational Complexity

We will, generally, exploit the criteria specified in Definition 9 in order to validate that the labellings in the constructions are, indeed, strongly admissible labellings.

Formally, the bounded labelling problem is given as:

BOUNDED STRONG ADMISSIBLE LABELLING (BSAL)
Instance: An AF, $\mathcal{H} = (Ar, att)$, an argument $x \in Ar$ and a positive integer $k \in \mathbb{N}$.
Question: Is there a strongly admissible labelling, $\mathcal{L}ab$, of Ar for which

$$\mathcal{L}ab(x) = \mathtt{in} \text{ and } |\{\ y\ :\ \mathcal{L}ab(y) = \mathtt{in}\ \} \cup \{\ y\ :\ \mathcal{L}ab(y) = \mathtt{out}\}|\ \leq\ k\ ?$$

Theorem 1. BSAL *is* NP*–complete.*

Proof. We first note that BSAL $\in$ NP by virtue of the fact that for any strongly admissible labelling

$$\mathcal{L}ab\ :\ Ar \rightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$$

its correctness may be checked in polynomial time (cf. [4]).

In order to show that BSAL is NP–hard we use a reduction from the well-known NP–complete problem of CNF satisfiability (CNF-SAT).

Given $\varphi(Z)$ a CNF formula over the propositional variables $Z = \{z_1, \ldots, z_n\}$ and having m clauses, $\{C_1, C_2 \ldots, C_m\}$ we form the AF, $\mathcal{H}_\varphi(Ar_\varphi, att_\varphi)$ with $|Ar_\varphi| = 3n + m + 1$ and arguments named

φ	
C_j	For each clause C_j and $1 \leq j \leq m$
D_i	For each variable z_i in Z
z_i	For each variable z_i in Z
$\neg z_i$	For each variable z_i in Z

The attacks in att_φ are

$< C_j, \varphi >$	For each $1 \leq j \leq m$
$< D_i, \varphi >$	For each $1 \leq i \leq n$
$< z_i, D_i >$	For each $1 \leq i \leq n$
$< \neg z_i, D_i >$	For each $1 \leq i \leq n$
$< z_i, C_j >$	if z_i is a literal in clause C_j of $\varphi(Z)$
$< \neg z_i, C_j >$	if $\neg z_i$ is a literal in clause C_j of $\varphi(Z)$

The instance of BSAL is formed as $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$. Notice that as $\mathcal{H}_{varphi}$ is an acyclic AF, each admissible labelling is a strongly admissible labelling [9] and vice versa. We therefore only need to prove admissibility in order to show strong admissibility.

We claim this instance is accepted if and only if $\varphi(Z)$ is satisfiable.

First notice that **any** admissible labelling of $\mathcal{H}_\varphi$ in which $\mathcal{L}ab(\varphi) = \texttt{in}$ must be such that $|\{x\ :\ \mathcal{L}ab(x) = \texttt{undec}\}| \leq n$. In other words a labelling with minimal size must fix the status of at least $1+m+2n$ arguments. If $\mathcal{L}ab(\varphi) = \texttt{in}$ then we must have $\mathcal{L}ab(C_j) = \texttt{out}$ for every $1 \leq j \leq m$ and $\mathcal{L}ab(D_i) = \texttt{out}$ for every $1 \leq i \leq n$. In order to ensure the second of these we must have at least one of $\mathcal{L}ab(z_i) = \texttt{in}$ or $\mathcal{L}ab(\neg z_i) = \texttt{in}$. In total any strongly admissible labelling with $\mathcal{L}ab(\varphi) = \texttt{in}$ commits at least $1+m+2n$ arguments to a definite status (in or out).

Now suppose that $\varphi(Z)$ is satisfiable using some setting $\alpha = (a_1, a_2, \ldots, a_n)$ of its propositional variables. Choose the labelling, $\mathcal{L}ab_\alpha$ of Ar_φ for which

$$\mathcal{L}ab_\alpha(x) \quad = \quad = \begin{cases} \texttt{in} & \text{if } x = \varphi \\ \texttt{out} & \text{if } x \in \{C_1, \ldots, C_m\} \\ \texttt{out} & \text{if } x \in \{D_1, \ldots, D_n\} \\ \texttt{in} & \text{if } x = z_i \text{ and } a_i = \textbf{true} \\ \texttt{undec} & \text{if } x = z_i \text{ and } a_i = \textbf{false} \\ \texttt{in} & \text{if } x = \neg z_i \text{ and } a_i = \textbf{false} \\ \texttt{undec} & \text{if } x = \neg z_i \text{ and } a_i = \textbf{true} \end{cases}$$

It is not hard to see that this labelling satisfies the requirements needed to be an strongly admissible labelling: each $\{z_i, \neg z_i\}$ is unattacked and may be labelled as either undec or in; every D_i argument is correctly labelled out since it is attacked by an argument labelled in (i.e. z_i or $\neg z_i$); every C_j argument is, also, correctly labelled out as, since the labelling of $\{z_i,\ \neg z_i\ :\ 1 \leq i \leq n\}$ is determined by α (with $\varphi(\alpha) = \textbf{true}$) it follows that every C_j is attacked by some argument labelled in (since, in order for $\varphi(\alpha)$ to be **true**, every clause C_j must contain a literal which evaluates to **true** under α). Finally there are exactly n (the minimum possible) arguments labelled undec.

We conclude that if $\varphi(Z)$ is satisfiable then $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL.

For the converse argument, suppose $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL. Let $\mathcal{L}ab$ be the labelling of Ar_φ which witnesses this. That is to say, $\mathcal{L}ab(\varphi) = \texttt{in}$ and $|\{x\ :\ \mathcal{L}ab(x) \neq \texttt{undec}\}| = 1+m+2n$.

As we argued previously, from the fact that $\mathcal{L}ab(\varphi) = \texttt{in}$ we must have an additional $m+n$ arguments whose status is committed to being out: namely the $n+m$ clause arguments $\{\ C_j\ :\ 1 \leq j \leq m\} \cup \{\ D_i\ :\ 1 \leq i \leq n\}$. Furthermore for the labelling correctly to ensure $\mathcal{L}ab(D_i) = \texttt{out}$ we need either $\mathcal{L}ab(z_i) = \texttt{in}$ or $\mathcal{L}ab(\neg z_i) = \texttt{in}$. Now since we have assumed that $\mathcal{L}ab$ commits the status of at most $1+m+2n$ arguments and we have already determined how $1+m+2n$ must be set it must be the case that *exactly* one of $\{\ z_i, \neg z_i\}$ is set to in and the other to undec. Consider the setting, $\alpha_{\mathcal{L}ab}$ of the propositional variables:

$$\alpha_{\mathcal{L}ab}(z_i) \quad = \quad \begin{cases} \textbf{true} & \text{if } \mathcal{L}ab(z_i) = \texttt{in} \\ \textbf{false} & \text{if } \mathcal{L}ab(\neg z_i) = \texttt{in} \end{cases}$$

This assignment must satisfy $\varphi(Z)$: every clause argument, C_j, is correctly labelled `out` by $\mathcal{L}ab$ and, therefore, must be attacked by some z_i or $\neg z_i$ labelled `in`. In the assignment $\alpha_{\mathcal{L}ab}$ just described the corresponding setting of z_i as **true** ($\mathcal{L}ab(z_i) = \texttt{in}$) or **false** ($\mathcal{L}ab(\neg z_i) = \texttt{in}$) will lead to the clause C_j taking the value **true**.

We deduce that if $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL then $\varphi(Z)$ is accepted as an instance of CNF-SAT. □

The decision problem BSAL is in essence an *existence* question: can we find a suitable labelling that commits the status of *at most* some number of arguments? A related question is that of *verifying* that a given labelling is indeed minimal. Formally this is the verification problem, MSAL:

Minimal strong admissible labelling (MSAL)
Instance: An AF, $\mathcal{H} = (Ar, att)$, an argument $x \in Ar$ and a strongly admissible labelling, $\mathcal{L}ab$ of Ar with which $\mathcal{L}ab(x) = \texttt{in}$.
Question: Does $\mathcal{L}ab$ have a minimal size? i.e. for any strongly admissible labelling, $\mathcal{L}ab'$, of Ar with $\mathcal{L}ab'(x) = \texttt{in}$, $|\{y :\ \mathcal{L}ab'(y) \neq \texttt{undec}\} \geq |\{y :\ \mathcal{L}ab(y) \neq \texttt{undec}\}$?

Theorem 2. MSAL *is co*NP*–complete.*

Proof. First notice that MSAL $\in$ coNP. Simply check every labelling $\mathcal{L}ab'$ of Ar and for any which describe a strongly admissible labelling with $\mathcal{L}ab'(x) = \texttt{in}$ confirm that $|\{y : \mathcal{L}ab'(y) \neq \texttt{undec}\}| \geq |\{y : \mathcal{L}ab(y) \neq \texttt{undec}\}$. This entire computation may be realized in coNP.

For coNP–hardness we use a reduction from CNF-UNSAT.

A key point in this reduction are that the instances of CNF-UNSAT are restricted to those having n propositional variables *and* exactly $m = 4n-1$ clauses.[3]

Given $\varphi(Z)$ a propositional formula over n variables and $4n-1$ clauses $\{C_1, \ldots, C_{4n-1}\}$ the AF, $\mathcal{G}_\varphi$ consists of two parts:

1. The AF, $\mathcal{H}_\varphi$ from the proof of Theorem 1. Notice that this contains exactly $7n$ arguments: the literals $\{\ z_i,\ \neg z_i\ :\ 1 \leq i \leq n\}$; $4n-1$ clauses $\{C_j\ :\ 1 \leq j \leq 4n-1\}$; n clauses $\{D_i\ :\ 1 \leq i \leq n\}$ and φ.
2. The second section also uses the literal $(z_i,\ \neg z_i)$ arguments from $\mathcal{H}_\varphi$ and an additional $4n+1$ arguments:

$$\begin{array}{cc} \{\ b_i,\ \neg b_i\ :\ 1 \leq i \leq n\} & \{\ c_i\ :\ 1 \leq i \leq n\} \\ \{\ g_i\ :\ 1 \leq i \leq n\} & \pi \end{array}$$

In order to combine these structures two further arguments are introduced: ψ whose only attackers are φ and π; and θ whose only attacker is ψ.

The AF is completed by adding to those already in $\mathcal{H}_\varphi$ and the three attacks

[3] A standard "padding" argument such as that from [15, Thm. 2] easily shows this variant remains coNP–complete.

$$\{<\varphi,\psi>,<\pi,\psi>,<\psi,\theta>\}$$

the new attacks:

$$\begin{array}{lcl}
\{<z_i,b_i> \ : \ 1\le i\le n\} & \quad & \{<\neg z_i,\neg b_i> \ : \ 1\le i\le n\} \\
\{<b_i,c_i> \ : \ 1\le i\le n\} & & \{<\neg b_i,c_i> \ : \ 1\le i\le n\} \\
\{<c_i,g_i> \ : \ 1\le i\le n\} & & \{<g_i,\pi> \ : \ 1\le i\le n\}
\end{array}$$

$\mathcal{G}_\varphi$ is illustrated in Figure 2. As $\mathcal{G}_\varphi$ is acyclic, it suffices to prove admissibility in order to show strong admissibility [9].

The labelling, $\mathcal{L}ab$, of which the minimal size is to be checked uses

$$\mathcal{L}ab(x) \quad = \quad \begin{cases}
\texttt{in} & \text{if } x\in\{z_i,\ \neg z_i \ : \ 1\le i\le n\} \\
\texttt{out} & \text{if } x\in\{b_i,\ \neg b_i \ : \ 1\le i\le n\} \\
\texttt{in} & \text{if } x\in\{c_i \ : \ 1\le i\le n\} \\
\texttt{out} & \text{if } x\in\{g_i \ : \ 1\le i\le n\} \\
\texttt{in} & \text{if } x=\pi \\
\texttt{out} & \text{if } x=\psi \\
\texttt{in} & \text{if } x=\theta \\
\texttt{undec} & \text{otherwise}
\end{cases}$$

We claim that $<\mathcal{G}_\varphi,\theta,\mathcal{L}ab>$ is accepted as an instance of MSAL if and only if $\varphi(Z_n)$ is unsatisfiable.

Suppose that $\varphi(Z)$ is in fact satisfiable using an assignment of propositional values $(a_1,\ldots,a_n)$. Notice that $\mathcal{L}ab$ has exactly $m+n+1$ arguments labelled `undec` which given the conditions on m evaluates to $5n$. Consider the alternative labelling, $\mathcal{L}ab'$, in which

$$\mathcal{L}ab'(x) \quad = \quad \begin{cases}
\texttt{in} & \text{if } \ x=z_i \text{ and } a_i=\textbf{true} \\
\texttt{in} & \text{if } \ x=\neg z_i \text{ and } a_i=\textbf{false} \\
\texttt{undec} & \text{if } \ x=z_i \text{ and } a_i=\textbf{false} \\
\texttt{undec} & \text{if } \ x=\neg z_i \text{ and } a_i=\textbf{true} \\
\texttt{undec} & \text{if } \ x\in\{b_i,\ \neg b_i \ : \ 1\le i\le n\} \\
\texttt{undec} & \text{if } \ x\in\{c_i \ : \ 1\le i\le n\} \\
\texttt{undec} & \text{if } \ x\in\{g_i \ : \ 1\le i\le n\} \\
\texttt{undec} & \text{if } \ x=\pi \\
\texttt{out} & \text{if } \ x\in\{D_i \ : \ 1\le i\le n\} \\
\texttt{out} & \text{if } \ x\in\{C_j \ : \ 1\le j\le 4n-1\} \\
\texttt{in} & \text{if } \ x=\varphi \\
\texttt{out} & \text{if } \ x=\psi \\
\texttt{in} & \text{if } \ x=\theta
\end{cases}$$

The labelling, $\mathcal{L}ab'$ is easily checked to be a valid admissible labelling by virtue of the fact that $(a_1,\ldots,a_n)$ satisfies $\varphi(Z_n)$ every clause argument, C_j, can be labelled `out` since it is attacked by (at least one) z_i or $\neg z_i$ labelled `in`. Similarly each D_i is attacked by z_i labelled `in` or $\neg z_i$ labelled `in`. Finally since φ is attacked only by arguments labelled `out` it may be labelled `in` leading to $\mathcal{L}ab'(\psi)=$ `out` (the other attacker of ψ being `undec`) and $\mathcal{L}ab'(\theta)=$ `in`. The number of `undec`

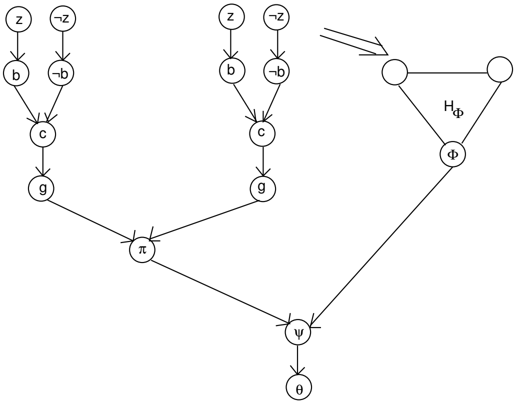


Figure 2. Construction of AF used to show MSAL is coNP–hard.

arguments is, however, more than those in $\mathcal{L}ab$ since $\mathcal{L}ab'$ labels n arguments (from $\{z_i,\ \neg z_i\}$) as `undec`, the $2n$ arguments in $\{\ b_i, \neg b_i\}$, the n arguments in $\{c_i : 1 \leq i \leq n\}$ and the n arguments in $\{g_i : 1 \leq i \leq n\}$. Finally π is also labelled `undec`. In total this gives $n + 2n + n + n + 1 \ = \ 5n + 1$ so that,

$$|\{\ y\ :\ \mathcal{L}ab'(y) = \texttt{undec}\}| \ = \ 5n + 1 \ \ > \ \ 5n \ \ = |\{\ y\ :\ \mathcal{L}ab(y) = \texttt{undec}\}|$$

and the conclusion that if $\varphi(Z_n)$ is not accepted as an instance of CNF–UNSAT then $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is not accepted as an instance of MSAL.

For the converse implication, suppose that $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is rejected as an instance of MSAL.

In order for this to be the case we must have some admissible labelling, $\mathcal{L}ab'$, of $\mathcal{G}_\varphi$ in which $\mathcal{L}ab'(\theta) = \texttt{in}$ and $|\{y\ :\ \mathcal{L}ab'(y) = \texttt{undec}\}| > |\{y\ :\ \mathcal{L}ab(y) = \texttt{undec}\}$

It is not hard to see that any such labelling must use $\mathcal{L}ab'(\pi) = \texttt{undec}$ and $\mathcal{L}ab'(\varphi) = \texttt{in}$: in $\mathcal{L}ab$ every C_j and D_i argument together with φ are already `undec`; in order to ensure ψ can properly be labelled `out` at least one of π or φ must be labelled `in`. In order, however, properly to label π as `in` the status of **every** $\{z_i,\ \neg z_i\}$ has to be fixed.

Now in order for $\mathcal{L}ab'$ properly to label φ as `in` there are are two possibilities arising from the way in which $\{D_i\ :\ 1 \leq i \leq n\}$ may properly be labelled `out`.

Case 1: $\mathcal{L}ab'$ properly labels all C_j arguments as `out` through a labelling of $\{z_i, \neg z_i\}$ in which (at least) one z_i has $\mathcal{L}ab'(z_i) = \mathcal{L}ab'(\neg z_i) = \texttt{in}$.

Since at least one argument from each pair $\{z_i, \neg z_i\}$ must be committed to be `in` (in order properly to label D_i as `out`) a labelling, $\mathcal{L}ab'$ meeting the criteria in Case 1 contributes $n - 1$ `undec` (from $\{z_i,\ \neg z_i\}$); $2n$ (from $\{\ b_i, \neg b_i\}$); a further n ($\{c_i : 1 \leq i \leq n\}$); n more (from $\{g_i : 1 \leq i \leq n\}$) and the argument π. In total

$$(n - 1)\ +\ 2n\ + n\ +\ n\ +\ 1\ \ =\ \ 5n$$

Thus Case 1 (effectively using an invalid assignment to satisfy φ as a variable needs to be both **true** and **false**) leads to a labelling which is exactly the same size as $\mathcal{L}ab$: both have exacly $5n$ `undec` arguments.

Case 2: $\mathcal{L}ab'$ properly labels all C_j arguments as `out` through a labelling of $\{z_i, \neg z_i\}$ in which exactly one of $\mathcal{L}ab'(z_i) = \texttt{in}$ or $\mathcal{L}ab'(\neg z_i) = \texttt{in}$ holds.

Now this case has n (z arguments) together with $4n + 1$ other arguments ($\{b_i, \neg b_i\}$, $\{c_i\}$, $\{g_i\}$, π) whose status is `undec`, leading to $n{+}2n{+}n{+}n{+}1 \;=\; 5n{+}1$ undecided arguments and a smaller number of committed arguments than $\mathcal{L}ab$. Consider, however, the assignment of propositional $(a_1, a_2, \ldots, a_n)$ values to Z formed through

$$a_i \quad = \quad \begin{cases} \textbf{true} \text{ if } \mathcal{L}ab'(z_i) = \texttt{in} \text{ and } \mathcal{L}ab'(\neg z_i) = \texttt{undec} \\ \textbf{false} \text{ if } \mathcal{L}ab'(z_i) = \texttt{undec} \text{ and } \mathcal{L}ab'(\neg z_i) = \texttt{in} \end{cases}$$

This assignment guarantees that every clause C_j of $\varphi(Z)$ will have at least one literal which evaluates to **true** (since the corresponding C_j argument is correctly labelledl `out` by virtue of being attacked by a literal labelled `in`).

In total $(a_1, a_2, \ldots, a_n)$ is a setting of Z in which every clause of φ contains a **true** literal, i.e. $(a_1, a_2, \ldots, a_n)$ witnesses that $\varphi(Z)$ would be rejected as an instance of CNF−UNSAT.

We deduce that if $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is rejected as an instance of MSAL then $\varphi(Z)$ is rejected as an instance of CNF−UNSAT.

In total, $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ describes an admissible labelling with minimal size of θ as `in` if and only if φ is unsatisfiable. □

4. Discussion

The concept of a strong admissibility is related to grounded semantics in a similar way as the concept of admissibility is related to preferred semantics. In order to prove that an argument is in the grounded extension, we do not have to construct the entire grounded extension. Instead, it is sufficient to construct a strongly admissible set containing it. Similarly, in order to prove that an argument is in a preferred extension, we do not have to construct the entire preferred extension. Instead, it is sufficient to construct an admissible set containing it.

In essence, constructing an admissible set is what is being done by the Preferred Discussion Game [10]. The rules of this game are such that an argument is in an admissible set (and therefore in a preferred extension) if the proponent has a winning strategy for this game. Such a winning strategy can be derived using an admissible set $\mathcal{A}rgs$ that contains the argument A in question. When doing so, the resulting game will have a number of moves that is no greater than $2 \cdot |\mathcal{A}rgs^-| + 1$. It has been shown [10] that in order to minimize the number of moves required in the Preferred Discussion Game, one needs to obtain an admissible set $\mathcal{A}rgs$ that contains A and where $|\mathcal{A}rgs^-|$ is minimal among all the admissible sets that contain A.

The desire to minimize $|\mathcal{A}rgs^-|$ leads to two relevant decision problems: that of *verification* where given an AF (Ar, att) and a set $\mathcal{A}rgs$ that contains argument A it is asked if $\mathcal{A}rgs$ is an admissible set where $|\mathcal{A}rgs^-|$ is minimal among all admissible sets containing A; and the *existence* where given an AF (Ar, att), an argument A and an integer k it is asked if there is an admissible set $\mathcal{A}rgs$ that contains A with $|\mathcal{A}rgs^-| \leq k$.

It was found that the verification problem is coNP–complete, and the existence problem is NP–complete [10].

Table 1 provides an overview of how the main results of the current paper (Theorem 1, Theorem 2) compare with the status of similar problems with respect to standard Dung-style admissibility.

Table 1. Admissibility vs. Strong Admissibility

Problem	Complexity (ADM)	Complexity (Strong ADM)
Verification	Polynomial	Polynomial
Acceptability	NP–complete	Polynomial
Minimal Labelling (existence)	NP–complete	NP–complete
Minimal Labelling (verification)	coNP–complete	coNP–complete

With the exception of (credulous) acceptability these have similar complexity. The discrepancy that acceptability is NP–complete (standard Dung admissibility) whereas the analogous decision problem for strong admissibility is polynomial time decidable, arises from the fact that there is a unique maximal (w.r.t $\subseteq$) strongly admissible set, namely the grounded extension. Thus a simple test as to whether x is contained in a strongly admissible set is just to check if x is in the grounded extension.

It is also worth noting the differences between the reductions to establish intractability as given for admissibility (from [10]) and the constructions in Theorem 1, Theorem 2 for the analogous strong admissibility problems. All four proofs turn on variations of the standard translation of CNF-SAT, see e.g [16, Defn. 5.1, p. 91]. In both [10, Theorem 6.6] (verification of labelling minimality) and [10, Theorem 6.7] (existence of labelling with given size) the constructions used cyclic AFs whose grounded extension is empty. For the cases considered in Theorems 1, 2 we need to have AFs with a non-empty grounded extension. The constructions used, however, go one step further as summarized in the following.

Theorem 3.

a. BSAL *is* NP*–complete if instances are restricted to acyclic frameworks.*
b. MSAL *is co*NP*–complete if instances are restricted to acyclic frameworks.*

Proof. Immediate from the proofs of Theorem 1 and Theorem 2. □

It is worth noting that while there are a very small number of intractability results involving acyclic AFs (e.g. [14, Theorem 23] with binary tree forms) typically these rely on developments of standard Dung frameworks, e.g. the result from [14] exploits properties of value–based argumentation from [2]).

The research of the current paper fits into our long-term research agenda of using argumentation theory to provide explainable formal inference. In our view, it is not enough for a knowledge-based system to simply provide an answer regarding what to do or what to believe. There should also be a way for this answer to be explained. One way of doing so is by means of (formal) discussion [8]. Here, the idea is that the knowledge-based system should provide the argument that is at the basis of its advice. The user is then allowed to raise objections (counter-arguments) which the system then replies to (using counter-counter-arguments), etc. In general, we would like such a discussion to be (1) sound and complete for the underlying argumentation semantics, (2) not be unnecessarily long, and (3) be close enough to human discussion in order to be perceived as natural and convincing.

As for point (1), sound and complete discussion games have been identified for grounded, preferred, stable and ideal semantics [8]. As for point (2), this is what we studied in the current paper, as well as in [10]. As for point (3), this is something that we are aiming to report on in future work.

References

[1] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.

[2] T. J. M. Bench-Capon. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[3] R. Booth, M.W.A. Caminada, and B. Marshall. DISCO: A web-based implementation of discussion games for grounded and preferred semantics. In *Proceedings of COMMA 2018*, pages 453–454, 2018.

[4] M. W. A. Caminada and P. E. Dunne. Strong admissibility revised: theory and applications. *Argument & Computation*, 10:277–300, 2019.

[5] M.W.A. Caminada. On the issue of reinstatement in argumentation. In M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, pages 111–123. Springer, 2006. LNAI 4160.

[6] M.W.A. Caminada. Strong admissibility revisited. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Computational Models of Argument; Proceedings of COMMA 2014*, pages 197–208. IOS Press, 2014.

[7] M.W.A. Caminada. A discussion game for grounded semantics. In E. Black, S. Modgil, and N. Oren, editors, *Theory and Applications of Formal Argumentation (proceedings TAFA 2015)*, pages 59–73. Springer, 2015.

[8] M.W.A. Caminada. Argumentation semantics as formal discussion. In *Handbook of Formal Argumentation*, volume 1, pages 487–518. College Publications, 2018.

[9] M.W.A. Caminada. Strong admissibility in acyclic argumentation frameworks. Technical report, Cardiff University, 2020.

[10] M.W.A. Caminada, W. Dvořák, and S. Vesic. Preferred semantics as socratic discussion. *Journal of Logic and Computation*, 26:1257–1292, 2014.

[11] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.

[12] M.W.A. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[13] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[14] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10):701 – 729, 2007.

[15] P. E. Dunne, A. Gibbons, and M. Zito. Complexity-theoretic models of phase transitions in search problems. *Theoretical Computer Science*, 249(2):243 – 263, 2000.

[16] P. E. Dunne and M. J. Wooldridge. Complexity of abstract argumentation. In G. Simari and I. Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 85–104. Springer US, Boston, MA, 2009.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200500

Deductive Joint Support for Rational Unrestricted Rebuttal

Marcos CRAMER [a] and Meghna BHADRA [a]
[a] *International Center for Computational Logic, TU Dresden, Germany*

Abstract. In ASPIC-style structured argumentation an argument can rebut another argument by attacking its conclusion. Two ways of formalizing rebuttal have been proposed: In restricted rebuttal, the attacked conclusion must have been arrived at with a defeasible rule, whereas in unrestricted rebuttal, it may have been arrived at with a strict rule, as long as at least one of the antecedents of this strict rule was already defeasible. One systematic way of choosing between various possible definitions of a framework for structured argumentation is to study what rationality postulates are satisfied by which definition, for example whether the closure postulate holds, i.e. whether the accepted conclusions are closed under strict rules. While having some benefits, the proposal to use unrestricted rebuttal faces the problem that the closure postulate only holds for the grounded semantics but fails when other argumentation semantics are applied, whereas with restricted rebuttal the closure postulate always holds. In this paper we propose that ASPIC-style argumentation can benefit from keeping track not only of the attack relation between arguments, but also the relation of deductive joint support that holds between a set of arguments and an argument that was constructed from that set using a strict rule. By taking this deductive joint support relation into account while determining the extensions, the closure postulate holds with unrestricted rebuttal under all admissibility-based semantics. We define the semantics of deductive joint support through the flattening method.

Keywords. knowledge representation, structured argumentation, ASPIC, bipolar argumentation, rationality postulates, unrestricted rebuttal

1. Introduction

Formal argumentation has become a fruitful field of research within AI [16]. It comprises two main branches: Abstract argumentation is based on the idea promoted by Dung [11] that under some conditions, the acceptance of arguments depends only on the *attack relation* between the arguments, i.e. on the relation that holds between a counterargument and the argument that it counters. This idea gives rise to the notion of an *argumentation framework* (*AF*), a directed graph whose nodes represent arguments and whose edges represent the attack relations between them, as well as to the notion of an *argumentation semantics*, a way of choosing accepted arguments from an argumentation framework. Structured argumentation, on the other hand, studies the internal structure of arguments that are constructed in some logical language and specifies how this internal structure determines the attack relation between the arguments. Once the attack relation has been specified, the argumentation semantics from abstract argumentation can be applied to determine the acceptability of arguments.

One important approach within structured argumentation is that of ASPIC-style frameworks like ASPIC+ [13] and ASPIC$-$ [5], in which arguments are constructed by applying strict and defeasible rules to strict and defeasible premises. In these ASPIC-style frameworks one can distinguish various kinds of attacks depending on which part of an argument gets questioned. One kind of attack is a rebuttal, in which one argument attacks the conclusion of another argument. Two ways of formalizing rebuttal have been proposed: ASPIC+ makes use of *restricted rebuttal*, in which the attacked conclusion must have been arrived at with a defeasible rule, whereas ASPIC$-$ makes use of *unrestricted rebuttal*, in which the attacked conclusion may have been arrived at with a strict rule, as long as at least one of the antecedents of this strict rule was already defeasible.

One systematic way of choosing between various possible definitions of a framework for structured argumentation is to study what rationality postulates are satisfied by which definition [3]. One example of such a rationality postulate is the closure postulate, according to which the accepted conclusions should be closed under strict rules, i.e. a statement that is derivable by applying a strict rule to some accepted conclusion should itself be an accepted conclusion. For all admissibility-based argumentation semantics, e.g. grounded, complete, stable or preferred semantics, ASPIC+ satisfies the closure postulate. ASPIC$-$ on the other hand only satisfies closure under the grounded semantics, but fails to do so for the others, such as the preferred semantics. This failure of the closure postulate is due to the use of unrestricted rebuttal. On the other hand, from the point of view of human argumentation, unrestricted rebuttal seems to be a very natural way of attacking an argument. This intuition has also been underpinned through an empirical study of human evaluation of arguments [18].

In this paper we propose a modification to ASPIC$-$, called *Deductive ASPIC$-$*, that ensures that the closure postulate is satisfied under all admissibility-based argumentation semantics. The underlying idea is to keep track not only of the attack relation between arguments, but also the relation of deductive joint support that holds between a set of arguments and an argument that was constructed from that set using a strict rule. For this purpose, we introduce the notion of a *Joint Support Bipolar Argumentation Framework* (*JSBAF*), which contains an attack relation like usual AFs and additionally a joint support relation, whose intuitive interpretation is a deductive support from the supporting arguments towards the supported arguments due to the latter being constructed by applying a strict rule to the former. We show how existing argumentation semantics for AFs can be adapted to semantics for JSBAFs using the flattening method. We prove that the resulting framework for structured argumentation satisfies closure as well as two other important rationality postulates, direct consistency and indirect consistency. In this paper we limit ourselves to structured argumentation without preferences, leaving the generalization of the results to preference-based argumentation for future work.

The paper is structured as follows: Section 2 contains required preliminaries from abstract and structured argumentation. In Section 3 we define JSBAFs and show how existing argumentation semantics for AFs can be adapted to semantics for JSBAFs using the flattening method. In Section 4 we define Deductive ASPIC$-$, prove that it satisfies the closure postulate and the two consistency postulates, and finally illustrate the functioning of Deductive ASPIC$-$ by adapting Caminada's tandem example [3] to Deductive ASPIC$-$. Section 5 concludes the paper and presents avenues for further research.

2. Preliminaries of Abstract and Structured Argumentation

This section briefly presents some required preliminaries of abstract and structured argumentation, starting with the notion of an *argumentation framework* due to Dung [11].

Definition 2.1. An *argumentation framework (AF)* $F = (Ar, \rightarrow)$ is a (finite or infinite) directed graph in which the set Ar of vertices is considered to represent arguments and the set $\rightarrow \subseteq Ar \times Ar$ of edges is considered to represent the attack relation between arguments, i.e. the relation between a counterargument and the argument that it counters.

Given an argumentation framework, we want to choose sets of arguments for which it is rational and coherent to accept them together. Such a set of arguments that may be accepted together is called an *extension*. Multiple *argumentation semantics* have been defined in the literature, i.e. multiple different ways of defining extensions given an argumentation framework. Before we consider specific argumentation semantics, we first give a formal definition of the notion of an *argumentation semantics*:

Definition 2.2. An *argumentation semantics* is a function σ that maps any AF $F = (Ar, \rightarrow)$ to a set $\sigma(F) \subseteq 2^{Ar}$. The elements of $\sigma(F)$ are called σ-extensions of F.

In this paper we consider the complete, stable, grounded and preferred semantics:

Definition 2.3. Let $F = (Ar, \rightarrow)$ be an AF, and let $S \subseteq Ar$. The set S is called *conflict-free* iff there are no arguments $b, c \in S$ such that b attacks c (i.e. such that $(b,c) \in \rightarrow$). Argument $a \in Ar$ is *defended* by S iff for every $b \in Ar$ such that b attacks a there exists $c \in S$ such that c attacks b. We say that S is *admissible* iff S is conflict-free and every argument in S is defended by S.

- S is a *complete extension* of F iff S is admissible and S contains all the arguments it defends.
- S is a *stable extension* of F iff S is admissible and S attacks all arguments in $Ar \setminus S$.
- S is the *grounded extension* of F iff S is the minimal (with respect to set inclusion) complete extension of F.
- S is a *preferred extension* of F iff S is a maximal (with respect to set inclusion) complete extension of F.

All of these four semantics satisfy the property that every extension is an admissible set. Due to this property they are called *admissibility-based semantics*.

We now turn towards the definition of ASPIC−, a framework for structured argumentation introduced by [5].

Definition 2.4. Given a logical language L that is closed under negation ($\neg$), an *argumentation system* over L is a tuple $AS = (R_s, R_d, n)$ where:

- R_s is a finite set of strict inference rules of the form $\varphi_1, \ldots, \varphi_k \mapsto \varphi$ where φ_i, φ are elements in L and $k \geq 0$.
- R_d is a finite set of defeasible inference rules of the form $\varphi_1, \ldots, \varphi_k \Rightarrow \varphi$ where φ_i, φ are elements in L and $k \geq 0$.
- n is a partial function such that $n : R_d \rightarrow L$.

Definition 2.5. Let φ and ψ be formulas. $\varphi = -\psi$ means that $\varphi = \neg\psi$ or $\psi = \neg\varphi$.

Definition 2.6. An *argument A on the basis of* an argumentation system $AS = (R_s, R_d, n)$ is defined recursively as follows:

- $A_1, \dots, A_n \mapsto \psi$ is an argument if $A_1, \dots, A_k (k \geq 0)$ are arguments, and there is a strict rule $r = Conc(A_1), \dots, Conc(A_n) \mapsto \psi$ in R_s. In that case $DefRules(A) := DefRules(A_1) \cup \dots \cup DefRules(A_k)$.
- $A_1, \dots, A_n \Mapsto \psi$ is an argument if $A_1, \dots, A_k (k \geq 0)$ are arguments, and there is a defeasible rule $r = Conc(A_1), \dots, Conc(A_n) \Mapsto \psi$ in R_d. In that case $DefRules(A) := DefRules(A_1) \cup \dots \cup DefRules(A_k) \cup \{r\}$.

In both cases we define $Conc(A) := \psi$, $Sub(A) := Sub(A_1) \cup \dots \cup Sub(A_k) \cup \{A\}$ and $TopRule(A) := r$. Furthermore, we call an argument A *defeasible* iff $DefRules(A) \neq \emptyset$.

Definition 2.7. An argumentation system AS is called *consistent* iff there are no strict arguments A, B on the basis of AS such that $Conc(A) = -Conc(B)$.

Definition 2.8. Let A and B be arguments on the basis of an argumentation system $AS = (R_s, R_d, n)$. We say that A *undercuts* B (on B') iff $Conc(A) = -n(r)$ for some $B' \in Sub(B)$ with $TopRule(B') = r$ and $r \in R_d$. We say that A *unrestrictedly rebuts* B (on B') iff $Conc(A) = -Conc(B')$ for some defeasible $B' \in Sub(B)$.

Definition 2.9. Let $AS = (R_s, R_d, n)$ be an argumentation system. The *argumentation framework corresponding to AS according to ASPIC−*, denoted by $F_{\text{ASPIC}-}(AS)$, is the AF $(Ar, \rightarrow)$, where Ar is the set of arguments on the basis of AS and $A \rightarrow B$ holds whenever A undercuts or unrestrictedly rebuts B.

Definition 2.10. Let $AS = (R_s, R_d, n)$ be an argumentation system and let σ be an argumentation semantics. For every σ-extension E of $F_{\text{ASPIC}-}(AS)$, the set $\{Conc(A) \mid A \in E\}$ is called a *set of ASPIC− conclusions of AS under* σ.

This concludes the definition of the structured argumentation framework ASPIC−. Finally we define three rationality postulates due to Caminada and Amgoud [4] that structured argumentation frameworks should ideally satisfy.

Definition 2.11. Let L be a logical language, let $AS = (R_s, R_d, n)$ be an argumentation system over L, and let $S \subseteq L$. The *closure of S under strict rules*, denoted $Cl_{R_s}(S)$, is the smallest set such that $Cl_{R_s}(S) \supseteq S$ and for every strict rule $a_1, \dots, a_n \mapsto b \in R_s$ such that $a_1, \dots, a_n \in Cl_{R_s}(S)$, we have $b \in Cl_{R_s}(S)$.

Definition 2.12. Let $\mathscr{F}$ be a framework for structured argumentation (e.g. ASPIC−) and let σ be an argumentation semantics.

- $\mathscr{F}$ satisfies *closure* under σ iff for every consistent argumentation system AS and for every set C of $\mathscr{F}$-conclusions of AS under σ, we have $Cl_{R_s}(C) = C$.
- $\mathscr{F}$ satisfies *direct consistency* under σ iff for every consistent argumentation system AS and every set C of $\mathscr{F}$-conclusions of AS under σ, there is no ϕ such that $\phi, \neg\phi \in C$.
- $\mathscr{F}$ satisfies *indirect consistency* under σ iff for every consistent argumentation system AS and every set C of $\mathscr{F}$-conclusions of AS under σ, there is no ϕ such that $\phi, \neg\phi \in Cl_{R_s}(C)$.

Caminada et al. [5] showed that ASPIC− satisfies these three postulates under the grounded semantics, whereas Caminada and Wu [6] showed that closure and indirect consistency are violated by ASPIC− under the preferred semantics.

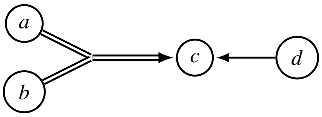
3. Abstract Argumentation with Deductive Joint Support

Multiple ways of augmenting argumentation frameworks with a support relation between arguments have been considered in the literature [2,14,15], giving rise to a lively research field called *bipolar argumentation*. One of the ways of formally interpreting the support relation is called *deductive support* [2]. Here the idea is that when an argument a deductively supports an argument b, this is similar to the situation when a formula ϕ logically entails a formula ψ, i.e. if one accepts a, one should also accept b.

In this section we extend the deductive support relation to a *deductive joint support* relation, in which multiple arguments together can deductively support another argument. Again the intuitive idea is similar to when multiple formulas entail another formula: When $a_1,\dots,a_n$ jointly deductively support b, this means that if $a_1,\dots,a_n$ are all accepted, b should also be accepted. In this section we introduce the notion of a *Joint Support Bipolar Argumentation Framework* (JSBAF) in which such joint support relations appear alongside the usual attack relation. We then show how the flattening methodology (see [1]) can be used to adapt standard argumentation semantics to semantics for JSBAFs that give a deductive interpretation to the joint support relation.

Definition 3.1. A *Joint Support Bipolar Argumentation Framework* (JSBAF) is a triple $(Ar, \rightarrow, \Rightarrow)$ such that Ar is the set of all arguments in the framework, $\rightarrow \subseteq Ar \times Ar$ is an attack relation and $\Rightarrow \subseteq 2^{Ar} \times Ar$ is a joint support relation.

Example 3.2. As an example, we illustrate below a JSBAF J_1 in which arguments a and b jointly support c which is attacked by argument d:



Now that we have formally defined a JSBAF, let us move on to its semantics. The principle idea is the same as in argumentation frameworks.

Definition 3.3. A JSBAF Semantics is a function that maps every JSBAF $J = (Ar, \rightarrow, \Rightarrow)$ to a set $\sigma(J) \subseteq 2^{Ar}$. The elements of $\sigma(J)$ are called σ-extensions.

The deductive property of the joint support relation inspires the following notion of a deductive JSBAF semantics:

Definition 3.4. A JSBAF semantics σ is called *deductive* iff for every JSBAF $J = (Ar, \rightarrow, \Rightarrow)$, for every σ-extension E of J, every set $S \subseteq E$ and every $A \in Ar$ such that $S \Rightarrow A$, we have $A \in E$.

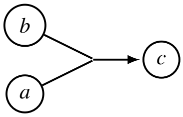
Furthermore, we will require the notion of a conflict-free JSBAF semantics:

Definition 3.5. A JSBAF semantics σ is called *conflict-free* iff for every JSBAF $J = (Ar, \rightarrow, \Rightarrow)$, for every σ-extension E and any $a, b \in E$, $(a,b) \notin \rightarrow$.

We will flatten JSBAFs to standard AFs in a two-step process. In the first step, we flatten JSBAFs to *higher-level argumentation frameworks* (originally introduced by Gabbay [12]) that contain joint attacks.

Definition 3.6. We define a *higher-level argumentation framework* (*higher level AF*) as a tuple $(Ar, \rightarrow)$ where Ar is the set of arguments in the framework and $\rightarrow \subseteq (2^{Ar} \setminus \{\emptyset\}) \times Ar$ is a joint attack relation.

Example 3.7. The following is an example of a higher-level AF, where arguments a and b jointly attack c:



We will now define the two-step process of flattening a JSBAF into a standard AF. The first step involves transforming the JSBAF to a higher-level AF by converting joint supports to joint attacks while introducing some meta-arguments. The second step involves transforming the higher-level AF to a standard AF by converting joint attacks to regular one-on-one attacks.

Definition 3.8. Let $J = (Ar, \rightarrow, \Rightarrow)$ be a JSBAF. The one-step flattening of J, denoted by $\mathit{flat}_1(J)$, is a higher-level AF $(MS, \rightarrow_1)$, where $MS := Ar \cup \{\bar{b} \mid (X,b) \in \Rightarrow\}$ and the joint attack relation $\rightarrow_1$ is defined as follows:

- For each $(a,b) \in \rightarrow$, we have $(a,b) \in \rightarrow_1$.
- For each $(X,b) \in \Rightarrow$, we have $b \rightarrow_1 \bar{b}$.
- For each X with $(X,b) \in \Rightarrow$ and for every $a \in X$, we have $(X \setminus \{a\}) \cup \{\bar{b}\} \rightarrow_1 a$.

In what follows, we present three examples of this flattening:

Example 3.9. On the left, Figure 1 depicts a JSBAF J_2 consisting of an argument a supporting another argument b. On the right, Figure 1 depicts its one-step flattening $\mathit{flat}_1(J_2)$.

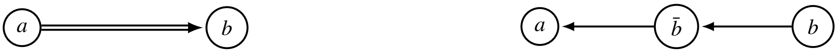
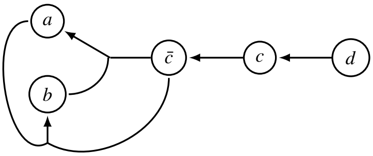
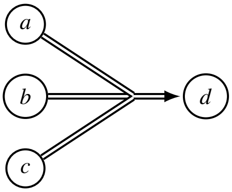


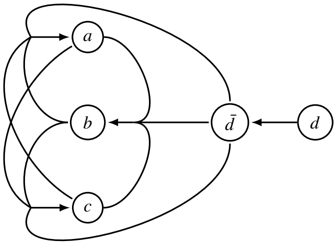
Figure 1. JSBAF J_2 and its one-step flattening $\mathit{flat}_1(J_2)$

Example 3.10. As a second example we reconsider the JSBAF J_1 from Example 3.2. The following is its one-step flattening $\mathit{flat}_1(J_1)$:



Example 3.11. As a last example, we illustrate a JSBAF J_3 where arguments $\{a, b, c\}$ jointly support d (on the left) as well as its one-step flattening $\mathit{flat}(J_3)$ (on the right):





In the next step, we define how a higher-level AF can be flattened to a standard AF. This flattening is originally due to Gabbay [12].

Definition 3.12. Let $H = (Ar, \twoheadrightarrow)$ be a higher-level AF. The flattening of H, denoted by $\mathit{flat}_2(H)$, is a standard argumentation framework $(MS, \twoheadrightarrow_2)$, where the set MS of meta-arguments and the attack relation $\twoheadrightarrow_2$ are defined as follows:

- $MA = Ar \cup \{\bar{a} \mid$ there exists an attack $(X,b) \in \twoheadrightarrow$ with $a \in X$ and $|X| > 1\} \cup \{e(X) \mid$ there exists an attack $(X,b) \in \twoheadrightarrow$ with $|X| > 1\}$.
- For all $(\{a\},b) \in \twoheadrightarrow$, we have $a \twoheadrightarrow_2 b$.
- For all $(X,b) \in \twoheadrightarrow$ where $|X| > 1$, we have $e(X) \twoheadrightarrow_2 b$, and for every $a \in X$, we have $a \twoheadrightarrow_2 \bar{a}$ and $\bar{a} \twoheadrightarrow_2 e(X)$.

Example 3.13. The following figure depicts the flattening $\mathit{flat}_2(\mathit{flat}_1(J_1))$ of the higher-level AF $\mathit{flat}_1(J_1)$ depicted in Example 3.10. This example also illustrates how to combine the flattenings flat_1 and flat_2 in order to flatten a JSBAF a standard AF in two flattening steps.

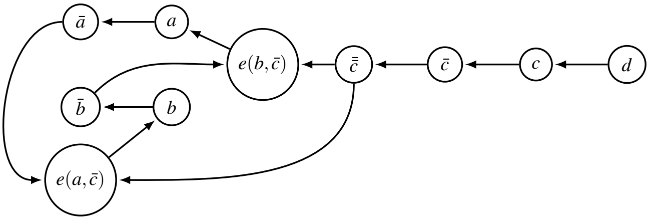


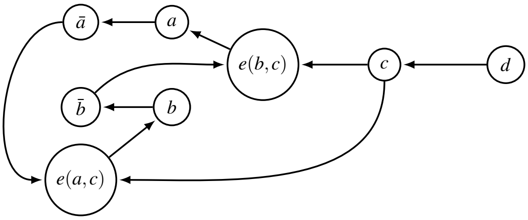
Figure 3. The flattening $\mathit{flat}_2(\mathit{flat}_1(J_1))$ of the higher-level AF $\mathit{flat}_1(J_1)$ depicted in Example 3.10

During the two-step process of flattening a JSBAF framework to an abstract argumentation framework, meta-arguments like the $\bar{c}$ and the $\bar{\bar{c}}$ were introduced. As shown in Figure 3, $\bar{\bar{c}}$ is attacked only by $\bar{c}$ which is in turn attacked only by c. So intuitively, if c is accepted $\bar{\bar{c}}$ should also be accepted. Similarly if argument c is rejected, $\bar{\bar{c}}$ should also be rejected. As a result both $\bar{c}$ and $\bar{\bar{c}}$ can be omitted, thus simplifying the flattened framework. This inspires the following definition for the simplified flattening of a JSBAF to an AF:

Definition 3.14. Let $J = (Ar, \twoheadrightarrow, \Rightarrow)$ be a JSBAF. We write $(MA', \twoheadrightarrow')$ for $\mathit{flat}_2(\mathit{flat}_1(J))$. Then the simplified flattening of J, denoted by $\mathit{flat}(J)$, is a standard argumentation framework $(MA^*, \twoheadrightarrow^*)$ defined as follows:

- $MA^* := MA' \setminus \{\bar{a}, \bar{\bar{a}} \mid \text{there is some } (S,a) \in \Rightarrow \text{ with } |S| > 1\}$
- $\twoheadrightarrow^* := \{(a,b) \in MA^* \times MA^* \mid a \twoheadrightarrow' b\} \cup \{(a,b) \in MA^* \times MA^* \mid \bar{\bar{a}} \twoheadrightarrow' b\}$

Example 3.15. The following figure depicts the simplified flattening $\mathit{flat}(J_1)$ of the JSBAF J_1 from Example 3.2:



The flattening function *flat* allows us to define semantics of JSBAFs by first applying the flattening function and then applying a standard argumentation semantics:

Definition 3.16. For any Dung semantics σ, we define a JSBAF semantics $sup(\sigma)$ as follows: E is a $sup(\sigma)$-extension of $(Ar, \rightarrow, \Rightarrow)$ iff there is an extension E' of $flat(Ar, \rightarrow, \Rightarrow)$ such that $E = E' \cap Ar$.

The following two lemmas establish useful facts about $sup(\sigma)$:

Lemma 3.17. *For any admissibility-based semantics σ of Dung's abstract frameworks, $sup(\sigma)$ is a deductive JSBAF semantics.*

Proof. Let $J = (Ar, \rightarrow, \Rightarrow)$ be a JSBAF and let E be a $sup(\sigma)$-extension of J. By Definition 3.16, there is a σ-extension E' of $flat(J) = (MA^*, \rightarrow^*)$ such that $E = E' \cap Ar$. Let $S \subseteq E$ and $d \in Ar$ be such that $S \Rightarrow d$. We need to show that $d \in E$. We distinguish two cases:
Case 1: $|S| = 1$, say $S = \{a\}$. In this case, in $flat(J)$, $S \Rightarrow d$ is flattened to the attacks $d \rightarrow^* \bar{d}$ and $\bar{d} \rightarrow^* a$. Since $a \in E'$ and E' is admissible, E' defends a, i.e. E' attacks $\bar{d}$. But d is the only attacker of $\bar{d}$ in $flat(J)$, so $d \in E'$, i.e. $d \in E$.
Case 2: $|S| > 1$. In this case, in $flat(J)$, $S \Rightarrow d$ is flattened into the following attacks for each element a in S:

- $a \rightarrow^* \bar{a}$
- for each $b \in S \setminus \{a\}$: $\bar{b} \rightarrow^* e(\{d\} \cup (S \setminus \{a\}))$
- $e(\{d\} \cup (S \setminus \{a\})) \rightarrow^* a$
- $d \rightarrow^* e(\{d\} \cup (S \setminus \{a\}))$

Let $a \in S$. Then $a \in E'$, i.e. E' defends a. Since $e(\{d\} \cup (S \setminus \{a\})) \rightarrow^* a$, so E' attacks $e(\{d\} \cup (S \setminus \{a\}))$. However, for each element b in $(S \setminus \{a\})$, $b \in E'$, so by the conflict-freeness of E', $\bar{b} \notin E'$. So the only element of E' that can attack $e(\{d\} \cup (S \setminus \{a\}))$ is d. So $d \in E'$, i.e. $d \in E$. □

Lemma 3.18. *For any admissibility-based semantics σ of Dung's abstract frameworks, $sup(\sigma)$ is a conflict-free JSBAF semantics.*

Proof. Let $J = (Ar, \rightarrow, \Rightarrow)$ be a JSBAF and let E be a $sup(\sigma)$-extension of J. By Definition 3.16, there is a σ-extension E' of $flat(J)$ such that $E = E' \cap Ar$. E is conflict-free because $E = E' \cap Ar$ and E' is conflict-free. □

Example 3.19. By Lemmas 3.17 and 3.18, $sup(complete)$, $sup(stable)$, $sup(grounded)$ and $sup(preferred)$ are deductive, conflict-free JSBAF semantics.

4. Deductive ASPIC− and the Rationality Postulates

In this section we show how the deductive joint support relation can be applied to structured argumentation in order to satisfy the three rationality postulates (closure, direct and indirect consistency) in the context of unrestricted rebuttal. For this purpose, we define *Deductive ASPIC−* (abbreviated as *DA−*), show that it satisfies these three postulates and illustrate the result with an example.

In Deductive ASPIC−, arguments and attacks are defined in the same way as in ASPIC−, but we also define a deductive joint support relation between arguments:

Definition 4.1. Let AS be an argumentation system. The *JSBAF corresponding to AS according to Deductive ASPIC−*, denoted by $J_{\text{DA}-}(AS)$, is the JSBAF $(Ar, \rightharpoonup', \Rightarrow')$, where Ar is the set of arguments on the basis of AS, $A \rightharpoonup' B$ holds iff A undercuts or unrestrictedly rebuts B, and $S \Rightarrow' A$ holds iff A is of the form $S \mapsto \varphi$ for some formula φ.

Definition 4.2. Let $AS = (R_s, R_d, n)$ be an argumentation system and let σ be a JSBAF semantics. For every σ-extension E of $J_{\text{DA}-}(AS)$, the set $\{Conc(A) \mid A \in E\}$ is called a *set of DA− conclusions of AS under* σ.

We now show that Deductive ASPIC− satisfies closure, direct consistency and indirect consistency under any deductive, conflict-free JSBAF semantics.

Lemma 4.3. *Let σ be a deductive, conflict-free JSBAF semantics. Then Deductive ASPIC− satisfies closure under σ.*

Proof. Let $AS = (R_s, R_d, n)$ be an argumentation system and let C be a set of DA− conclusions of AS under σ. Let $S \mapsto \varphi$ be a strict rule in R_s with $S \subseteq C$. We need to show that $\varphi \in C$. By Definition 4.2, there is a σ-extension E of $J_{\text{DA}-}(AS) = (Ar, \rightharpoonup', \Rightarrow')$ such that $C = \{Conc(A) \mid A \in E\}$. Since $S \subseteq C$, there is a set $F \subseteq E$ such that $S = \{Conc(A) \mid A \in F\}$. Then $F \mapsto \varphi$ is an argument on the basis of AS. By Definition 4.1, $F \Rightarrow' (F \mapsto \varphi)$. Since E is a σ-extension for a deductive JSBAF semantics σ, $F \subseteq E$ and $F \Rightarrow' (F \mapsto \varphi)$ imply that $(F \mapsto \varphi) \in E$. So $\varphi = Conc(F \mapsto \varphi) \in \{Conc(A) \mid A \in E\} = C$. □

Lemma 4.4. *Let σ be a deductive, conflict-free JSBAF semantics. Then Deductive ASPIC− satisfies direct consistency under σ.*

Proof. Let $AS = (R_s, R_d, n)$ be a consistent argumentation system and let C be a set of DA− conclusions of AS under σ. Suppose for a contradiction that $\varphi, \neg\varphi \in C$. By Definition 4.2, there is a σ-extension E of $J_{\text{DA}-}(AS) = (Ar, \rightharpoonup', \Rightarrow')$ such that $C = \{Conc(A) \mid A \in E\}$. Then there are $F, F' \in E$ such that $Conc(F) = \phi$ and $Conc(F') = \neg\phi$. By the consistency of AS, F and F' cannot both be strict. Without loss of generality, assume F' is not strict. Then F unrestrictedly rebuts F', which contradicts the conflict-freeness of E. □

Lemmas 4.3 and 4.4 together imply the following lemma about indirect consistency:

Lemma 4.5. *Let σ be a deductive, conflict-free JSBAF semantics. Then Deductive ASPIC− satisfies indirect consistency under σ.*

These three lemmas together with Lemmas 3.17 and 3.18 imply the following theorem:

Theorem 4.6. *Let σ be an admissibility-based argumentation semantics. Then Deductive ASPIC− satisfies closure, direct consistency and indirect consistency under $sup(\sigma)$.*

We now illustrate the functioning of Deductive ASPIC− on an example, which is based on the example that Caminada [3] used to show that closure is not satisfied under preferred semantics in ASPIC−. We show how the same example interpreted in Deductive ASPIC− does satisfy closure.

Example 4.7. Suppose Harry, Sally and Tom want to go on a bicycle ride with a tandem. Since the tandem only has two seats, only two of the three can ride it at a time. To formalize this scenario, we use the language $L = \{hw, sw, tw, ht, st, tt\}$, where *hw* means “Harry wants to ride the tandem”, *ht* means “Harry will ride the tandem”, and analogously for Sally (*sw*, *st*) and Tom (*tw*, *tt*). The scenario can be represented by an argumentation system $AS = \{R_s, R_d, n\}$, where $R_s = \{\mapsto hw; \mapsto sw; \mapsto tw; ht, st \mapsto \neg tt; st, tt \mapsto \neg ht; tt, ht \mapsto \neg st\}$, $R_d = \{hw \Mapsto ht; sw \Mapsto st; tw \Mapsto tt\}$ and n is empty. Intuitively, the strict rules $\mapsto hw$, $\mapsto sw$, and $\mapsto tw$ represent that all three of them want to ride on the tandem, the strict rules $ht, st \mapsto \neg tt$, $st, tt \mapsto \neg ht$ and $tt, ht \mapsto \neg st$ represent that the tandem can only seat two people, and the defeasible rules $hw \Mapsto ht$, $sw \Mapsto st$ and $tw \Mapsto tt$ represent that as far as possible each person gets to do what they want to do.

The following arguments on the basis of AS can be constructed:

- $A_1 : \mapsto hw$
- $A_2 : \mapsto sw$
- $A_3 : \mapsto tw$
- $A_4 : A_1 \Mapsto ht$
- $A_5 : A_2 \Mapsto st$
- $A_6 : A_3 \Mapsto tt$
- $A_7 : A_5, A_6 \mapsto \neg ht$
- $A_8 : A_6, A_4 \mapsto \neg st$
- $A_9 : A_4, A_5 \mapsto \neg tt$

The JSBAF $J_{\text{DA}-}(AS)$ is depicted in Figure 4. Its flattening $\mathit{flat}(J_{\text{DA}-}(AS))$ is depicted in Figure 5. The preferred extensions of $\mathit{flat}(J_{\text{DA}-}(AS))$ are as follows:

- $E1' = \{A_1, A_2, A_3, A_9, \bar{A}_6, A_4, A_5, e(A_5, A_7), e(A_4, A_8)\}$
- $E2' = \{A_1, A_2, A_3, A_8, \bar{A}_5, A_4, A_6, e(A_6, A_7), e(A_4, A_9)\}$
- $E3' = \{A_1, A_2, A_3, A_7, \bar{A}_4, A_6, A_5, e(A_6, A_8), e(A_5, A_9)\}$

Given the complexity of $\mathit{flat}(J_{\text{DA}-}(AS))$, we have provided a proof that these three extensions are the only preferred extensions of $\mathit{flat}(J_{\text{DA}-}(AS))$ in a technical report [7].

From this it follows that the *sup*(preferred)-extensions of $J_{\text{DA}-}(AS)$ are $\{A_1, A_2, A_3, A_9, A_4, A_5\}$, $\{A_1, A_2, A_3, A_8, A_4, A_6\}$ and $\{A_1, A_2, A_3, A_7, A_6, A_5\}$. Therefore the sets of DA− conclusions under the *sup*(preferred)-semantics are $\{hw, sw, tw, \neg tt, ht, st\}$, $\{hw, sw, tw, \neg st, ht, tt\}$ and $\{hw, sw, tw, \neg ht, tt, st\}$. The reader can easily verify that each set of DA− conclusions under preferred semantics is closed under the set of strict rules R_s, in line with the closure postulate.

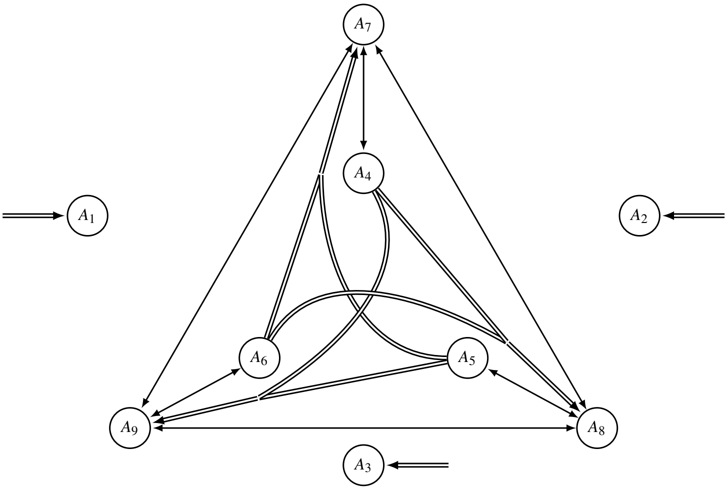


Figure 4. The JSBAF $J_{\text{DA}-}(AS)$

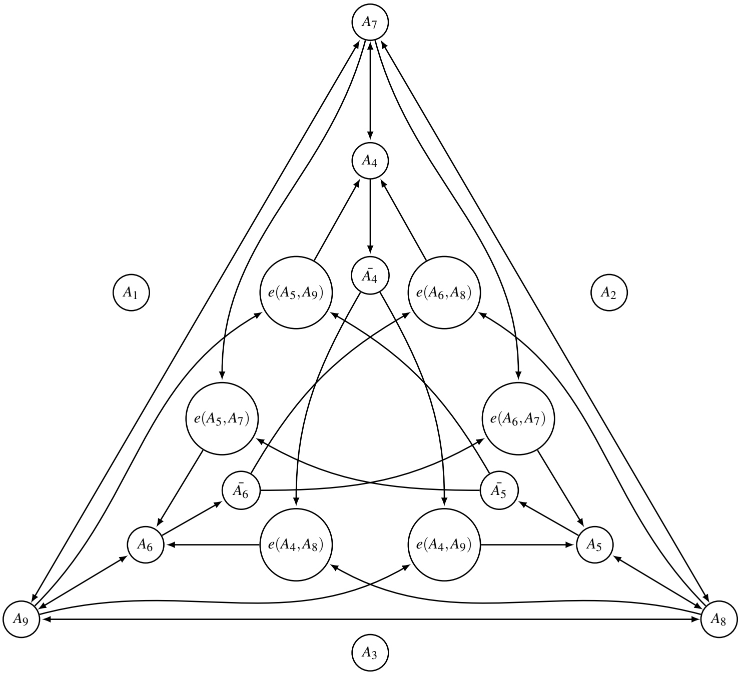


Figure 5. Flattening *flat*$(J_{\text{DA}-}(AS))$ of the JSBAF $J_{\text{DA}-}(AS)$

5. Conclusion and Future Work

Caminada [3] has established that ASPIC+, which uses restricted rebuttal, satisfies the three rationality postulates defined in Section 2 under any of the standard admissibility-based semantics, whereas ASPIC−, which uses unrestricted rebuttal, satisfied closure and indirect consistency only under the grounded semantics. In this paper we defined a modification of ASPIC− called Deductive ASPIC−, which also uses unrestricted rebuttal, but which satisfies all three rationality postulates under any admissibility-based semantics. This is attained by keeping track not only of the attack relation between arguments, but also of the deductive joint support relation between arguments linked by an application of a strict rule.

The methodology introduced in this paper opens up multiple avenues for future research. First, the results presented in this paper have been limited to structured argumentation without preferences, so future work should study how these results could be generalized to a variant of Deductive ASPIC− with preferences.

Furthermore, while the results presented in this paper are limited to admissibility-based semantics, the general methodology is also applicable to naive-based semantics like CF2, SCF2, stage and stage2. So far, the application of these semantics to structured argumentation was limited by the fact that the closure postulate is violated under these semantics, even when restricted rebuttal is used. Given that empirical cognitive studies have found CF2 and SCF2 to be good models of human argument evaluation (see [8,9,10]), it seems to us to be a very worthwhile endeavor to attempt to remedy this situation. However, the approach of using the *sup* operator, i.e. to use JSBAF semantics like *sup*(CF2) or *sup*(SCF2), will not yield to satisfaction of the closure postulate.

Instead, one can adapt these naive-based argumentation semantics to JSBAF semantics in a different way. For example, a JSBAF variant of CF2 could be defined by ensuring the deductiveness property on the level of each SCC. Additionally, the definition of SCC would have to be adapted to account for the effect of deductive joint support (the paths required in the definition of SCCs should be able to pass thorough the support relation as well, however in the backward direction). This way the closure postulate can be made to be satisfied in combination with these naive-based semantics.

Another avenue for future research is to apply the methodology introduced in this paper to tackle the rationality postulates of non-interference and crash resistance (see [3]). Wu and Podlaszewski [17] have introduced an approach to satisfying these postulates by deleting inconsistent arguments, but when preferences are taken into account, this approach fails to satisfy closure. Combining their approach with ours yields a framework in which closure as well as non-interference and crash resistance can be satisfied in the presence of preferences.

References

[1] Guido Boella, Dov M Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling I: Methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[2] Guido Boella, Dov M Gabbay, Leon van der Torre, and Serena Villata. Support in abstract argumentation. In *Computational Models of Argument: Proceedings of COMMA 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 111–122. IOS Press, 2010.

[3] Martin Caminada. Rationality postulates: applying argumentation theory for non-monotonic reasoning. *Journal of Applied Logics*, 4(8):2707–2734, 2017.

[4] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[5] Martin Caminada, Sanjay Modgil, and Nir Oren. Preferences and Unrestricted Rebut. In *Computational Models of Argument - Proceedings of COMMA 2014*, pages 209–220, 2014.

[6] Martin Caminada and Yining Wu. On the limitations of abstract argumentation. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, pages 59–66, 2011.

[7] Marcos Cramer and Meghna Bhadra. Technical Report of "Deductive Joint Support for Rational Unrestricted Rebuttal". *arXiv e-prints*, page arXiv:2005.03620, May 2020.

[8] Marcos Cramer and Mathieu Guillaume. Empirical Cognitive Study on Abstract Argumentation Semantics. *Frontiers in Artificial Intelligence and Applications*, pages 413–424, 2018.

[9] Marcos Cramer and Mathieu Guillaume. Empirical study on human evaluation of complex argumentation frameworks. In *European Conference on Logics in Artificial Intelligence*, pages 102–115. Springer, 2019.

[10] Marcos Cramer and Leendert van der Torre. SCF2 – an argumentation semantics for rational human judgments on argument acceptability. 2019.

[11] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[12] Dov M Gabbay. Fibring argumentation frames. *Studia Logica*, 93(2-3):231–295, 2009.

[13] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[14] Farid Nouioua and Vincent Risch. Argumentation frameworks with necessities. In *International Conference on Scalable Uncertainty Management*, pages 163–176. Springer, 2011.

[15] Nir Oren and Timothy J Norman. Semantics for evidence-based argumentation. *Computational Models of Argument*, 2008.

[16] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[17] Yining Wu and Mikołaj Podlaszewski. Implementing crash-resistance and non-interference in logic-based argumentation. *Journal of Logic and Computation*, 25(2):303–333, 2015.

[18] Zhe Yu, Kang Xu, and Beishui Liao. Structured argumentation: Restricted rebut vs. unrestricted rebut. *Studies in Logic*, 11(3):3–17, 2018.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200501

A First Approach to Argumentation Label Functions

Marcos Cramer [a] and Jérémie Dauphin [b]
[a] *TU Dresden*
[b] *University of Luxembourg*

Abstract. An important approach to abstract argumentation is the labeling-based approach, in which one makes use of labelings that assign to each argument one of three labels: `in`, `out` or `und`. In this paper, we address the question, which of the twenty-seven functions from the set of labels to the set of labels can be represented by an argumentation framework. We prove that in preferred, complete and grounded semantics, eleven label functions can be represented in this way while sixteen label functions cannot be represented by any argumentation framework. We show how this analysis of label functions can be applied to prove an impossibility result: Argumentation frameworks extended with a certain kind of weak attack relation cannot be flattened to the standard Dung argumentation frameworks.

Keywords. knowledge representation, abstract argumentation, argumentation semantics, labelings, flattening

1. Introduction

Abstract argumentation frameworks (AFs) [12] are reasoning structures where one aims at extracting sets of jointly acceptable arguments. One of the central methods to do so is the labeling-based approach [2], in which one derives labelings which assign to each argument one of three labels: `in`, `out` or `und`. The arguments that are labeled `in` represent the arguments that are jointly acceptable, while the arguments that are `out` represent the ones that are defeated by those. The last label, `und` (*undecided*), represents the cases where one cannot, or decides with proper justification, not to assign either of these two labels. One advantage of the labeling approach is that to verify that an argument is correctly labeled, one only needs to check the labels of its direct ancestors. This allows for a more local evaluation, which is still equivalent to other global approaches such as the extension-based approach.

Many enrichments of abstract argumentation frameworks have been studied, e.g. with bipolar argumentation frameworks which add a second relation of support [9], or with argumentation frameworks with recursive attacks (AFRA) [3] in which attacks may also target other attacks. One methodology for evaluating such enriched frameworks while staying coherent with the basic framework is the flattening approach [6], where the enrichments added to the abstract argumentation frameworks are expressed in terms of extra arguments and attacks, allowing one to evaluate them as abstract argumentation frameworks. An essential concern in the flattening approach is whether the extra arguments and attacks produce the same behavior as the one intended by the enrichment they flatten. This raises a question: Which relations connecting two arguments can be expressed in terms of arguments and attacks alone?

In this paper we propose to address this research question by studying the representability of label functions, i.e. of functions which map each of the three labels to one of these labels. We prove that in preferred, complete and grounded semantics, eleven label functions can be represented by an AF while sixteen label functions cannot be represented by any AF. We show how this analysis of label functions can be applied to prove an impossibility result: Argumentation frameworks extended with a certain kind of weak attack relation cannot be flattened to the standard Dung argumentation frameworks. Furthermore we also briefly discuss representability of label functions with respect to the stable semantics.

The structure of the paper is as follows: in Section 2 we formally define the notion of label function and what it means to represent them as abstract argumentation frameworks. In Section 3 we show which of the twenty-seven label functions are representable and which ones are unrepresentable in the context of the complete, grounded and preferred semantics, and briefly mention the case of the stable semantics. In Section 4 we discuss the implications of these impossibility results for the flattening of a particular relation: a weak attack relation that does not propagate the undecided label. We then discuss related work in Section 5 and future work in Section 6. We provide a short conclusion in Section 7.

Due to space limitations, we assume the reader to be familiar with the labeling approach for abstract argumentation [2]. A summary of the required existing notions as well as the proofs of the results of this paper are presented in a technical report [10].

2. Label Functions

In this section we define the basic notions of a label function, an input-output argumentation framework and the representability of a label function. We write *Labs* for the set of possible labels $\{\mathtt{in}, \mathtt{out}, \mathtt{und}\}$.

Definition 1. *A* label function *LF is a function from Labs to Labs.*

Definition 2. *Let LF_1 and LF_2 be two label functions. Then $LF_1 \circ LF_2$ denotes the composition of these two label functions that is defined as $LF_1 \circ LF_2(L) = LF_1(LF_2(L))$.*

We use the triplet $(LF(\mathtt{in}), LF(\mathtt{out}), LF(\mathtt{und}))$ to refer to LF in a concise way. For example, the triplet $(\mathtt{out}, \mathtt{und}, \mathtt{in})$ denotes the label function that maps $\mathtt{in}$ to $\mathtt{out}$, $\mathtt{out}$ to $\mathtt{und}$ and $\mathtt{und}$ to $\mathtt{in}$.

Definition 3. *An* input-output argumentation framework *(*I/O AF*) is a tuple $(\mathscr{A}, \mathscr{R}, i, o)$, where $(\mathscr{A}, \mathscr{R})$ is an argumentation framework and $i, o \in \mathscr{A}$.*

Definition 4. *Given an input-output argumentation framework $G = (\mathscr{A}, \mathscr{R}, i, o)$, with an argument $b \notin \mathscr{A}$ and a label $L \in Labs$, the* standard argumentation framework w.r.t. G and L *– denoted $F_{st}(G, L)$ – is the argumentation framework $(\mathscr{A}', \mathscr{R}')$, where $\mathscr{A}'$ and $\mathscr{R}'$ are defined through the following case distinction:*

- *If $L = \mathtt{in}$, then $\mathscr{A}' = \mathscr{A}$ and $\mathscr{R}' = \mathscr{R}$.*

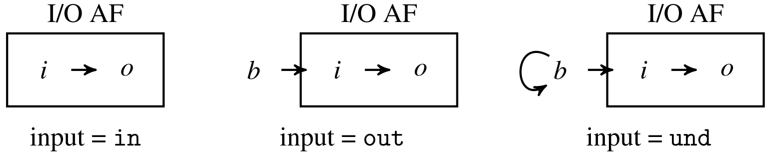


Figure 1. The three standard AFs for the I/O AF that cgp-represents the label function $(\texttt{out},\texttt{in},\texttt{und})$.

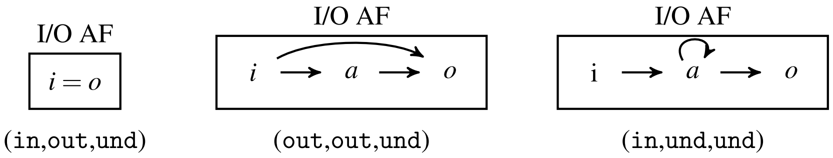


Figure 2. cgp-representation of three label functions.

- *If* $L = \texttt{out}$, *then* $\mathscr{A}' = \mathscr{A} \cup \{b\}$ *and* $\mathscr{R}' = \mathscr{R} \cup \{(b,i)\}$.
- *If* $L = \texttt{und}$, *then* $\mathscr{A}' = \mathscr{A} \cup \{b\}$ *and* $\mathscr{R}' = \mathscr{R} \cup \{(b,b),(b,i)\}$.

Definition 5. *Let σ be an argumentation semantics. An input-output argumentation framework G* represents *a label function LF w.r.t. σ iff for every $L \in Labs$, $\sigma(F_{st}(G,L)) \neq \emptyset$ and for every labeling* $\texttt{Lab} \in \sigma(F_{st}(G,L))$, $\texttt{Lab}(i) = L$ *and* $\texttt{Lab}(o) = LF(L)$.

Definition 6. *Let σ be an argumentation semantics. A label function LF is called σ-*representable *iff there is some input-output argumentation framework G that represents LF w.r.t. σ.*

In this work, we shall focus on three of the most well-known semantics, namely complete, grounded and preferred. The principles that these semantics satisfy make them the most appropriate to start with.

Definition 7. *We define* cgp *to be the set of semantics $\{$complete, grounded, preferred$\}$. If a label function can be σ-represented for every $\sigma \in$* cgp, *we say that the function is* cgp-*representable. Similarly, if a label function cannot be σ-represented for any $\sigma \in$* cgp, *we say that the function is* cgp-*unrepresentable.*

Example 1. *Consider the label function* $(\texttt{out},\texttt{in},\texttt{und})$ *which maps* in *to* out *and vice-versa, leaving* und *as it is. This function can be* cgp-*represented as depicted in Fig. 1. By having the input directly attack the output, when the input is* in, *it forces the output to be* out. *Conversely, when the input is* out, *there is no attacker of the output left, so it must be* in. *And finally when the input is* und, *the undecided label propagates to the output.*

Example 2. *Fig. 2 depicts three I/O AFs that* cgp-*represent the label functions* $(\texttt{in},\texttt{out},\texttt{und})$, $(\texttt{out},\texttt{out},\texttt{und})$ *and* $(\texttt{in},\texttt{und},\texttt{und})$ *respectively. Note that the I/O AF that represents the identity function* $(\texttt{in},\texttt{out},\texttt{und})$ *consists only of a single argument, so that the input argument i and the output argument o are the same argument.*

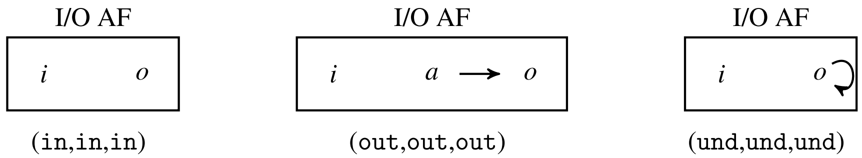


Figure 3. cgp-representation of the three constant label functions.

We now define how two input-output argumentation frameworks can be composed into a single one. The intuitive idea is that the output of the first I/O AF is used as input for the second I/O AF.

Definition 8. *Let $G_1 = (\mathscr{A}_1, \mathscr{R}_1, i_1, o_1)$ and $G_2 = (\mathscr{A}_2, \mathscr{R}_2, i_2, o_2)$ be two input-output argumentation frameworks with $\mathscr{A}_1 \cap \mathscr{A}_2 = \emptyset$, and let $c \notin \mathscr{A}_1 \cup \mathscr{A}_2$. Then we define $G_1 \oplus G_2$ to be the input-output argumentation framework $(\mathscr{A}_1 \cup \mathscr{A}_2 \cup \{c\}, \mathscr{R}_1 \cup \mathscr{R}_2 \cup \{(o_1, c)\} \cup \{(c, i_2)\}, i_1, o_2)$.*

The following theorem establishes that composed AFs represent composed label functions with respect to the complete, grounded and preferred semantics.

Theorem 1. *Let LF_1 and LF_2 be representable label functions, and let $G_1 = (\mathscr{A}_1, \mathscr{R}_1, i_1, o_1)$ and $G_2 = (\mathscr{A}_2, \mathscr{R}_2, i_2, o_2)$ be input-output argumentation frameworks that represent LF_1 and LF_2 respectively. Then $G_1 \oplus G_2$ cgp-represents $LF_2 \circ LF_1$.*

The following corollary directly follows from Theorem 1

Corollary 1. *If LF_1 and LF_2 are cgp-representable, then $LF_1 \circ LF_2$ is cgp-representable.*

3. Representability of Label Functions

In this section, we will categorize the twenty seven label functions into eleven functions that are cgp-representable and sixteen functions that are not cgp-representable.

As we will show below, a label function is cgp-representable iff it is either a constant function or maps und to und. This motivates the following definition:

Definition 9. *We define the set* Rep *as the following set of label functions:*

$$\mathtt{Rep} = \{(\mathtt{in}, \mathtt{in}, \mathtt{in}), (\mathtt{out}, \mathtt{out}, \mathtt{out})\} \cup \{(l, l', \mathtt{und}) \mid l, l' \in Labs\}$$

Theorem 2. *Every function in* Rep *is* cgp-*representable.*

The following theorem establishes that the sixteen label functions not included in Rep are actually cgp-unrepresentable.

Theorem 3. *The sixteen label functions not in* Rep *are* cgp-*unrepresentable.*

Aside from the widely used semantics included in the set cgp, the stable semantics is another well-known semantics which is also complete-based. Notice however that the stable semantics does not allow for any und arguments, and thus no framework could stable-represent a label function as defined in Def. 5, since having und as input would automatically mean there is no extension in the corresponding standard AF, so no output could be given. We can however define a similar notion over 2-valued labelings, i.e. restricting the functions to only two possible inputs and outputs: in and out.

This restriction leaves us with only four different possible label functions, and an interesting small result is that all of these are stable-representable. $(\texttt{out},\texttt{in})$ is stable-represented by the I/O AF in Figure 1 and $(\texttt{in},\texttt{out})$ by the I/O AF on the left in Figure 2. $(\texttt{in},\texttt{in})$ and $(\texttt{out},\texttt{out})$ are stable-represented by the I/O AFs in Figure 3, respectively on the left and in the middle.

Proposition 1. *The four 2-valued label functions* $(\texttt{in},\texttt{out}),(\texttt{out},\texttt{in}),(\texttt{in},\texttt{in})$ *and* $(\texttt{out},\texttt{out})$ *are all stable-representable.*

4. Impossibility of Flattening Weak Attacks

Various extensions of argumentation frameworks have been studied in the literature. One fruitful approach to studying such extensions is the flattening methodology, in which extensions of argumentation frameworks are mapped to standard argumentation frameworks through a flattening function that is faithful with respect to the semantics of the extended argumentation frameworks. Some explanations about this flattening approach and existing work applying it to argumentation frameworks with a support relation can be found in the technical report.

In this section we show how the theory of label functions can be used prove impossibility results concerning flattenings of certain extensions of argumentation frameworks, namely frameworks with a weak attack relation additionally to the standard attack relation. Note that for the formal definition of an extended framework, it is irrelevant whether the second relation that gets added to the standard attack relation is a relation of support or a second attack relation. This motivates the following definitions:

Definition 10. *A* two-relation framework *is a triple* $(\mathscr{A},\mathscr{R},\mathscr{T})$ *such that* $\mathscr{R} \subseteq \mathscr{A} \times \mathscr{A}$ *and* $\mathscr{T} \subseteq \mathscr{A} \times \mathscr{A}$*.*

Definition 11. *A* two-relation semantics *is a function* σ *that maps any two-relation framework* $B = (\mathscr{A},\mathscr{R},\mathscr{T})$ *to a set* $\sigma(B)$ *of labelings of* B*. The elements of* $\sigma(B)$ *are called* σ*-labelings of* B*.*

Definition 12. *Let* σ *be an argumentation semantics and let* σ' *be a two-relation semantics. We say that* σ' extends σ *iff for every two-relation framework* $B = (\mathscr{A},\mathscr{R},\mathscr{T})$ *with* $\mathscr{T} = \emptyset$, $\sigma'(B) = \sigma((\mathscr{A},\mathscr{R}))$*.*

We want flattenings to be defined in a local way, which we formalize as follows:

Definition 13. *Let* $B = (\mathscr{A},\mathscr{R},\mathscr{T})$ *be a two-relation framework, and let* $G = (\mathscr{A}',\mathscr{R}',i,o)$ *be an I/O AF. The* G-flattening of B *is the AF* $\mathit{flat}_G(B) = (\mathscr{A}^*,\mathscr{R}^*)$*, where* $\mathscr{A}^* := \mathscr{A} \cup \{(a,b,c) \mid (a,b) \in \mathscr{T} \text{ and } c \in \mathscr{A}' \setminus \{i,o\}\}$ *and* $\mathscr{R}^* := \mathscr{R} \cup \{((a,b,c),(a,b,c') \mid (a,b) \in \mathscr{T}, (c,c') \in \mathscr{R}' \text{ and } c,c' \notin \{i,o\}\} \cup \{(a,(a,b,c)) \mid (a,b) \in \mathscr{T} \text{ and } (i,c) \in \mathscr{R}'\} \cup \{((a,b,c),a) \mid (a,b) \in \mathscr{T} \text{ and } (c,i) \in \mathscr{R}'\} \cup \{(b,(a,b,c)) \mid (a,b) \in \mathscr{T} \text{ and } (o,c) \in \mathscr{R}'\} \cup \{((a,b,c),b) \mid (a,b) \in \mathscr{T} \text{ and } (c,o) \in \mathscr{R}'\}$.

Definition 14. *Let σ be an argumentation semantics and let σ' be a two-relation semantics that extends σ. We say that σ'* admits a uniform local flattening w.r.t. σ *iff there exists an I/O AF G such that for every two-relation argumentation framework B,* $\sigma'(B) = \sigma(\mathit{flat}_G(B))$.

We now consider a way of interpreting two-relation frameworks in which the second relation is not a support relation, but rather a *weak attack* relation. The intention behind our notion of a weak attack is that when an argument a is weakly attacked by an argument b, one can accept a without being able to defend a against the weak attack from b, but that in all other respects (such as conflict-freeness), weak attacks behave like the standard attacks of abstract argumentation, which we from now on call *strong attacks* to distinguish them clearly from weak attacks. In the labeling-based approach this can be formalized as follows (the abbreviation "s/w" stands for "strong/weak"):

Definition 15. *Let $B = (\mathscr{A}, \mathscr{R}, \mathscr{T})$ be a two-relation framework, and let* `Lab` *be a labeling of B.*

- *An argument $a \in \mathscr{A}$ is called* s/w-legally `in` w.r.t. `Lab` *iff every argument that strongly attacks a is labeled* `out` *by* `Lab` *and every argument that weakly attacks a is labeled either* `out` *or* `und`.
- *An argument $a \in \mathscr{A}$ is called* s/w-legally `out` w.r.t. `Lab` *iff some argument that strongly or weakly attacks a is labeled* `in` *by* `Lab`.
- *An argument $a \in \mathscr{A}$ is called* s/w-legally `und` w.r.t. `Lab` *iff no argument that strongly or weakly attacks a is labeled* `in` *by* `Lab` *and some argument that strongly attacks a is labeled* `und` *by* `Lab`.

Now we define the semantics for two-relation frameworks with strong and weak attacks analogously as for standard AFs:

Definition 16. *Let $B = (\mathscr{A}, \mathscr{R}, \mathscr{T})$ be a two-relation framework, and let* `Lab` *be a labeling of B.*

- `Lab` *is an* s/w-complete labeling *of B iff every argument that* `Lab` *labels* `in` *is s/w-legally* `in` *w.r.t.* `Lab`*, every argument that* `Lab` *labels* `out` *is s/w-legally* `out` *w.r.t.* `Lab`*, and every argument that* `Lab` *labels* `und` *is s/w-legally* `und` *w.r.t.* `Lab`.
- `Lab` *is an* s/w-grounded labeling *of B iff* `Lab` *is an s/w-complete labeling of B in which the set of* `in`*-labeled arguments is minimal w.r.t. set inclusion.*
- `Lab` *is an* s/w-preferred labeling *of B iff* `Lab` *is an s/w-complete labeling of B in which the set of* `in`*-labeled arguments is maximal w.r.t. set inclusion.*

One can easily see that these three semantics extend the corresponding semantics of standard AFs.

The following theorem establishes that the weak attack relation cannot be flattened to the strong attack relation in a uniform local way:

Theorem 4. *Let $\sigma \in$* `cgp`*. Then s/w-σ does not admit a uniform local flattening w.r.t. σ.*

5. Related Work

In the work of Baroni et al. [1], a similar methodology is introduced, where argumentation frameworks are partitioned, allowing for partitions to be evaluated locally. This local evaluation function needs to condition on the potential statuses of attackers from outside the partition, but does not need to consider the whole rest of the framework. From their results on decomposability of semantics, one could derive a result similar to our Theorem 1 but restricted to finite argumentation frameworks. We however chose to consider infinite argumentation frameworks as well in our work, as it grants more weight to the unrepresentability result derived in Section 3.

The work of Rienstra et al. [14] considers the partitioning of argumentation frameworks such that different semantics are applied to different partitions. In these cases, when evaluating the acceptance status of arguments within a partition, only the outside arguments which are the source of an attack targeting an argument inside that partition need to be considered, using a similar input/output methodology.

Enrichments of argumentation frameworks, such as the AFRA [3] and the BAF [9] have been interpreted in some cases using a flattening approach [7,6] which expresses higher-level relations in terms of auxiliary arguments and attacks, which can replace the original relation in a local fashion. Our results would prove useful when devising flattenings for existing or future enrichments, or showing no such flattening is possible.

6. Future Work

In future work, one could generalize the concept of a label function by dropping the requirement that the output argument always has the same label; these generalized label functions would therefore have a set of possible labels as their output value. Additionally one could drop the distinction between input argument and output argument, thus allowing an external effect on both arguments and looking at the set of label pairs that these two arguments may take over the different extensions. This would yield to a generalized theory of binary relations between arguments that have a local effect expressible in the 3-label approach. While there are only 27 label functions, the number of such different relations between arguments is 2^{36}, so the classification according to their representability is likely to be much more complex. Such a classification would allow one to extend the impossibility result from Section 4 to other enrichments of abstract argumentation frameworks, or provide insights on how to flatten new enrichments.

Another line of future work would be to investigate the representability with respect to other semantics such as semi-stable [8], stage [15], stage2 [13], CF2 [4], and the more recent SCF2 [11] and weakly complete [5]. Some preliminary findings for representability with respect to the semi-stable semantics can be found in a technical report [10].

7. Conclusion

In this paper, we formally introduce argumentation label functions, and address the question of which functions are representable with an argumentation framework, focusing on the complete, grounded and preferred semantics, for which the labeling approach has been widely studied. We provide a proof that two representations of label functions can be composed to yield the composed label function, and use this finding to categorize the twenty seven label functions into eleven label functions that are representable and sixteen that are unrepresentable with respect to these three semantics. We also briefly investigate the case of the stable semantics, which is quite straightforward since it only allows for two different labels. We then discuss how the label function approach can be used to prove an impossibility result about the flattening approach for enrichments of abstract argumentation frameworks.

References

[1] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert Van Der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Afra: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, 52(1):19–37, 2011.

[4] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1):162–210, 2005.

[5] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Revisiting the foundations of abstract argumentation–semantics based on weak admissibility and weak defense. *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (2020)*, 2020.

[6] Guido Boella, Dov M Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling I: Methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[7] Guido Boella, Dov M Gabbay, Leendert WN van der Torre, and Serena Villata. Support in Abstract Argumentation. *COMMA*, 216:111–122, 2010.

[8] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.

[9] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.

[10] Marcos Cramer and Jérémie Dauphin. Argumentation Label Functions - Technical Report. https://orbilu.uni.lu/bitstream/10993/43078/1/report.pdf. University of Luxembourg, 2020.

[11] Marcos Cramer and Leendert van der Torre. SCF2 – an argumentation semantics for rational human judgments on argument acceptability. *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI and Kognition (KIK-2019)*, 2019.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[13] Wolfgang Dvořák and Sarah Alice Gaggl. Stage semantics and the scc-recursive schema for argumentation semantics. *Journal of Logic and Computation*, 26(4):1149–1202, 2014.

[14] Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M Gabbay, and Leendert van der Torre. Multi-sorted argumentation. In *International Workshop on Theorie and Applications of Formal Argumentation*, pages 215–231. Springer, 2011.

[15] Bart Verheij. Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. In *In Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop*, pages 357–368. Universiteit, 1996.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200502

A Principle-Based Analysis of Weakly Admissible Semantics

Jeremie DAUPHIN [a] Tjitze RIENSTRA [b] and Leendert VAN DER TORRE [a,c]
[a] *University of Luxembourg*
[b] *University of Koblenz-Landau*
[c] *Zhejiang University*

Abstract. Baumann, Brewka and Ulbricht recently introduced weak admissibility as an alternative to Dung's notion of admissibility, and they use it to define weakly preferred, weakly complete and weakly grounded semantics of argumentation frameworks. In this paper we analyze their new semantics with respect to the principles discussed in the literature on abstract argumentation. Moreover, we introduce two variants of their new semantics, which we call *qualified* and *semi-qualified* semantics, and we check which principles they satisfy as well. Since the existing principles do not distinguish our new semantics from the ones of Baumann *et al.*, we also introduce some new principles to distinguish them. Besides selecting a semantics for an application, or for algorithmic design, our new principle-based analysis can also be used for the further search for weak admissibility semantics.

Keywords. Formal argumentation, abstract argumentation, principle-based analysis, weak admissibility

1. Introduction

There are three classes of abstract argumentation semantics, which can be illustrated on their behaviour on odd and even cycles in the three argumentation frameworks in Figure 1. Roughly, in Dung's admissibility-based semantics [8], the maximal extensions may contain arguments of even-length cycles but no arguments of odd-length cycles, unless the odd-length cycle is attacked by some accepted argument. For example, the set of preferred extensions of F_1 is $\{\emptyset\}$, of F_2 is $\{\{d,g\},\{e,g\}\}$, and of F_3 is $\{\emptyset\}$. In naive-based semantics like the CF2 semantics [2], the extensions typically include arguments that are only attacked by self-attacking arguments, such as the argument b in F_1 below. In addition, odd-length cycles and even-length cycles are treated similarly in the sense that

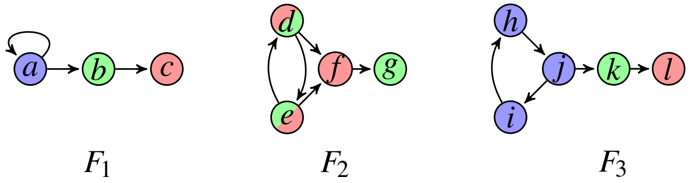


Figure 1. Three argumentation frameworks

naive extensions may also contain arguments from odd-length cycles, for example one of h, i or j in F_3. Under the *weakly admissible* semantics, recently introduced by Baumann, Brewka and Ulbricht (BBU) [3], the set of weakly preferred extensions of F_1 is $\{\{b\}\}$, of F_2 is $\{\{d,g\},\{e,g\}\}$, and of F_3 is $\{\{k\}\}$. These extensions are visualised in Figure 1: green arguments are in all the extensions, red arguments are not in the extensions and attacked by an argument in the extension (called out) and blue arguments are not in the extension and not out (called undecided). The arguments colored both red and green are in some but not all extensions.

At the moment of writing of this paper, the BBU semantics was only compared to existing semantics by their behaviour on a few examples, but a more systematic comparison was lacking. Just before sending the camera-ready version of this paper, we received a paper of the same authors [4] which will appear at a conference this year. That paper contains a table with a principle-based analysis, though most of the principles introduced and discussed in that paper are quite different from the ones in this paper, and thus that paper is complementary to this one.

The weakly admissible semantics are defined in terms of a recursive definition, which makes the analysis more difficult. Whereas many different variants of admissibility-based and naive-based semantics have been introduced and analysed, thus far only weakly complete, weakly grounded and weakly preferred semantics have been introduced from the third category. We therefore raise the following questions in this paper:

1. How do BBU's weak-admissibility based semantics compare to the existing semantics? That is, which principles does they satisfy?
2. Which other semantics can be defined along the lines of weak admissibility, giving the same results for the frameworks in Figure 1?
3. How can these new semantics be distinguished from the weak-admissibility based semantics? Which principles do these new semantics satisfy?

In general, one of the main purposes of axiomatisation in formal logic is to understand the logic with an intuitively understandable small set of principles. In proposing axioms, care should be taken to ensure that each axiom is sufficiently reasonable and sufficiently independent of others. Ideally, there should be some degree of philosophical motivation behind them. However, in the principle-based analysis of abstract argumentation, thus far the focus has been on the use of principles to differentiate semantics, and to assist computational techniques using decomposibility. Concerning the first question, Baumann *et al.* show that the weakly grounded extensions are not necessarily unique, and the principle-based analysis in this paper shows that weakly complete semantics does not satisfy directionality or SCC decomposibility.

The new semantics we define in this paper are based on SCC decomposability principles due to Baroni *et al.* [2]. This approach has previously been used to define the CF2 and Stage2 semantics. When we consider only the extensions of the framework in Figure 1, a recursive procedure comes to mind. As we show in detail later, if we use the scheme introduced by Baroni *et al.* to define CF2 (where all arguments are qualified), and we replace the base function with Dung's semantics, we get a procedure which gives the same extensions as BBU's weakly preferred semantics for the argumentation frameworks in Figure 1.

The layout of this paper is as follows. In Section 2 we introduce the *reduct admissibility* principle. We also repeat the definitions of weak admissibility and the related semantics, and we illustrate them using some new examples. In Section 3 we introduce the *semi-qualified admissibility* principle and we show that it is not satisfied by the BBU semantics. In Section 4 we introduce *weak SCC decomposability* and show it is not satisfied by the BBU semantics. In Section 5 we introduce our new two variants of semi-admissible semantics and we show which principles they satisfy. In Section 6 we discuss related and future work.

2. Weak Admissibility And The Reduct Principle

In this section we recall the definitions of the recently introduced weak-admissibility based semantics [3], and we introduce the *reduct admissibility* principle to characterise these semantics. Some notation: Given an AF $F = (A, \rightarrow)$ and $E \subseteq A$, we use E^+ to denote the set $\{b \in A \mid a \rightarrow b, a \in E\}$, use $F{\downarrow_E}$ to denote the set $(E, \rightarrow \cap E \times E)$, and use F^E to denote the *E-reduct* of F, which is the set $F{\downarrow_{E^*}}$ where $E^* = A \setminus (E \cup E^+)$. So the E-reduct of an argumentation framework F consists of the arguments that are neither in E nor attacked by E, and the attacks between these arguments.

The notion of weak admissibility weakens the requirement that every argument is defended against every attacker. Whereas an admissible extension must defend every member from every attacker, a weakly admissible extension does not require defence from attackers that do not appear in any weakly admissible set of F^E.

Definition 1. *[3] Let $F = (A, \rightarrow)$ be an AF. The set of* weakly admissible *sets of F is denoted $ad^w(F)$ and defined by $E \in ad^w(F)$ if and only if E is conflict-free (i.e., there are no $x, y \in E$ such that $x \rightarrow y$) and for every attacker y of E we have $y \notin \cup ad^w(F^E)$.*

Furthermore, a set E is said to *weakly defend* a set X if for every attacker y of X we have that either E attacks y, or y does not appear in E, does not appear in some weakly admissible set of F^E, and X is included in some weakly admissible set of F.

Definition 2. *[3] Let $F = (A, \rightarrow)$ be an AF. A set $E \subseteq A$* weakly defends *a set $X \subseteq A$ whenever, for every attacker y of X, either E attacks y, or $y \notin \cup ad^w(F^E)$, $y \notin E$ and $X \subseteq X' \in ad^w(F)$.*

Weak defense is related to weak admissibility in the sense that every conflict-free set is weakly admissible if and only if it weakly defends itself [3]. The weakly complete, preferred and grounded semantics are defined as follows.

Definition 3. *[3] Let $F = (A, \rightarrow)$ be an AF and $E \subseteq A$. We say that E is:*

- *a* weakly complete *extension of F ($E \in co^w(F)$) iff $E \in ad^w(F)$ and for every X such that $E \subseteq X$ that is w-defended by E, we have $X \subseteq E$.*
- *a* weakly preferred *extension of F ($E \in pr^w(F)$) iff E is $\subseteq$-maximal in $ad^w(F)$.*
- *a* weakly grounded *extension of F ($E \in gr^w(F)$) iff E is $\subseteq$-minimal in $co^w(F)$.*

We give some examples to illustrate these definitions.

Example 1. *The AF visualized on the left in Fig. 2 consists of a cycle of length three with one self-attacking argument. While the unique complete extension is $\emptyset$, the unique weakly complete extension is $\{a_1\}$. Intuitively, since b is self-attacking, a_1 does not need*

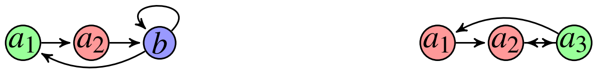


Figure 2. On the left: A self-attacking argument inside a 3-cycle, with a single weakly complete extension $\{a_1\}$. On the right: A 2-3 cycle, with only weakly complete extension $\{a_3\}$.

to be defended from b and can therefore be accepted. Now consider the AF visualized on the right in Fig. 2, consisting of a combination of a 2 cycle with a 3 cycle. While this AF has two complete extensions $\emptyset$ and $\{a_3\}$ it has one weakly complete extension $\{a_3\}$. Intuitively, since a_2 is not in any extension we have that the empty set weakly defends a_3, and so there is no reason not to accept it. The last AF we discuss is two connected 3-cycles, visualized in Fig. 3. This AF demonstrates that the weakly grounded extension

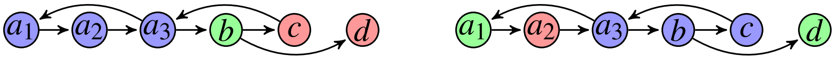


Figure 3. Two connected 3-cycles with one extra argument. $gr^w(F) = \{\{b\},\{a_1,d\}\}$.

is not unique. Since the empty set defends both $\{a_1\}$ and $\{b\}$, yet not both at the same time, both of these sets are weakly grounded.

We now define a weaker version of the admissibility principle based on the definition of ad^w, which we call reduct admissibility. The motivation for the reduct admissibility principle is taken directly from BBU's motivation of weak admissibility. We quote: "It is indeed important that a set of arguments defends itself. However, [...] isnt it sufficient to counterattack those arguments which have the slightest chance of being accepted?" [3].

Definition 4 (Reduct admissibility)**.** *We say that a semantics σ satisfies* reduct admissibility *iff for any argumentation framework $F = (A, \rightarrow)$, for every extension $E \in \sigma(F)$, we have that $\forall a \in E$, $(b,a) \in R$, we have $b \notin \bigcup \sigma(F^E)$.*

Proposition 1. *co^w, gr^w and pr^w satisfy reduct admissibility.*

Proof. For $\sigma \in \{co^w, gr^w, pr^w\}$, for all $E \in \sigma(F)$ we have that E is weakly admissible. So, for every attacker y of E, $y \notin \bigcup ad^w(F^E)$, and therefore also $y \notin \bigcup \sigma(F^E)$. □

As a principle, reduct admissibility is a bit complex due to the use of the reduct. In the following section we define an alternative principle that formalizes the same idea without referring to the reduct.

3. The Semi-qualified Admissibility Principle

We now define the *semi-qualified admissibility* principle and determine which semantics satisfy it. We then focus on some of the principles already found in the literature [9] and investigate whether the BBU semantics satisfy them.

When looking at the reduct admissibility principle, we may ask why the acceptability of an attacker is judged based on the reduct, and not on the original framework itself. For the definition of a semantics, assessing the acceptability of attackers on the reduct allows for a recursive definition that is guaranteed to terminate for finite AFs. When looking at a principle, this concern disappears and we therefore provide the definition of a different principle, which we call *semi-qualified admissibility*. Semi-qualified admissibility states that an extension only needs to defend itself against attackers that appear in at least one extension of the same framework.

Definition 5 (Semi-Qualified admissibility). *We say that a semantics σ satisfies* semi-qualified admissibility *iff for every argumentation framework $F = (A, \rightarrow)$ and every extension $E \in \sigma(F)$ we have that $\forall a \in E$, if $b \rightarrow a$ and $b \in \bigcup \sigma(F)$ then $\exists c \in E$ s.t. $c \rightarrow b$.*

Proposition 2. *co^w, gr^w and pr^w don't satisfy semi-qualified admissibility.*

Proof. Consider the AF F shown in Figure 3. Here, the set $\{a_1, d\}$ is a co^w, gr^w and pr^w extension of F. This extension is attacked by b and we also have $b \in \cup co^w(F)$ (similarly for gr^w and pr^w). However there is no $x \in \{a_1, d\}$ such that $x \rightarrow b$. □

One can easily see that our two new principles fail for CF2 and stage2 in a 3-cycle.

Proposition 3. *CF2 and stage2 don't satisfy reduct nor semi-qualified admissibility.*

The following definition introduces a number of well-known principles from the literature [9].

Definition 6. *A semantics σ satisfies the principle of:*

- admissibility *iff for every argumentation framework F, every $E \in \sigma(F)$ is conflict-free and classically defends itself in F;*
- naivety *iff for every argumentation framework F, for every $E \in \sigma(F)$, E is a $\subseteq$-maximal conflict-free set in F;*
- reinstatement *iff for every argumentation framework $F = (A, \rightarrow)$, for every $E \in \sigma(F)$ and $a \in A$ it holds that if E classically defends a then $a \in E$;*
- I-maximality *iff for every AF F, for every $E_1, E_2 \in \sigma(F)$, if $E_1 \subseteq E_2$ then $E_1 = E_2$;*
- allowing abstention *iff for every AF $F = (A, \rightarrow)$ and $a \in A$, if there exist $E_1, E_2 \in \sigma(F)$ s.t. $a \in E_1$ and $a \in E_2^+$, then there exists $E_3 \in \sigma(F)$ s.t. $a \notin E_3 \cup E_3^+$;*
- directionality *iff for every AF $F = (A, \rightarrow)$ and $S \subseteq A$ s.t. $S \cap (A \setminus S)^+ = \emptyset$, it holds that $\sigma(F\downarrow_S) = \{E \cap S \mid E \in \sigma(F)\}$.*

We now state some results regarding these principles for the semantics based on weak admissibility. Table 1 summarises our findings. Note that one can easily see that reduct and semi-qualified admissibility follow from admissibility.

Proposition 4. *co^w, gr^w and pr^w don't satisfy admissibility.*

Proof. Consider the AF $F = (\{a,b\}, \{(a,a),(a,b)\}$. We have $co^w(F) = gr^w(F) = pr^w(F) = \{\{b\}\}$, but $\{b\}$ does not classically defend itself from a. □

Proposition 5. *co^w, gr^w and pr^w don't satisfy naivety.*

Proof. Consider the AF $F = (\{a,b,c\}, \{(a,b),(b,c),(c,a)\})$. We have $co^w(F) = gr^w(F) = pr^w(F) = \{\emptyset\}$ while e.g. $\{a\}$ is conflict-free. □

	co	pr	CF2	st2	co^w	gr^w	pr^w	q-**co**	q-**gr**	q-**pr**	sq-**co**	sq-**pr**
Admissibility	✓	✓	×	×	×	×	×	×	×	×	×	×
Naivety	×	×	✓	✓	×	×	×	×	×	×	×	×
Reinst.	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓
I-Max.	×	✓	✓	✓	×	✓	✓	×	✓	✓	×	✓
Allow. abs.	✓	×	×	×	×	×	×	×	✓	×	✓	×
Directionality	✓	✓	✓	✓	×	×	?	✓	✓	✓	✓	✓
Semi-qual. adm.	✓	✓	×	×	×	×	×	×	✓	×	✓	✓
Reduct adm.	✓	✓	×	×	✓	✓	✓	×	?	?	?	?
SCC Decomp.	✓	✓	✓	✓	×	×	?	✓	✓	✓	×	×
W-SCC Decomp.	✓	✓	✓	✓	?	?	?	✓	✓	✓	✓	✓

Table 1. Principles satisfied by the weak-admissibility and qualified and semi-qualified semantics, with complete, preferred, CF2 and stage2 for comparison.

Proposition 6. *co^w, gr^w and pr^w satisfy reinstatement.*

Proof. Follows from Proposition 5.9 from Baumann *et al.* [3]. □

Proposition 7. *co^w does not satisfy I-maximality.*[1]

Proof. Consider the AF $F = (\{a,b\},\{(a,b),(b,a)\})$. $co^w = \{\{a\},\{b\},\emptyset\}$, and $\emptyset \subseteq \{a\}$ but $\emptyset \neq \{a\}$. □

Proposition 8. *gr^w and pr^w satisfy I-maximality.*

Proof. By definition, every set in pr^w is a $\subseteq$-maximal weakly admissible set, therefore none is a strict subset of the other. Similarly, every set in gr^w is by definition a $\subseteq$-minimal weakly grounded extension, and therefore none is a strict subset of another. □

Proposition 9. *co^w, gr^w and pr^w do not satisfy allowing abstention.*

Proof. Consider the AF visualized in Fig. 3. $co^w = gr^w = pr^w = \{\{a_1,d\},\{b\}\}$, and $d \in \{b\}^+$, but there is no extension E_3 where $d \notin E_3 \cup E_3^+$. □

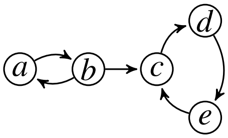


Figure 4. The weak complete and grounded semantics are not directional.

The following proposition answers two open questions of Baumann *et al.* [4].

Proposition 10. *co^w and gr^w are not directional.*

Proof. Consider the AF $F = (\{a,b,c,d,e\},\{(a,b),(b,a),(b,c),(c,d),(d,e),(e,c)\})$ visualized in Figure 4. Directionality would imply that

[1] In [4] it is mistakenly mentioned that gr^w does not satisfy I-maximality.

$$co^w(F{\downarrow_{\{a,b\}}}) = co^w(F){\downarrow_{\{a,b\}}}.$$

However we have $co^w(F{\downarrow_{\{a,b\}}}) = \{\{a\},\{b\},\emptyset\}$ and $co^w(F) = \{\{a\},\{b,d\}\}$ and hence $co^w(F){\downarrow_{\{a,b\}}} = \{\{a\},\{b\}\}$. Similarly, we have $gr^w(F{\downarrow_{\{a,b\}}}) = \{\emptyset\}$, but $gr^w(F){\downarrow_{\{a,b\}}} = \{\{a\},\{b\}\}$. □

Open Question 1. *Is the weakly preferred semantics directional? Baumann et al. [4] answer this question affirmative, but they do not provide a proof.*

4. Decomposability Principles

We now discuss two additional principles. We will use these principles as a basis for the definition of two families of semantics in the next section. The first principle is *SCC decomposability*. This principle was introduced by Baroni *et al.* under the name *full decomposability w.r.t. SCC partitioning* [1].

SCC decomposability is defined for labelling-based semantics [6]. We first need to introduce some notation. A labelling L of an AF F is a function that maps each argument of F to a label `I` (in, or accepted), `O` (out, or rejected) or `U` (undecided). We use $\mathscr{L}(F)$ to denote the set of all possible labellings of F. A labelling-based semantics σ maps each AF F to a set $\mathscr{L}_\sigma(F) \subseteq \mathscr{L}(F)$. We denote the set of SCCs (strongly connected components) of F by $\mathscr{S}(F)$. Let $F = (A, \rightarrow)$ be an AF. An *outparent* of an SCC S of F is an argument $x \in A \setminus S$ such that $x \rightarrow y$ for some $y \in S$. We denote by $OP_F(S)$ the set of outparents of S. Given a labelling $L \in \mathscr{L}(F)$ we denote by $L{\downarrow_S}$ the restriction of L to S and, given a set $X \subseteq \mathscr{L}(F)$ of labellings, denote by $X{\downarrow_S}$ the set $\{L{\downarrow_S} \mid L \in X\}$.

The SCC decomposability principle states that the set of labellings of an AF F is decomposable into the product of the sets of labellings of each SCC of F, where the set of labellings of an SCC S is a function of the labels of the outparents of S. To formalise the principle we first define the notion of *AF with input*.

Definition 7. *An* AF with input *is a tuple $(F, A_{in}, \rightarrow_{in}, L_{in})$ where $F = (A, \rightarrow)$ is an AF; A_{in} a set of* input arguments *such that $A \cap A_{in} = \emptyset$; $\rightarrow_{in} \subseteq A_{in} \times A$ is an* input attack relation*; and $L_{in} \in \mathscr{L}(A_{in})$ is an* input labelling.

A semantics is SCC decomposable if it is *represented* by some *local function*. A local function is a function f that maps each AF with input to a set of labellings. A semantics σ is represented by a local function f if the set of σ labellings of every AF F coincides with the product of the labellings of each SCC of F as determined by f.

Definition 8. *A* local function *f assigns to every AF with input $(F, A_{in}, \rightarrow_{in}, L_{in})$ a set $f(F, A_{in}, \rightarrow_{in}, L_{in}) \subseteq \mathscr{L}(F)$. We say that f* represents *the semantics σ if for every AF F,*

$$L \in \mathscr{L}_\sigma(F) \leftrightarrow \forall S \in \mathscr{S}(F), L{\downarrow}S \in f(F{\downarrow}S, OP_F(S), \rightarrow \cap OP_F(S) \times S, L{\downarrow}OP_F(S)).$$

A semantics σ is SCC decomposable *if it is represented by some local function.*

Examples of semantics that are known to be SCC decomposable are the complete, grounded and preferred semantics. We denote by $f_{\mathbf{co}}$, $f_{\mathbf{gr}}$ and $f_{\mathbf{pr}}$ the local functions representing these semantics. Their definition can be found in [1]. As for the weak-admissibility based semantics, we observe that the weak complete and weak grounded semantics are not SCC decomposable. We first define a labelling-based version of these semantics in the usual way [6]: given an AF $F = (A, \rightarrow)$, define $Ext2Lab_F : 2^A \rightarrow \mathscr{L}(F)$ by $Ext2Lab_F(E)(x) = \mathtt{I}$, if $x \in E$; $Ext2Lab_F(E)(x) = \mathtt{O}$, if $y \rightarrow x$ for some $y \in E$; and $Ext2Lab_F(E)(x) = \mathtt{U}$, otherwise. We then define the labelling-based weak complete semantics co^w by $\mathscr{L}_{co^w}(F) = \{Ext2Lab_F(E) \mid E \in co^w(F)\}$ and define the labelling-based weak preferred pr^w and grounded gr^w similarly. It then holds that

Proposition 11. *The co^w and gr^w semantics are not SCC decomposable.*

Proof. We prove it for gr^w (co^w is similar). Consider the AFs $F_1 = (\{b,c\}, \{(b,b),(b,c)\})$ and $F_2 = (\{a,b,c\}, \{(a,b),(b,a),(b,c)\})$. We then have $\mathscr{L}_{gr^w}(F_1) = \{\{a : \mathtt{U}, b : \mathtt{I}\}\}$ and $\mathscr{L}_{gr^w}(F_2) = \{\{a_1 : \mathtt{U}, a_2 : \mathtt{U}, b : \mathtt{U}\}\}$ If the gr^w semantics is SCC decomposable then there must be a local function f_{gr^w} that represents gr^w. But then $f_{gr^w}(\{c\}, \{b\}, \{(b,c)\}, \{b \colon \mathtt{U}\})$ equals both $\{\{c \colon \mathtt{I}\}\}$ and $\{\{c \colon \mathtt{U}\}\}$, which is impossible. Hence, the gr^w semantics is not SCC decomposable. □

Open Question 2. *Is the pr^w semantics SCC Decomposable?*

We now introduce a new principle called *weak SCC decomposability*. Like SCC decomposability, this principle states that the set of labellings of an AF F can be decomposed into the product of the sets of labellings of each SCC of F. The difference with SCC decomposability is that the set of labellings of an SCC S is a function not only of a particular labelling of the outparents of S, but also of the set of all other labellings that the outparents of S may receive. This provides extra information in how the labellings of an SCC are determined since, in addition to knowing the actual labels of the outparents, we also know how these arguments are labelled in other labellings. To define it we extend the notion of AF with input to that of AF with *total input* as follows.

Definition 9. *An* AF with total input *is a tuple $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ where $F, A_{in}, \rightarrow_{in}$ and L_{in} are defined as in definition 7, $S_{in} \subseteq \mathscr{L}(A_{in})$, and $L_{in} \in S_{in}$. We call S_{in} the set of* total input labellings *and $L_{in} \in S_{in}$ the* actual input labelling.

We say that σ is weakly SCC decomposable if there exists a *weak local function* (i.e., a function that maps each AF with total input to a set of labellings) that represents σ. A weak local function represents a semantics σ if the set of σ labellings of every AF F coincides with the product of the labellings of each SCC $S \in \mathscr{S}(F)$ as a determined by the weak local function.

Definition 10. *A* weak local function *g assigns to every AF with total input $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ a set $g(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in}) \subseteq \mathscr{L}(F)$. A weak local function g represents a semantics σ whenever, for every AF F, $L \in \mathscr{L}_\sigma(F)$ if and only if*

$$\forall S \in \mathscr{S}(F), L{\downarrow}S \in g(F{\downarrow}S, OP_F(S), \rightarrow \cap OP_F(S) \times S, L{\downarrow}OP_F(S), \mathscr{L}_\sigma(F){\downarrow}OP_F(S)).$$

A semantics σ is weakly SCC decomposable *if some weak local function represents σ.*

Note that SCC decomposability implies weak SCC decomposability but that the reverse does not hold. In the next section we use the weak SCC decomposability principle to define new semantics. For the weak admissibiity-based semantics we have:

Open Question 3. *Are the weakly complete, weakly grounded and weakly preferred semantics weakly SCC decomposable? Our conjecture is that they are.*

5. Qualified and Semi-Qualified Semantics

We now define two new families of semantics. They are neither admissible nor naive and they represent two new ways to deal with propagation of undecidedness. The first family are the *qualified* semantics. A qualified semantics builds on the SCC decomposability principle and is based on applying the local function of any SCC decomposable semantics with one change: in determining the labellings of an SCC S, the label $\mathtt{U}$ for an outparent x of S is treated like the label $\mathtt{O}$. This means that, if an argument x is attacked by an $\mathtt{U}$-labelled argument y, and if x and y are elements of different SCCs, then y is still qualified for acceptance (i.e., may still be labelled $\mathtt{I}$).

Definition 11. *Let σ be an SCC decomposable semantics. Let f_σ denote the local function that represents σ. We define the* qualified *σ (or q-σ) semantics as the semantics represented by the local function $f_{q\text{-}\sigma}$ defined by*

$$f_{q\text{-}\sigma}((A,\to),A_{in},\to_{in},L_{in}) = f_\sigma((A,\to),A_{in},\to_{in},L'_{in})$$

where $L'_{in}(x) = \mathtt{I}$ if $L_{in}(x) = \mathtt{I}$, and $L'_{in}(x) = \mathtt{O}$, if $L_{in}(x) = \mathtt{O}$ or $L_{in}(x) = \mathtt{U}$.

We now focus on three examples of qualified semantics, namely the qualified complete (q-**co**), qualified grounded (q-**gr**), and qualified preferred (q-**pr**) semantics. Note that, by definition, all these semantics are SCC decomposable.

Example 2. *Consider the argumentation frameworks shown in Figure 1. The AF F_1 has a unique q-**co**, q-**gr** and q-**pr** labelling, namely $\{a\colon \mathtt{U}, b\colon \mathtt{I}, c\colon \mathtt{O}\}$. The AF F_2 has three q-**co** labellings, namely $\{d\colon \mathtt{I}, e\colon \mathtt{O}, f\colon \mathtt{O}, g\colon \mathtt{I}\}$, $\{d\colon \mathtt{O}, e\colon \mathtt{I}, f\colon \mathtt{O}, g\colon \mathtt{I}\}$, and $\{d\colon \mathtt{U}, e\colon \mathtt{U}, f\colon \mathtt{I}, g\colon \mathtt{O}\}$, where the first two are also the q-**pr** labellings and the last one is also the q-**gr** labelling. The AF F_3 has a unique q-**co**, q-**gr** and q-**pr** labelling, namely $\{h\colon \mathtt{U}, i\colon \mathtt{U}, j\colon \mathtt{U}, k\colon \mathtt{I}, l\colon \mathtt{O}\}$.*

This example shows that the qualified **co**/**gr**/**pr** and weak **co**/**gr**/**pr** semantics of the three AFs in Figure 1 coincide for the AFs F_1 and F_3 but not for F_2. In F_2, the set $\{f\}$ (which corresponds to the labelling $\{d\colon \mathtt{U}, e\colon \mathtt{U}, f\colon \mathtt{I}, g\colon \mathtt{O}\}$) is not weakly admissible because it does not defend itself from d, while d does appear in some weakly admissible set of the $\{f\}$-reduct of F_2. To capture this intuition we define a second family of semantics, which builds on the weak SCC decomposability principle. It is based on applying the local function of any SCC decomposable semantics with the following change: in determining the labellings of an SCC S, the label $\mathtt{U}$ for an outparent x of S is treated like the label $\mathtt{O}$, *but only if there is no other labelling of the outparents of S where x is labelled* $\mathtt{I}$. This means that, if an argument x is attacked by an $\mathtt{U}$-labelled argument y, and if x and y are elements of different SCCs, and there is no other labelling in which y is labelled $\mathtt{I}$, then x may still be labelled $\mathtt{I}$. We call the resulting semantics *semi-qualified*.

Definition 12. *Let σ be an SCC decomposable semantics. Let f_σ denote the local function that represents σ. We define the* semi-qualified σ *(or sq-σ) semantics as the semantics represented by the weak local function $g_{sq-\sigma}$ defined by*

$$g_{sq-\sigma}((A,\rightarrow),A_{in},\rightarrow_{in},L_{in},S_{in}) = g_\sigma((A,\rightarrow),A_{in},\rightarrow_{in},L'_{in})$$

where $L'_{in}(x) = \mathtt{I}$, if $L_{in}(x) = \mathtt{I}$; $L'_{in}(x) = \mathtt{O}$, if $L_{in}(x) = \mathtt{O}$; $L'_{in}(x) = \mathtt{O}$, if $L_{in}(x) = \mathtt{U}$ and there is no $L \in S_{in}$ such that $L(x) = \mathtt{I}$; and $L'_{in}(x) = \mathtt{U}$, if $L_{in}(x) = \mathtt{U}$ and there is some $L \in S_{in}$ such that $L(x) = \mathtt{I}$.

Any semi-qualified semantics is, by definition, weakly SCC decomposable. Furthermore, note that, for a unique status semantics σ (such as the grounded semantics) the qualified σ and semi-qualified σ semantics coincide.

Example 3. *The sq-**co**, sq-**gr** and sq-**pr** labellings of the AFs F_1 and F_3 shown in Figure 1 are the same as the q-**co**, q-**gr** and q-**pr** labellings (see Example 2). The sq-**co** labellings of F_2 are different from the q-**co** labellings. The sq-**co** labellings of F_2 are $\{d\colon \mathtt{I}, e\colon \mathtt{O}, f\colon \mathtt{O}, g\colon \mathtt{I}\}$, $\{d\colon \mathtt{O}, e\colon \mathtt{I}, f\colon \mathtt{O}, g\colon \mathtt{I}\}$, and $\{d\colon \mathtt{U}, e\colon \mathtt{U}, f\colon \mathtt{U}, g\colon \mathtt{U}\}$, where the first two are also the q-**pr** labellings and the last one is also the q-**gr** labelling.*

Note that the semi-qualified labellings in the example above coincide with the weak-admissibility based extensions. Thus, they provide an alternative approach to achieve weak-admissibility like behaviour. They are not equivalent, however. In particular, the semi-qualified complete, grounded and preferred semantics are different in how they evaluate isolated SCCs. For instance, consider the AF shown in Figure 3 but without the argument d. This AF consists of a single SCC and has only one semi-qualified complete (and hence grounded and preferred) labelling in which all arguments are undecided.

Table 1 includes an overview of principles satisfied by the (semi)-qualified complete, grounded and preferred semantics. We omit the *sq*-**gr** semantics, which is equivalent to the *q*-**gr** semantics. Failure of admissibility and naivety is demonstrated by Examples 2 and 3. The same holds for failure of allowing abstention under the *q*-**pr** and *sq*-**pr** semantics and I-maximality under the *q*-**co** and *sq*-**co** semantics. Satisfaction of reinstatement under all semantics follows easily, and so does satisfaction of I-maximality under the *q*-**pr** and *q*-**pr** semantics. The *q*-**gr** semantics trivially satisfies allowing abstention and I-maximality. Non-interference and Directionality follow from weak SCC decomposability together with the property that a local function returns a non-empty set of labellings for all possible inputs, which holds for the local functions that we use. Finally, allowing abstention does not hold under the *q*-**co** semantics (see the argument b in the AF F_2 in Example 2). We now consider the remaining principles and state a number of open questions at the end.

Proposition 12. *sq-**co** satisfies allowing abstention.*

Proof. (Sketch) We show that we can transform an AF F into an AF F' such that $\mathscr{L}_{sq\text{-}\mathbf{co}}(F) = \mathscr{L}_{\mathbf{co}}(F')$. Let $S_1,\ldots,S_n$ be an ordering SCCs of F such that if a directed path from S_i to S_j exists, then $i < j$. Define A_i and $\rightarrow_i$ by $A_0 = \emptyset$, $\rightarrow_0 = \emptyset$, and for $i > 0$, $A_i = A_{i-1} \cup S_i$ and $\rightarrow_i = \rightarrow_{i-1} \cup (\rightarrow \cap (S_i \times S_i)) \cup (\rightarrow \cap (X_i \times S_i))$, where $X_i = \{x \in A_{i-1} \mid \exists L \in \mathscr{L}_{sq\text{-}\mathbf{co}}((A_{i-1},\rightarrow_{i-1})), L(x) = \mathtt{I}\}$. We then have $\mathscr{L}_{sq\text{-}\mathbf{co}}(F) = \mathscr{L}_{\mathbf{co}}((A_n,\rightarrow_n))$. Since **co** satisfies allowing abstention it thus follows that *sq*-**co** does too. □

Proposition 13. *q-**pr** does not satisfy semi-qualified admissibility.*

Proof. The AF $F = (\{a,b,c,d,e,f\},\{(a,b),(b,a),(b,c),(c,d),(d,e),(e,c),(d,f)\})$ has a q-**pr** labelling $L = \{(a,\mathtt{I}),(b,\mathtt{O}),(c,\mathtt{U}),(d,\mathtt{U}),(e,\mathtt{U}),(f,\mathtt{I})\}$, which corresponds to the extension $E = \{a,f\}$. We have $d \rightarrow E$. Therefore, according to semi-qualified admissibility, since there is no $x \in E$ such that $x \rightarrow d$ it must hold that d is not in any q-**pr** extension of F. However, this is false, because F has a s-**pr** labelling $\{(a,\mathtt{O}),(b,\mathtt{I}),(c,\mathtt{O}),(d,\mathtt{I}),(e,\mathtt{O}),(f,\mathtt{O})\}$, which corresponds to the extension $\{b,d\}$. □

Proposition 14. *q-**co** does not satisfy reduct or semi-qualified admissibility.*

Proof. Consider the AF $F = (\{a,b,c\},\{(a,b),(b,a),(b,c)\})$. This AF has a q-**co** extension $E = \{c\}$. Since $b \rightarrow E$, according to reduct admissibility, b may not be in any q-**co** extension of F^E. But $F^E = \{(\{a,b\},\{(a,b),(b,a)\})$ has a q-**co** extension $\{b\}$. This violates reduct admissibility. Semi-qualified admissibility is violated similarly. □

Proposition 15. *sq-**co**, sq-**gr**, sq-**pr** and q-**gr** satisfy semi-qualified admissibility.*

Proof. Let $F = (A,\rightarrow)$ be an AF and let $L \in \mathscr{L}_{sq\text{-}\mathbf{co}}(F)$. Let $E = \{x \in A \mid L(x) = \mathtt{I}\}$. Suppose $x \rightarrow y$ for some $y \in E$. Then either $L(x) = \mathtt{O}$, which implies that there is a z such that $z \rightarrow x$ and $x \in E$; or $L(x) = \mathtt{U}$, which implies (via Definition 12) that there is no $L' \in \mathscr{L}_{sq\text{-}\mathbf{co}}(F)$ such that $L'(x) = \mathtt{I}$ and hence no sq-**co** extension E' of F such that $x \in E'$. Hence the sq-**co** semantics, and thus also the sq-**co** and sq-**co** semantics, satisfy semi-qualified admissibility, and so does q-**gr**, which coincides with sq-**gr**. □

Open Question 4. *Do q-**pr**, q-**gr**, sq-**co** and sq-**pr** satisfy reduct admissibility?*

6. Related and Future Work

The principle-based approach was initiated by Baroni *et al.* to distinguish argumentation semantics, and then taken up by various researchers widening the scope of the "principle-based approach." For example, Doutre and colleagues have been promoting a principle-based approach to abstract argumentation, and the SESAME software [5] is an achievement in this respect. Motivated by empirical cognitive studies on argumentation semantics, Cramer and van der Torre [7] have introduced a new naive-based argumentation semantics called SCF2. A principle- based analysis shows that it has two distinguishing features:

1. If an argument is attacked by all extensions, then it can never be used in a dialogue and therefore it has no effect on the acceptance of other arguments. They call it Irrelevance of Necessarily Rejected Arguments.
2. Within each extension, if none of the attackers of an argument is accepted and the argument is not involved in a paradoxical relation, then the argument is accepted. They define paradoxicality as being part of an odd cycle, and they call this principle Strong Completeness Outside Odd Cycles.

They argue that these features together with the findings from empirical cognitive studies make SCF2 a good candidate for an argumentation semantics that corresponds well to what humans consider a rational judgment on the acceptability of arguments.

As mentioned in the introduction, just before sending the camera-ready version of this paper, we received a paper [4] with another principle-based analysis for weak admissibility, though most of the principles introduced and discussed in that paper are quite different from the ones in this paper, and thus that paper is complementary to this one.

A topic for further research is the development of a labeling-based semantics for weak admissibility, and the weakly complete, weakly grounded and weakly preferred semantics. We are also looking for labeling-based definitions of the new semantics introduced in this paper. We believe that labeling-based semantics can also be instrumental in the search for new argumentation semantics.

Acknowledgement

The work was partially supported by the German Research Foundation (DFG) project CAML (grant KE 1686/3-1).

References

[1] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert van der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[2] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005.

[3] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Revisiting the foundations of abstract argumentation - semantics based on weak admissibility and weak defense. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA, February 7 - February 12, 2020*. AAAI Press, 2020.

[4] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Comparing weak admissibility semantics to their dung-style counterparts reduct, modularization, and strong equivalence in abstract argumentation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 17th International Conference, KR 2020, Rhodes, Greece, September 12-18, 2020*, 2020 (To Appear).

[5] Philippe Besnard, Sylvie Doutre, Van Hieu Ho, and Dominique Longin. SESAME - A system for specifying semantics in abstract argumentation. In Matthias Thimm, Federico Cerutti, Hannes Strass, and Mauro Vallati, editors, *Proceedings of the First International Workshop on Systems and Algorithms for Formal Argumentation (SAFA), Potsdam, Germany, September 13, 2016*, volume 1672 of *CEUR Workshop Proceedings*, pages 40–51. CEUR-WS.org, 2016.

[6] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[7] Marcos Cramer and Leendert van der Torre. SCF2 - an argumentation semantics for rational human judgments on argument acceptability. In Christoph Beierle, Marco Ragni, Frieder Stolzenburg, and Matthias Thimm, editors, *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI & Kognition (KIK-2019), Kassel, Germany, September 23, 2019*, volume 2445 of *CEUR Workshop Proceedings*, pages 24–35. CEUR-WS.org, 2019.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[9] Leendert van der Torre and Srdjan Vesic. The Principle-Based Approach to Abstract Argumentation Semantics. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 12, pages 2735–2778. College Publications, London, 2018.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200503

Computing Strongly Admissible Sets

Wolfgang DVOŘÁK and Johannes P. WALLNER
TU Wien, Vienna, Austria

Abstract. In this work we revisit computational aspects of strongly admissible semantics in Dung's abstract argumentation frameworks. First, we complement the existing complexity analysis by focusing on the problem of computing strongly admissible sets of minimum size that contain a given argument and providing NP-hardness as well as hardness of approximation results. Based on these results, we then investigate two approaches to compute (minimum-sized) strongly admissible sets based on Answer Set Programming (ASP) and Integer Linear Programming (ILP), and provide an experimental comparison of their performance.

Keywords. abstract argumentation, strongly admissible, computational complexity, answer set programming, integer linear programming

1. Introduction

A key part to argumentative reasoning in Artificial Intelligence (AI) [3,6] are argumentation frameworks (AFs) due to Phan Minh Dung [14], which provide a formal approach to represent arguments as abstract entities together with directed conflicts (attacks) among the arguments. Semantics of AFs define criteria which sets of arguments can be deemed acceptable, where the notions of conflict-freeness and defense of arguments prove to be essential. A set of arguments is conflict-free if no arguments in the set are conflicting, and defense requires that each attack onto a set is counter-attacked from inside the set.

A particularly cautious representative of AF semantics is the grounded semantics that includes all unattacked (undoubted) arguments and each argument that can be iteratively defended from these unattacked arguments in the grounded extension. Almost all major semantics of AFs contain the arguments of the grounded extension [14]. An important reasoning task for the grounded semantics is to verify whether a queried argument is part of the grounded extension, or not. Notably, to answer this question, not all arguments within the grounded extension are necessary.

Example 1. *Consider an AF as shown in Figure 1. Say we desire to understand the acceptability of argument a under grounded semantics. The unique grounded extension of this AF is $\{a,c,g,h,j,l\}$ which answers this question positively. Yet, not all arguments are required to answer this question: e.g., argument c is sufficient to counter argument b and to defend a, and g can be used to defend argument c from its three attackers.*

A general observation from the preceding example can be made, and was formalized in the literature: one can define dialectical proof procedures, or game-theoretic notions, that specify which parts of the grounded extension suffice to witness containment in the grounded extension. Under a game-theoretic perspective a proponent only needs to con-

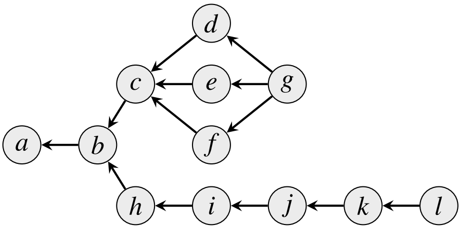


Figure 1. Two strongly admissible sets containing a: $\{a, c, g\}$ and $\{a, h, j, l\}$

sider arguments g and c to defend a against each possible counter-argument. Such game-theoretic notions for the grounded semantics were studied and resulted in the Standard Grounded Game [25,23], and the Grounded Discussion Game [8].

Importantly, so-called strongly admissible sets [4,9] turned out to be key components for winning strategies for such games. Admissible sets are conflict-free sets of arguments where each argument in the set is defended by the set. Strongly admissible sets, in contrast, require, intuitively speaking, that defense is "rooted" in unattacked arguments. In addition, strongly admissible sets were not only shown to be viable for explaining acceptance under grounded semantics, but, recently, also shown to be useful for explaining certain notions of non-acceptance [26].

Interestingly, while several papers provide results for strongly admissible sets [4, 9,5,15], in the literature strongly admissible sets were not yet studied in-depth from a computational perspective. While the grounded extension, which can be computed in polynomial time [16], would suffice to give a (maximal) strongly admissible set, explanations, such as via winning strategies for games, benefit from only requiring as few arguments as possible. Surprisingly, while all common reasoning tasks for the grounded semantics are polynomial time decidable, we show that finding a strongly admissible set of minimum size containing a queried argument is, in fact, a complex problem: we show that a natural decision variant is NP-complete. Even more, we show that approximating minimum-sized strongly admissible sets containing a queried argument remains NP-hard.

Our main contributions in this paper are as follows.

- We show NP-completeness of deciding whether there is a strongly admissible set of size at most a given integer that contains a queried argument. Moreover, we also turn some *in* P results of [9] to P-completeness results.
- We tighten the complexity landscape by showing NP-hardness for approximating strongly admissible sets of minimum size.
- We provide two computational approaches inspired by the success of the "reduction approach" to AF reasoning [12]: (a) an encoding in Answer Set Programming (ASP) and (b) an Inter Linear Programming (ILP) formulation.
- Finally we provide experiments that show feasibility of our approaches, even for large AFs, based on instances of recent competitions.

2. Argumentation Frameworks

We recall basics of argumentation frameworks (AFs) [14] and their semantics (see also [2] for an introduction).

Definition 1. An *argumentation framework (AF)* is a pair $F = (A, R)$ where A is a finite [1] set of arguments and $R \subseteq A \times A$ is the attack relation. We say that $S \subseteq A$ *attacks* b if $(a, b) \in R$ for some $a \in S$. Moreover, an argument $a \in A$ is *defended* (in F) by $S \subseteq A$ if each b with $(b, a) \in R$ is attacked by S in F.

Furthermore we denote by $S^+ = \{b \in A \mid a \in S, (a, b) \in R\}$ the set of arguments attacked by S, and by $S^- = \{b \in A \mid a \in S, (b, a) \in R\}$ the set of arguments attacking an argument in S. We call $S \cup S^+$ the *range* of S in F.

Semantics for AFs are defined as functions σ which assign to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider for σ the functions *cf*, *adm*, *com*, *grd*, and *strAdm* which stand for conflict-free, admissible, complete, grounded, and strongly admissible extensions, respectively. We first recall some semantics already introduced by Dung [14].

Definition 2. Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is *conflict-free (in F)*, if there are no $a, b \in S$, such that $(a, b) \in R$. By $cf(F)$ we denote the collection of conflict-free sets. For a conflict-free set $S \in cf(F)$, we say

- $S \in adm(F)$, if each $a \in S$ is defended by S;
- $S \in com(F)$, if $a \in S$ iff a is defended by S; and
- $S \in grd(F)$, if $S = \bigcap_{T \in com(F)} T$.

For each AF F we have $grd(F) \subseteq com(F) \subseteq adm(F) \subseteq cf(F)$ and $|grd(F)| = 1$, i.e. there is a unique grounded extension which is the $\subseteq$-minimal complete extension.

Next we introduce strongly admissible semantics as introduced by Baroni and Giacomin [4]. To this end, we first recall the notion of strong defence.

Definition 3. Let $F = (A, R)$ be an AF. An argument $a \in A$ is *strongly defended* by a set $S \subseteq A$ iff each $b \in \{a\}^-$ is attacked by some argument $c \in S \setminus \{a\}$ such that c is strongly defended by $S \setminus \{a\}$.

We are now ready to provide the definition of strongly admissible semantics.

Definition 4. Let $F = (A, R)$ be an AF. An $E \subseteq A$ is *strongly admissible* ($E \in strAdm(F)$) iff E strongly defends each of its arguments.

Caminada and Dunne [9] provide some useful characterizations of strongly admissible semantics. In particular, one characterization avoids the notion of strong defence but recursively refers to smaller strongly admissible sets.

Proposition 1 ([9])**.** *Let $F = (A, R)$ be an AF. It holds that $E \in strAdm(F)$ iff each $a \in E$ is defended by some strongly admissible set $S \subseteq E \setminus \{a\}$.*

Another useful characterization is based on the well-known (restricted) characteristic function of AFs, recalled next.

Definition 5. Given an AF $F = (A, R)$, the *characteristic function* $\mathscr{F}_F : 2^A \to 2^A$ of F is defined as $\mathscr{F}_F(S) = \{x \in A \mid S \text{ defends } x\}$. We will also consider the characteristic function restricted to a given set $E \subseteq A$: $\mathscr{F}_{F,E}(S) = \{a \in E \mid S \text{ defends } a\}$.

[1] Notice that the original definition is not limited to finite frameworks, but as we are studying computational properties we are only concerned with finite AFs.

By [14] it holds that the grounded extension of an AF F is the least fixed-point of the characteristic function $\mathscr{F}_F$. Caminada and Dunne [9] use the restricted variant of the characteristic function to characterize strongly admissible sets.

Proposition 2 ([9])**.** *Let $F = (A,R)$ be an AF. We have $E \in strAdm(F)$ iff E is the least fixed-point of $\mathscr{F}_{F,E}(.)$.*

We next recall useful properties of strongly admissible semantics [4,9]. For each AF F we have $grd(F) \subseteq strAdm(F) \subseteq adm(F) \subseteq cf(F)$. Moreover, $strAdm(F)$ forms a lattice, with the grounded extension as the top element and the empty set as the bottom element. That is, the grounded extension acts as the top element of the $strAdm(F)$-lattice as well as the bottom element of $com(F)$-semi-lattice. This yields the observation that $grd(F) = strAdm(F) \cap com(F)$, which we will use later on.

Lemma 1. *Let $F = (A,R)$ be an AF. It holds that $S \in grd(F)$ iff $S \in strAdm(F) \cap com(F)$.*

Proof. First, by the above the grounded extension is both strongly admissible and complete. Next, recall that the grounded extension is the unique minimal complete extension and the unique maximal strongly admissible set. That is, each strongly admissible set different from the grounded extension is not complete and each complete extension different from the grounded extension is not strongly admissible. □

3. Complexity Results

In this section we recap existing complexity result for strong admissibility and complement them by P-hardness results as well as by studying the problem of computing a minimum sized strongly admissible set for a given argument.

3.1. Standard Reasoning Problems

The standard problems in abstract argumentation (cf. [16]) are: Credulous acceptance $Cred_\sigma$, deciding whether a given argument is in at least one σ-extension; Skeptical acceptance $Skept_\sigma$, deciding whether a given argument is in all σ-extensions; Verification Ver_σ, deciding whether a given set of arguments is a σ-extension; and Non-emptiness $Exists_\sigma^{\neg\emptyset}$, deciding whether the AF has a non-empty σ-extension.

First, credulous reasoning with strongly admissible semantics corresponds to credulous reasoning with grounded semantics [9] and is thus P-complete [17]. Moreover, as the empty-set is always strongly admissible no argument is skeptically accepted and the problem becomes trivial. The Non-emptiness problem again corresponds to the respective problem for grounded semantics and is in L. Next, as shown in [9], verifying a strongly admissible set is in P and we next show that it is also P-hard by relating it to verifying the grounded extension.

Lemma 2. *Verifying whether a given set is strongly admissible is* P*-complete.*

Proof. As shown in [9] we can verify a strongly admissible set E in P by computing the least fixed-point of$\mathscr{F}_{F,E}(.)$. We know that verifying the grounded extension is P-complete [17] and verifying a complete extension is in L [16]. A logspace reduction from verifying the grounded extension G of an AF F to verifying a strongly admissible set E of an AF F' simply tests whether G is complete. If not it returns a no instance (e.g., the AF $F' = (\{a\}, \{(a,a)\})$ and $E = \{a\}$), otherwise it returns the unmodified AF and extension, i.e., $F' = F$ and $E = G$. Then, by Lemma 1, it holds that G is the grounded extension of F iff E is strongly admissible in F'. □

3.2. Minimum size strongly admissible sets

Arguably, the standard reasoning problems fail to fully characterize the complexity of strongly admissible semantics as both credulous and skeptical acceptance can be solved without referring to the strongly admissible sets of the AF, i.e., only the empty-set and the grounded extension are used. In that light, and motivated by the usage of strongly admissible sets as justifications in grounded discussion games [9] we are interested in the problem of computing a minimum size strongly admissible set containing a given argument. The decision version of this problem is the k-Witness problem $k\text{-}Witness_\sigma$, deciding whether a given argument is in at least one σ-extension of size at most k. We remark that k is part of the input of this problem, but we keep the "k" in the problem name to emphasize the size constraint. We next show that $k\text{-}Witness_{strAdm}$ is NP-complete, which implies that there is no polynomial time algorithm that computes a minimum size strongly admissible set (unless P = NP).

Theorem 1. *$k\text{-}Witness_{strAdm}$ is NP-complete.*

Proof. For membership, non-deterministically construct a subset of the arguments and verify whether this set (i) contains the queried argument, (ii) contains at most k many arguments (for a given integer k), and (iii) is strongly admissible in the given AF. The last check can be done in polynomial time (see Lemma 2).

For hardness, we reduce from the NP-complete problem of deciding whether a given Boolean formula is satisfiable. Given a Boolean formula $\varphi = c_1 \wedge \cdots \wedge c_n$ in conjunctive normal form (CNF) over variables X with clause set C, construct AF $F_\varphi = (A, R)$ with $A = X \cup \overline{X} \cup C \cup D \cup \{\varphi\}$ and

$$\begin{aligned} R = & \{(c_i, \varphi) \mid c_i \in C\} \cup \{(d_x, \varphi) \mid d_x \in D\} \cup \\ & \{(x, c_i) \mid x \in c_i\} \cup \{(\bar{x}, c_i) \mid \neg x \in c_i\} \cup \\ & \{(x, d_x), (\bar{x}, d_x) \mid x \in X\} \end{aligned}$$

with $D = \{d_x \mid x \in X\}$. It follows that F_φ can be constructed in polynomial time for a given φ. An illustration of the reduction for an example formula is shown in Figure 2. We claim that φ is satisfiable iff there is an $E \in strAdm(F_\varphi)$ with (i) $\varphi \in E$ and (ii) $|E| \leq |X| + 1$. First assume that φ is satisfiable and let M be a model of φ. Consider the set $E = M \cup \{\bar{x} \in \bar{X} \mid x \notin M\} \cup \{\varphi\}$ of arguments. Clearly $|E| = |X| + 1$ and it remains to show that $E \in strAdm(F_\varphi)$. First we have $E \setminus \{\varphi\} \subseteq \mathscr{F}_{F,E}(\emptyset)$ as these arguments are not attacked at all. Moreover, by the assumption that M is a model it follows that all c_i and d_x are attacked by $E \setminus \{\varphi\}$ and thus φ is defended and thus $E = \mathscr{F}^2_{F,E}(\emptyset)$. We obtain that $E \in strAdm(F_\varphi)$.

Now assume $E \in strAdm(F_\varphi)$ with (i) $\varphi \in E$ and (ii) $|E| \leq |X| + 1$. In particular $E \in adm(F_\varphi)$. As $\varphi \in E$ we have $C \cup D \subseteq E^+$. By the arguments D we have for each

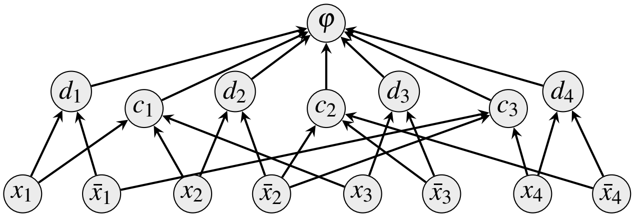


Figure 2. Example reduction for $\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_4)$.

$x \in X$ either $x \in E$ or $\bar{x} \in E$ and by the size constraint that not both of them are in E. As all C are attacked by E we obtain that $M = E \cap X$ is a model of φ. □

We summarize the complexity of all decision problems in Table 1.

Table 1. Computational complexity of strong admissibility

$Cred_{strAdm}$	$Skept_{strAdm}$	Ver_{strAdm}	$Exists^{\neg\emptyset}_{strAdm}$	$k\text{-}Witness_{strAdm}$
P-c	trivial	P-c	in L	NP-c

Given that we cannot compute a strongly admissible set of minimum size in polynomial time a standard approach would be to go for a strongly admissible sets whose size is a good approximation of the minimum size. We say a set S is an approximation within a factor α if we have $|S| \leq \alpha \cdot |opt|$ where opt is an optimal solution. An α-approximation algorithm is then a polynomial time algorithm that always returns a solution that is within a factor α.

In order to show that hardness even holds when approximating a strongly admissible set of minimum size with a queried argument, i.e., that under complexity theoretic assumptions there cannot be a c-approximation algorithm for this problem for any constant c, we consider the SET COVER problem. In the following we use $[n]$ as shorthand for the set $\{1, 2, \ldots, n\}$ (for a positive integer n).

Definition 6 (SET COVER)**.** Given a universe $\mathtt{U} = [n]$ and a collection $\mathtt{S} = \{S_1, \ldots, S_m\}$ with $S_i \subseteq \mathtt{U}$, the SET COVER problem is to find a smallest set $I \subseteq [m]$ such that $\bigcup_{i \in I} S_i = \mathtt{U}$.

Notice that SET COVER is not a decision problem as we are interested in computing (the size of) a cardinality minimum solution. For SET COVER it is well-known that there is no α-approximation algorithm where α is a constant unless P = NP. The actual lower bound for approximation algorithms is even stronger.

Proposition 3 ([13])**.** *Approximating* SET COVER *within a factor* $(1 - \varepsilon) \cdot \ln(n)$ *is* NP-*hard for every* $\varepsilon > 0$.

We next present a reduction from SET COVER to computing a minimum size strongly admissible set for a given argument.

Reduction 1. *For an instance* $(\mathtt{U}, \mathtt{S})$ *of* SET COVER *we define the AF* $F_{\mathtt{U},\mathtt{S}} = (A, R)$ *with* $A = \mathtt{U} \cup \mathtt{S} \cup \{t\}$ *and* $R = \{(i,t) \mid i \in \mathtt{U}\} \cup \{(S,i) \mid S \in \mathtt{S}, i \in S\}$.

An example instance of this reduction is shown in Figure 3. We next show that this reduction maintains the size of minimum solutions.

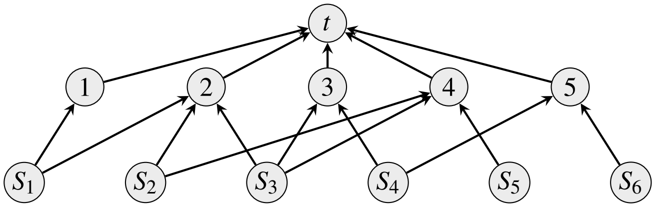


Figure 3. $F_{\mathtt{U},\mathtt{S}}$ with $\mathtt{U} = \{1,2,3,4,5\}$ and $\mathtt{S} = \{\{1,2\},\{2,4\},\{2,3,4\},\{3,5\},\{4\},\{5\}\}$.

Lemma 3. *Let $I_{\min}$ be a minimum set cover of* $\mathtt{U},\mathtt{S}$ *and E a minimum among the sets in $strAdm(F_{\mathtt{U},\mathtt{S}})$ containing t then $|I_{\min}| + 1 = |E|$.*

Proof. We show that there is a one-to-one correspondence between set covers and strongly admissible sets containing t which maintains the size of the solutions. First consider a set cover $I \subseteq [m]$. It is easy to verify that the set $E = \{S_i \mid i \in I\} \cup \{t\}$ is a strongly admissible set in $F_{\mathtt{U},\mathtt{S}}$ and the $|I| + 1 = |E|$, as by assumption the selected S_i attack all arguments in $\mathtt{U}$. Now consider $E \in strAdm(F_{\mathtt{U},\mathtt{S}})$ with $t \in E$ and define $I = \{i \in [m] \mid S_i \in E\}$. First as $t \in E$ we have $\mathtt{U} \cap E = \emptyset$. We thus have $\{S_i | i \in I\} = E \setminus \{t\} \subseteq [m]$ and $|I| + 1 = |E|$. Finally, as the arguments $\mathtt{U}$ are only attacked by arguments $\mathtt{S}$ we have that each $i \in \mathtt{U}$ is contained in some $S \in E \cap \mathtt{S}$ and thus I is a set cover. □

By Lemma 3, each c-approximation algorithm for computing a minimum size strongly admissible set would yield a $(2c)$-approximation[2] for SET COVER, which is in contradiction to Proposition 3.

Theorem 2. *Computing a c-approximation for the minimum size of a strongly admissible set for a given argument is* NP*-hard for every $c \geq 1$.*

Let us complete this section on the computational complexity with some final remarks. First, notice that Theorem 2 implies Theorem 1. We believe that the hardness proof in terms of the standard reduction is of additional value (e.g., when comparing with other semantics) and thus included both reductions. Second, while we focused on minimizing the size of the strongly admissible set, all the results can be easily extended to minimizing the number of attackers of a strongly admissible set or to minimizing a (weighted) combination of the size and the number of attackers. Finally, notice that the AFs constructed in the reductions have a rather simple graph structure, i.e., they are acyclic, bipartite and all paths are of length at most 2.

4. Two Reduction-based Implementations

As our complexity analysis shows NP-hardness for computing strongly admissible sets of minimum size, we implement computation of strongly admissible sets via using answer set programming (ASP) and integer linear programming (ILP), two approaches that showed promise for NP-hard problems in computational argumentation [11,12]. Both approaches make use of the characterization as least fixed point of the characteristic function from Proposition 2.

[2] Assume there is a c-approximation, i.e., a strongly admissible set E' with $|E'| \leq c \cdot |E|$. Then, by Lemma 3, there is set cover I with $|I| = |E'| - 1$ and thus we have $|I| = |E'| - 1 \leq c \cdot |E| - 1 = c \cdot (|I_{\min}| + 1) - 1 \leq 2c \cdot |I_{\min}|$.

4.1. Answer Set Programming Encodings

Background on ASP. We recall briefly ASP background [24,21]. We fix a countable set U of constants. An atom is an expression $p(t_1,\ldots,t_n)$, where p is a predicate of arity $n \geq 0$ and each term t_i is either a variable or an element from U. An atom is ground if it is free of variables. BU denotes the set of all ground atoms over U. A rule r is of the form

$$a \leftarrow b_1,\ldots,b_k,\ \texttt{not}\ b_{k+1},\ldots,\ \texttt{not}\ b_m.$$

with $m \geq k \geq 0$, where $a, b_1,\ldots,b_m$ are atoms, and "not" stands for default negation. The head of r is a and the body of r is $body(r) = \{b_1,\ldots,b_k,\ \texttt{not}\ b_{k+1},\ldots,\ \texttt{not}\ b_m\}$. Furthermore, $body^+(r) = \{b_1,\ldots,b_k\}$ and $body^-(r) = \{b_{k+1},\ldots,b_m\}$. A rule r is ground if r does not contain variables. A program is a finite set of rules. If each rule in a program is ground, we call the program ground.

For any program π, let UP be the set of all constants appearing in π. Define GP as the set of rules r_τ obtained by applying, to each rule $r \in \pi$, all possible substitutions τ from the variables in r to elements of UP. An interpretation $I \subseteq BU$ satisfies a ground rule r iff the head a of r is in I whenever $body^+(r) \subseteq I$ and $body^-(r) \cap I = \emptyset$. I satisfies a ground program π, if each $r \in \pi$ is satisfied by I. A non-ground rule r (resp., a program π) is satisfied by an interpretation I iff I satisfies all groundings of r (resp., GP). An interpretation $I \subseteq BU$ is an answer set of π if it is a subset-minimal set satisfying the Gelfond-Lifschitz reduct $\pi^I = \{head(r) \leftarrow body^+(r) \mid I \cap body^-(r) = \emptyset, r \in GP\}$.

ASP Encoding. As usual [18], we encode an AF $F = (A,R)$ as ASP facts $\{\textbf{arg}(x) \mid x \in A\}$ and $\{\textbf{att}(x,y) \mid (x,y) \in R\}$. We provide our ASP encoding for strongly admissible semantics in Listing 1. The first two lines generate a potential answer set for each subset E of the arguments, where the atoms with the **in** predicate contain the arguments in E. Lines 3 & 4 compute the least fixed-point of $\mathscr{F}_{F,E}(.)$, notice that in Line 3 we explicitly ensure that only arguments a with $\textbf{in}(a)$ can be in the fixed-point. The conditional $\textbf{defeated}(Y)$: $\textbf{att}(Y,X)$ stands for a conjunction (list) of all $\textbf{defeated}(Y)$ s.t. $\textbf{att}(Y,X)$ holds (i.e., the conditional is expanded to $\{\textbf{defeated}(y) \mid (y,x) \in R\}$). Finally, in Line 5 we rule out answer sets where the least fixed-point differs from the guessed set E. With the encoding in Listing 1 we can use clingo [20] to compute all strongly admissible

Listing 1: Encoding π_{strAdm}

```
in(X) ← arg(X), not out(X).
out(X)← arg(X), not in(X).
fixedPoint(X) ← in(X), defeated(Y) : att(Y,X).
defeated(X)← arg(X), fixedPoint(Y), att(Y,X).
← in(X), not fixedPoint(X).
```

sets of an AF and to solve all the standard reasoning tasks. Moreover, clingo also provides flexible optimization statements. To compute an optimal strongly admissible set that contains some argument t we first add a constraint "$\leftarrow$ not $\textbf{in}(t)$." to ensure that the computed set contains the argument and then add optimization constrains. To compute a minimum size set we add the constraint "`#minimize {1@1,X:in(X)}.`" which for each argument in the extension adds one to the objective function that is minimized.

If we want to take also the attackers of an extension into account we first add a rule "`attacker(X):- in(Y), att(X,Y).`" that computes the attacking arguments and then we can formulate minimize statements that also take attackers into account. For example such statements can minimize the size of the set plus the number of attackers(`#minimize {1@1,X:attacker(X)}. #minimize {1@1,X:in(X)}.`), among the minimum size sets minimize the number of attackers (`#minimize {1@2,X:in(X)}. #minimize {1@1,X:attacker(X)}.`), or weight between size and the number of attackers, e.g., by adding two for each argument in the set but just one for attackers (`#minimize {2@1,X:in(X)}. #minimize {1@1,X:attacker(X)}.`).

The encodings are available at `https://www.dbai.tuwien.ac.at/research/argumentation/aspartix/dung/min_extensions.html`.

4.2. Encoding as Integer Linear Programming

We describe an encoding of our problem as an Integer Linear Program (ILP) (see, e.g., [27]). Integer linear programming is a well-known NP-hard problem where one is given variables over the integer domain, a linear objective function and linear constraints, and one has to minimize (or maximize) the objective while satisfying the constraints.

In contrast to the ASP encoding we will require a quadratic number of variables and as preliminary tests showed that solvers are sensitive to the number of variables we implemented the following simplifications before encoding the problem as ILP. First we ignore all arguments that cannot reach the query argument t, second compute the grounded extension G of the simplified AF, and then give an encoding that only refers to arguments in G and G^-. Moreover, as our encoding mimics the (restricted) characteristic function we are also interested in the maximal number of iterations k until a fixed point is reached. We obtain that the number of iterations is at most $\min(|G|,|G^-|+1)$ as in each iteration we have to add an additional argument to G and attack an additional argument in G^-, otherwise we have reached a fixed-point.

Given an AF $F = (A,R)$, the grounded extension G, the attackers G^-, $k = \min(|G|,|G^-|+1)$, w_a, w_b coefficients to weight between $|E|$ and $|E|^-$, and a target argument $t \in A$ we define variables with domain $\{0,1\}$: $x_{i,\ell}$ encoding that argument $i \in G$ is accepted in the ℓ-th iteration of the fixed-point computation; and y_i encoding that $i \in G^-$ is an argument that attacks E. The ILP is then given as follows:

$$\min \quad w_a \cdot \sum_{i \in G} x_{i,\ell} + w_b \cdot \sum_{i \in G^-} y_i \tag{1}$$

$$x_{i,\ell} \leq x_{i,\ell+1} \qquad \forall i \in G, 1 \leq \ell < k \tag{2}$$

$$x_{j,\ell} \leq \sum_{(k,i) \in R} x_{k,\ell-1} \qquad \forall (i,j) \in R, 2 \leq \ell \leq k \tag{3}$$

$$x_{j,1} \leq 0 \qquad \forall (i,j) \in R \tag{4}$$

$$x_{j,k} \leq y_i \qquad \forall (i,j) \in R \tag{5}$$

$$x_t = 1 \tag{6}$$

In the objective function (1) we can use the parameters to specify if we want to minimize the arguments in the extension ($w_a = 1, w_b = 0$), the number of attackers of the extension ($w_a = 0, w_b = 1$), or the sum of both ($w_a = w_b = 1$). The constraint (2) ensures that if an argument is accepted in the ℓ-th iteration then it is also accepted in all the later iterations. By constraint (3) we get that an argument is accepted in the ℓ-th iteration only if it is defended by the arguments accepted at the $(\ell - 1)$-th iteration. With the exception of the first iteration where constraint (4) ensures that only unattacked arguments are accepted. Constraint (5) encodes that all arguments i that attack an accepted argument are marked as attackers of E. Finally, constraint (6) ensures that the computed strongly admissible set contains the query argument t.

5. Experimental Evaluation

We provide an empirical evaluation of our two reduction-based approaches to implement strongly admissible sets in ASP and ILP. We focus on the task of finding one strongly admissible set of minimum size that contains a queried argument for a given AF, for which we showed NP-hardness.

For instances, we considered AFs and queries provided by the benchmark sets of the two most recent argumentation competitions ICCMA'17 [19] and ICCMA'19[3]. From ICCMA'19 we considered all provided AFs and queries, and from ICCMA'17 we considered the AFs and queries from the "A" benchmark set. From these benchmark sets, we included in our experiments all AFs and queries whenever a query was provided by the competition. Additionally, for each AF we generated one query argument within the grounded extension of the AF (whenever the grounded is not empty). This resulted in 326 AFs from ICCMA'19 and 333 AFs from ICCMA'17 (17 AFs from ICCMA'17 included no query argument and have an empty grounded extension). Furthermore, to look at scalability, we generated 22 new AFs from the admbuster class [7], which is specifically designed for strongly admissible sets, with sizes from $10{,}000$ to $7{,}000{,}000$ arguments. For queries of the newly generated AFs, we included again one randomly chosen argument from the grounded extension, and also the distinguished argument "a", which requires the whole grounded extension to be included (i.e., the only strongly admissible set containing "a" is the grounded extension). Overall, we included 698 AFs and 1168 queries over these AFs.

We let clingo [20] v5.4.0 and IBM's CPLEX [1] v12.10.0.0 compute a strongly admissible set of minimum size containing the queried argument with a timeout limit of 900 seconds and a memory limit of 8GB per query. All experiments were run on a machine with two AMD Opteron Processors 6308, 12 x 16GB RAM, and Debian 8. For using CPLEX, we used the python LP modeler PuLP [4] to generate the ILP constraints.

We summarize the results obtained. Using clingo and the above encoding, 1157 instances were solved optimally (550) or clingo reported unsatisfiability (607). One timeout was encountered and ten times the memory limit was reached (for instances with at least $3{,}000{,}000$ arguments). Using CPLEX, overall 1089 instances were solved, either by reporting an optimal strongly admissible set (482) or by showing unsatisfiability (607). Further, using CPLEX two timeouts were reported and 76 times the memory

[3] `https://www.iccma2019.dmi.unipg.it/`

[4] https://pypi.org/project/PuLP/

Table 2. Summary of performance evaluation

approach	# optima found	# unsatisfiability reported	# timeouts	# memory limit reached
ASP	550	607	1	10
ILP	482	607	2	77

limit was reached. One time a memory error was reported. Considering the running times clingo solved 75% of the instances within 1.6 sec while CPLEX solved 75% of the instances within 2.5 sec. When considering the admbuster instances, clingo solved all instances up to $2,000,000$ arguments while CPLEX only solved some of the instances up to $20,000$ arguments. In Table 2 we summarize the results obtained (the memory error is included in the memory limit reached column).

From the results one can conclude that a large portion of the instances could be solved (optimally), even when faced with large and potentially complex AFs. Due to the low number of timeouts, we hypothesize that memory was the main limiting factor, for the instances considered. Both reduction-based approaches reported the same unsatisfiable instances, which plausibly seems to be a simple case: computing the grounded extension and checking inclusion is a poly-time decidable problem. While our approach utilizing CPLEX reported a higher number of cases where the memory limit was exceeded, we speculate that this is more inherent to the large number of constraints produced during construction of the ILP rather than due to (limitations of) CPLEX itself. More efficient constructions of constraints might lead to better performance. Nevertheless, both approaches solved a majority of the instances.

6. Conclusions

In this paper we studied the computational properties of strongly admissible sets. Concretely, we showed NP-hardness of finding a minimum-sized strongly admissible set containing a queried argument, a hardness result that we showed also to hold when approximating strongly admissible sets. To overcome the clear theoretic complexity barrier, we provided two approaches to compute strongly admissible sets in practice: based on the promising approaches of ASP and ILP, we provided one implementation each, with both of them showing good performance in our experiments. The implementation based on ASP was somewhat outperforming the approach based on ILP.

Directions for future work include extending our approaches to minimal admissible sets, which are also relevant for discussion-games [10], and abstract argumentation formalisms that enhance Dung AFs [22].

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) through project P30168-N31 and the Vienna Science and Technology Fund (WWTF) through project ICT19-065.

References

[1] IBM ILOG: CPLEX optimizer 12.10.0.0, 2020. Webpage at IBM: `https://www.ibm.com/analytics/cplex-optimizer`.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*. College Publications, 2018.

[4] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

[5] Ringo Baumann, Thomas Linsbichler, and Stefan Woltran. Verifiability of argumentation semantics. In *Proc. COMMA*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 83–94. IOS Press, 2016.

[6] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[7] Martin Caminada. Strong admissibility revisited. In *Proc. COMMA*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 197–208. IOS Press, 2014.

[8] Martin Caminada. A discussion game for grounded semantics. In *Proc. TAFA, Revised Selected Papers*, volume 9524 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2015.

[9] Martin Caminada and Paul E. Dunne. Strong admissibility revisited: Theory and applications. *Argument & Computation*, 10(3):277–300, 2019.

[10] Martin W. A. Caminada, Wolfgang Dvořák, and Srdjan Vesic. Preferred semantics as socratic discussion. *J. Log. Comput.*, 26(4):1257–1292, 2016.

[11] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. In *Handbook of Formal Argumentation*, chapter 15, pages 688–767. College Publications, 2018.

[12] Günther Charwat, Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – A survey. *Artif. Intell.*, 220:28–63, 2015.

[13] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proc. STOC*, pages 624–633. ACM, 2014.

[14] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[15] Paul E. Dunne. Characterizing strongly admissible sets. *Argument & Computation*, 2020. Accepted manuscript available at `http://dx.doi.org/10.3233/AAC-200483`.

[16] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*, chapter 13, pages 631–688. College Publications, 2018.

[17] Wolfgang Dvořák and Stefan Woltran. On the intertranslatability of argumentation semantics. *J. Artif. Intell. Res.*, 41:445–475, 2011.

[18] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[19] Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. *Artif. Intell.*, 279, 2020.

[20] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Patrick Lühne, Philipp Obermeier, Max Ostrowski, Javier Romero, Torsten Schaub, Sebastian Schellhorn, and Philipp Wanko. The Potsdam answer set solving collection 5.0. *KI*, 32(2-3):181–182, 2018.

[21] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, pages 1070–1080. MIT Press, 1988.

[22] Atefeh Keshavarzi Zafarghandi, Rineke Verbrugge, and Bart Verheij. Discussion games for preferred semantics of abstract dialectical frameworks. In *Proc. ECSQARU*, volume 11726 of *Lecture Notes in Computer Science*, pages 62–73. Springer, 2019.

[23] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[24] Ilkka Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Ann. Math. Artif. Intell.*, 25(3-4):241–273, 1999.

[25] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1):25–75, 1997.

[26] Zeynep G. Saribatur, Johannes P. Wallner, and Stefan Woltran. Explaining non-acceptability in abstract argumentation. In *Proc. ECAI*, 2020. Accepted for publication.

[27] Gerard Sierksma and Yori Zwols. *Linear and integer optimization: theory and practice*. CRC Press, 2015.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200504

Expressiveness of SETAFs and Support-Free ADFs Under 3-Valued Semantics

Wolfgang DVOŘÁK [a], Atefeh KESHAVARZI ZAFARGHANDI [b] and Stefan WOLTRAN [a]

[a] *Institute of Logic and Computation, TU Wien, Austria*
[b] *Department of Artificial Intelligence, Bernoulli Institute, University of Groningen, The Netherlands*

Abstract. Generalizing the attack structure in argumentation frameworks (AFs) has been studied in different ways. Most prominently, the binary attack relation of Dung frameworks has been extended to the notion of collective attacks. The resulting formalism is often termed SETAFs. Another approach is provided via abstract dialectical frameworks (ADFs), where acceptance conditions specify the relation between arguments; restricting these conditions naturally allows for so-called support-free ADFs. The aim of the paper is to shed light on the relation between these two different approaches. To this end, we investigate and compare the expressiveness of SETAFs and support-free ADFs under the lens of 3-valued semantics. Our results show that it is only the presence of unsatisfiable acceptance conditions in support-free ADFs that discriminate the two approaches.

Keywords. Abstract argumentation frameworks, Abstract dialectical frameworks, Collective attack.

1. Introduction

Abstract argumentation frameworks (AFs) as introduced by Dung [1] are a core formalism in formal argumentation. A popular line of research investigates extensions of Dung AFs that allow for a richer syntax (see, e.g. [2]). In this work we investigate two generalisations of Dung AFs that allow for a more flexible attack structure (but do not consider support between arguments).

The first formalism we consider are SETAFs as introduced by Nielsen and Parsons [3]. SETAFs extend Dung AFs by allowing for collective attacks such that a set of arguments B attacks another argument a but no proper subset of B attacks a. Argumentation frameworks with collective attacks have received increasing interest in the last years. For instance, semi-stable, stage, ideal, and eager semantics have been adapted to SETAFs in [4,5]; translations between SETAFs and other abstract argumentation formalisms are studied in [6]; [7] observed that for particular instantiations, SETAFs provide a more convenient target formalism than Dung AFs. The expressiveness of SETAFs with two-valued semantics has been investigated in [4] in terms of signatures. Signatures have been introduced in [8] for AFs. In general terms, a signature for a formalism and a semantics captures all possible outcomes that can be obtained by the instances of the formalism under the considered semantics. Besides that, signatures are recognized as crucial for operators in dynamics of argumentation (cf. [9]).

The second formalism we consider are support-free abstract dialectical frameworks (SFADFs), a subclass of abstract dialectical frameworks (ADFs) [10] which are known as an advanced abstract formalism for argumentation, that is able to cover several generalizations of AFs [2,6]. This is accomplished by acceptance conditions which specify, for each argument, its relation to its neighbour arguments via propositional formulas. These conditions determine the links between the arguments which can be, in particular, attacking or supporting. SFADFs are ADFs where each link between arguments is attacking; they have been introduced in a recent study on different sub-classes of ADFs [11].

For comparison of the two formalisms, we need to focus on 3-valued (labelling) semantics [12,13], which are integral for ADF semantics [10]. In terms of SETAFs, we can rely on the recently introduced labelling semantics in [5]. We first define a new class of ADFs (SETADFs) where the acceptance conditions strictly follow the nature of collective attacks in SETAFs and show that SETAFs and SETADFs coincide for the main semantics, i.e. the σ-labellings of a SETAF are equal to the σ-interpretations of the corresponding SETADF. We then provide exact characterisations of the 3-valued signatures for SETAFs (and thus for SETADFs) for most of the semantics under consideration. While SETADFs are a syntactically defined subclass of ADFs, the second formalism we study can be understood as semantical subclass of ADFs. In fact, for SFADFs it is not the syntactic structure of acceptance conditions that is restricted but their semantic behavior, in the sense that all links need to be attacking. The second main contribution of the paper is to determine the exact difference in expressiveness between SETADFs and SFADFs.

We briefly discuss related work. The expressiveness of SETAFs has first been investigated in [14] where different sub-classes of ADFs, i.e. AFs, SETAFs and Bipolar ADFs, are related w.r.t. their signatures of 3-valued semantics. Moreover, they provide an algorithm to decide realizability in one of the formalisms under different semantics. However, no explicit characterisations of the signatures are given. Recently, Pührer [15] presented explicit characterisations of the signatures of general ADFs (but not for the sub-classes discussed above). In contrast, [4] provides explicit characterisations of the two-valued signatures of SETAFs and shows that SETAFs are more expressive than AFs. In both works all arguments are relevant for the signature, while in [5] it is shown that when allowing to add extra arguments to an AF which are not relevant for the signature, i.e. the extensions/labellings are projected on common arguments, then SETAFs and AFs are of equivalent expressiveness. Other recent work [16] already implicitly showed that SFADFs with satisfiable acceptance conditions can be equivalently represented as SETAFs. This provides a sufficient condition for rewriting an ADF as SETAF and raises the question whether it is also a necessary condition. In fact, we will show that a SFADF has an equivalent SETAF if and only if all acceptance conditions are satisfiable. Different sub-classes of ADFs (including SFADFs) have been compared in [11], but no exact characterisations of signatures as we provide here are given in that work.
To summarize, the main contributions of our paper are as follows:

- We embed SETAFs under 3-valued labeling based semantics [5] in the more general framework of ADFs. That is, we show 3-valued labeling based SETAF semantics to be equivalent to the corresponding ADF semantics. As a side result, this also shows the equivalence of the 3-valued SETAF semantics in [14] and [5].

- We investigate the expressiveness of SETAFs under 3-valued semantics by providing exact characterizations of the signatures for preferred, stable, grounded and conflict-free semantics, thus complementing the investigations on expressiveness of SETAFs [4] in terms of extension-based semantics.
- We study the relations between SETAFs and support-free ADFs (SFADFs). In particular we give the exact difference in expressiveness between SETAFs and SFADFs under conflict-free, admissible, preferred, grounded, complete, stable and two-valued model semantics.

Some technical details had to be omitted but are available in an online appendix: https://www.dbai.tuwien.ac.at/research/argumentation/comma2020-1.pdf

2. Background

In this section we briefly recall the necessary definitions for SETAFs and ADFs.

Definition 1. A set argumentation framework (SETAF) is an ordered pair $F = (A,R)$, where A is a finite set of arguments and $R \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the attack relation.

The semantics of SETAFs are usually defined similarly to AFs, i.e., based on extensions. However, in this work we focus on 3-valued labelling based semantics, cf. [5].

Definition 2. A (3-valued) labelling of a SETAF $F = (A,R)$ is a total function $\lambda : A \mapsto \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$. For $x \in \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ we write λ_x to denote the sets of arguments $a \in A$ with $\lambda(a) = x$. We sometimes denote labellings λ as triples $(\lambda_{\mathtt{in}}, \lambda_{\mathtt{out}}, \lambda_{\mathtt{undec}})$.

Definition 3. Let $F = (A,R)$ be a SETAF. A labelling is called conflict-free in F if (i) for all $(S,a) \in R$ either $\lambda(a) \neq \mathtt{in}$ or there is a $b \in S$ with $\lambda(b) \neq \mathtt{in}$, and (ii) for all $a \in A$, if $\lambda(a) = \mathtt{out}$ then there is an attack $(S,a) \in R$ such that $\lambda(b) = \mathtt{in}$ for all $b \in S$. A labelling λ which is conflict-free in F is

- *admissible* in F iff for all $a \in A$ if $\lambda(a) = \mathtt{in}$ then for all $(S,a) \in R$ there is a $b \in S$ such that $\lambda(b) = \mathtt{out}$;
- *complete* in F iff for all $a \in A$ (i) $\lambda(a) = \mathtt{in}$ iff for all $(S,a) \in R$ there is a $b \in S$ such that $\lambda(b) = \mathtt{out}$, and (ii) $\lambda(a) = \mathtt{out}$ iff there is an attack $(S,a) \in R$ such that $\lambda(b) = \mathtt{in}$ for all $b \in S$;
- *grounded* in F iff it is complete and there is no λ' with $\lambda'_{\mathtt{in}} \subset \lambda_{\mathtt{in}}$ complete in F;
- *preferred* in F iff it is complete and there is no λ' with $\lambda'_{\mathtt{in}} \supset \lambda_{\mathtt{in}}$ complete in F;
- *stable* in F iff $\lambda_{\mathtt{undec}} = \emptyset$.

The set of all σ labellings for a SETAF F is denoted by $\sigma_{\mathscr{L}}(F)$, where $\sigma \in \{cf, adm, com, grd, prf, stb\}$ abbreviates the different semantics in the obvious manner.

Example 1. The SETAF $F = (\{a,b,c\}, \{(\{a,b\},c), (\{a,c\},b)\})$ is depicted in Figure 1. For instance, $(\{a,b\},c) \in R$ says that there is a joint attack from a and b to c. This represents that neither a nor b is strong enough to attack c by themselves. Further, $\{a \mapsto \mathtt{in}, b \mapsto \mathtt{undec}, c \mapsto \mathtt{in}\}$ is an instance of a conflict-free labelling, that is not an admissible labelling (since c is mapped to $\mathtt{in}$ but neither a nor b is mapped to $\mathtt{out}$). The labelling that maps all argument to $\mathtt{undec}$ is not a complete labelling, how-

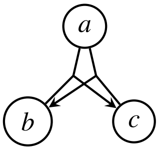


Figure 1. The SETAF of Example 1.

ever, it is an admissible labelling. Further, $\{a \mapsto \texttt{in}, b \mapsto \texttt{undec}, c \mapsto \texttt{undec}\}$ is an admissible, the unique grounded and a complete labelling, which is not a preferred labelling because $\lambda_{\texttt{in}} = \{a\}$ is not $\subseteq$-maximal among all complete labellings. Moreover, $prf_{\mathscr{L}}(F) = stb_{\mathscr{L}}(F) = \{\{a \mapsto \texttt{in}, b \mapsto \texttt{out}, c \mapsto \texttt{in}\}, \{a \mapsto \texttt{in}, b \mapsto \texttt{in}, c \mapsto \texttt{out}\}\}$.

We next turn to abstract dialectical frameworks [17].

Definition 4. An abstract dialectical framework (ADF) is a tuple $D = (S, L, C)$ where:

- S is a finite set of arguments (statements, positions);
- $L \subseteq S \times S$ is a set of links among arguments;
- $C = \{\varphi_s\}_{s \in S}$ is a collection of propositional formulas over arguments, called acceptance conditions.

An ADF can be represented by a graph in which nodes indicate arguments and links show the relation among arguments. Each argument s in an ADF is attached by a propositional formula, called acceptance condition, φ_s over $par(s)$ such that, $par(s) = \{b \mid (b, s) \in L\}$. Since in ADFs an argument appears in the acceptance condition of an argument s if and only if it belongs to the set $par(s)$, the set of links L of an ADF is given implicitly via the acceptance conditions. The acceptance condition of each argument clarifies under which condition the argument can be accepted and determines the type of links (see Definition 6 below). An *interpretation* v (for F) is a function $v : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, that maps arguments to one of the three truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$). Truth values can be ordered via information ordering relation $<_i$ given by $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair of truth values are related by $<_i$. Relation $\leq_i$ is the reflexive and transitive closure of $<_i$. An interpretation v is *two-valued* if it maps each argument to either $\mathbf{t}$ or $\mathbf{f}$. Let $\mathscr{V}$ be the set of all interpretations for an ADF D. Then, we call a subset of all interpretations of the ADF, $\mathbb{V} \subseteq \mathscr{V}$, an *interpretation-set*. Interpretations can be ordered via $\leq_i$ with respect to their information content, i.e. $w \leq_i v$ if $w(s) \leq_i v(s)$ for each $s \in S$. Further, we denote the update of an interpretation v with a truth value $x \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ for an argument b by $v|^b_x$, i.e. $v|^b_x(b) = x$ and $v|^b_x(a) = v(a)$ for $a \neq b$. Finally, the partial valuation of acceptance condition φ_s by v, is given by $\varphi^v_s = v(\varphi_s) = \varphi_s[p/\top : v(p) = \mathbf{t}][p/\bot : v(p) = \mathbf{f}]$, for $p \in par(s)$.

Semantics for ADFs can be defined via a *characteristic operator* Γ_D for an ADF D. Given an interpretation v (for D), the characteristic operator Γ_D for D is defined as

$$\Gamma_D(v) = v' \text{ such that } v'(s) = \begin{cases} \mathbf{t} & \text{if } \varphi^v_s \text{ is irrefutable (i.e., a tautology),} \\ \mathbf{f} & \text{if } \varphi^v_s \text{ is unsatisfiable,} \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

Definition 5. Given an ADF $D = (S, L, C)$, an interpretation v is

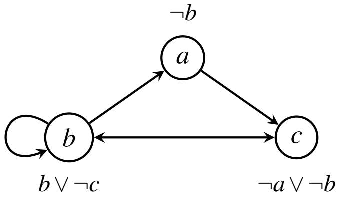


Figure 2. The ADF of Example 2.

- *conflict-free* in D iff $v(s) = \mathbf{t}$ implies φ_s^v is satisfiable and $v(s) = \mathbf{f}$ implies φ_s^v is unsatisfiable;
- *admissible* in D iff $v \leq_i \Gamma_D(v)$;
- *complete* in D iff $v = \Gamma_D(v)$;
- *grounded* in D iff v is the least fixed-point of Γ_D;
- *preferred* in D iff v is $\leq_i$-maximal admissible in D;
- a *(two-valued) model* of D iff v is two-valued and for all $s \in S$, it holds that $v(s) = v(\varphi_s)$;
- a *stable model* of D if v is a model of D and $v^{\mathbf{t}} = w^{\mathbf{t}}$, where w is the grounded interpretation of the *stb*-reduct $D^v = (S^v, L^v, C^v)$, where $S^v = v^{\mathbf{t}}$, $L^v = L \cap (S^v \times S^v)$, and $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ for each $s \in S^v$.

The set of all σ interpretations for an ADF D is denoted by $\sigma(D)$, where $\sigma \in \{cf, adm, com, grd, prf, mod, stb\}$ abbreviates the different semantics in the obvious manner.

Example 2. An example of an ADF $D = (S, L, C)$ is shown in Figure 2. To each argument a propositional formula is associated, the acceptance condition of the argument. For instance, the acceptance condition of c, namely $\varphi_c : \neg a \vee \neg b$, states that c can be accepted in an interpretation where either a or b (or both) are rejected.

In D the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$ is conflict-free. However, v is not an admissible interpretation, because $\Gamma_D(v) = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$, that is, $v \not\leq_i \Gamma_D(v)$. The interpretation $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{u}\}$ on the other hand is an admissible interpretation. Since $\Gamma_D(v_1) = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ and $v_1 \leq_i \Gamma_D(v_1)$. Further, $prf(D) = mod(D) = \{\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}\}$, but only the first interpretation in this set is a stable model. This is because for $v = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}$ the unique grounded interpretation w of D^v is $\{a \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ and $v^{\mathbf{t}} = w^{\mathbf{t}}$. The interpretation $v' = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}\}$ is not a stable model, since the unique grounded interpretation w' of $D^{v'}$ is $\{b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$ and $v'^{\mathbf{t}} \neq w'^{\mathbf{t}}$. Actually, v' is not a stable model because the truth value of b in v' is since of self-support. Moreover, the unique grounded interpretation of D is $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$. In addition, we have $com(D) = prf(D) \cup grd(D)$.

In ADFs links between arguments can be classified into four types, reflecting the relationship of attack and/or support that exists among the arguments. In Definition 6 we consider two-valued interpretations that are only defined over the parents of a, that is, only give values to $par(a)$.

Definition 6. Let $D = (S, L, C)$ be an ADF. A link $(b, a) \in L$ is called

- *supporting* (in D) if for every two-valued interpretation v of $par(a)$, $v(\varphi_a) = \mathbf{t}$ implies $v|_{\mathbf{t}}^{b}(\varphi_a) = \mathbf{t}$;

- *attacking* (in D) if for every two-valued interpretation v of $par(a)$, $v(\varphi_a) = \mathbf{f}$ implies $v|^b_\mathbf{t}(\varphi_a) = \mathbf{f}$;
- *redundant* (in D) if it is both attacking and supporting;
- *dependent* (in D) if it is neither attacking nor supporting.

The classification of the types of the links of ADFs is also relevant for classifying ADFs themselves. One particularly important subclass of ADFs is that of *bipolar* ADFs or BADFs for short. In such an ADF each link is either attacking or supporting (or both; thus, the links can also be redundant). Another subclass of ADFs, having only attacking links, is defined in [18], called *support free ADFs* (SFADFs) in the current work, defined formally as follows.

Definition 7. An ADF is called support-free if it has only attacking links.

For SFADFs, it turns out that the intention of stable semantics, i.e. to avoid cyclic support among arguments, becomes immaterial, thus $mod(D) = stb(D)$ for any ADF D; the property is called weakly coherent in [18].

Proposition 1. *For every SFADF D it holds that $mod(D) = stb(D)$.*

Proof. The result follows from the following observation: Let $D = (S, L, C)$ be an ADF, let v be a model of D and let $s \in S$ be an argument such that all parents of s are attackers. Thus, φ^v_s is irrefutable if and only if $\varphi_s[p/\bot \,:\, v(p) = \mathbf{f}]$ is irrefutable. □

3. Embedding SETAFs in ADFs

As observed by Polberg [19] and Linsbichler et.al [14], the notion of collective attacks can also be represented in ADFs by using the right acceptance conditions. We next introduce the class SETADFs of ADFs for this purpose.

Definition 8. An ADF $D = (S, L, C)$ is called SETAF-like (SETADF) if each of the acceptance conditions in C is given by a formula (with $\mathscr{C}$ a set of non-empty clauses)

$$\bigwedge_{cl \in \mathscr{C}} \bigvee_{a \in cl} \neg a.$$

That is, in a SETADF each acceptance condition is either $\top$ (if $\mathscr{C}$ is empty) or a proper CNF formula over negative literals. SETADFs and SETAFs can be embedded in each other as follows.

Definition 9. Let $F = (A, R)$ be a SETAF. The ADF associated to F is a tuple $D_F = (S, L, C)$ in which $S = A$, $L = \{(a, b) \mid (B, b) \in R, a \in B\}$ and $C = \{\varphi_a\}_{a \in S}$ is the collection of acceptance conditions defined, for each $a \in S$, as

$$\varphi_a = \bigwedge_{(B,a) \in R} \bigvee_{a' \in B} \neg a'.$$

Let $D = (S, L, C)$ be a SETADF. We construct the SETAF $F_D = (A, R)$ in which, $A = S$, and R is constructed as follows. For each argument $s \in S$ with acceptance formula $\bigwedge_{cl \in \mathscr{C}} \bigvee_{a \in cl} \neg a$ we add the attacks $\{(cl, s) \mid cl \in \mathscr{C}\}$ to R.

Clearly the ADF D_F associated to a SETAF F is a SETADF and D is the ADF associated to the constructed SETAF F_D. We next deal with the fact that SETAF semantics are defined as three-valued labellings while semantics for ADFs are defined as three valued interpretations. In order to compare these semantics we associate the *in* label with t, the *out* label with f, and the *undec* label with u.

Theorem 2. *For $\sigma \in \{cf, adm, com, prf, grd, stb\}$, a SETAF F and its associated SETADF D, we have that $\sigma_{\mathscr{L}}(F)$ and $\sigma(D)$ are in one-to-one correspondence with each labelling $\mathbb{L} \in \sigma_{\mathscr{L}}(F)$ corresponding to an interpretation $v \in \sigma(D)$ such that $v(s) = \mathbf{t}$ iff $\lambda(s) = \mathtt{in}$, $v(s) = \mathbf{f}$ iff $\lambda(s) = \mathtt{out}$, and $v(s) = \mathbf{u}$ iff $\lambda(s) = \mathtt{undec}$.*

Notice that by the above theorem we have that the 3-valued SETAF semantics introduced in [14] coincide with the 3-valued labelling based SETAF semantics of [5] and the model semantics of [14] corresponds to the stable semantics of [5].

4. 3-valued Signatures of SETAFs

We adapt the concept of signatures [8] towards our needs first.

Definition 10. The signature of SETAFs under a labelling-based semantics $\sigma_{\mathscr{L}}$ is defined as $\Sigma_{SETAF}^{\sigma_{\mathscr{L}}} = \{\sigma_{\mathscr{L}}(F) \mid F \in SETAF\}$. The signature of an ADF-subclass $\mathscr{C}$ under a semantics σ is defined as $\Sigma_{\mathscr{C}}^{\sigma} = \{\sigma(D) \mid D \in \mathscr{C}\}$.

By Theorem 2 we can use labellings of SETAFs and interpretations of the SETADF class of ADFs interchangeably, yielding that $\Sigma_{SETAF}^{\sigma_{\mathscr{L}}} \equiv \Sigma_{SETADF}^{\sigma}$, i.e. the 3-valued signatures of SETAFs and SETADFs only differ in the naming of the labels. For convenience, we will use the SETAF terminology in this section.

Proposition 3. *The signature $\Sigma_{SETAF}^{stb_{\mathscr{L}}}$ is given by all sets $\mathbb{L}$ of labellings such that*

1. *all $\lambda \in \mathbb{L}$ have the same domain $\textsc{Args}_{\mathbb{L}}$; $\lambda(s) \neq \mathtt{undec}$ for all $\lambda \in \mathbb{L}$, $s \in \textsc{Args}_{\mathbb{L}}$.*
2. *If $\lambda \in \mathbb{L}$ assigns one argument to $\mathtt{out}$ then it also assigns an argument to $\mathtt{in}$.*
3. *For arbitrary $\lambda_1, \lambda_2 \in \mathbb{L}$ with $\lambda_1 \neq \lambda_2$ there is an argument a such that $\lambda_1(a) = \mathtt{in}$ and $\lambda_2(a) = \mathtt{out}$.*

Proof. We first show that for each SETAF F the set $stb_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. First clearly all $\lambda \in stb_{\mathscr{L}}(F)$ have the same domain and by the definition of stable semantics do not assign $\mathtt{undec}$ to any argument. That is the first condition is satisfied. For Condition (2), towards a contradiction assume that the domain is non-empty and $\lambda \in stb_{\mathscr{L}}(F)$ assigns all arguments to $\mathtt{out}$. Consider an arbitrary argument a. By definition of stable semantics a is only labeled $\mathtt{out}$ if there is an attack (B, a) such that all arguments in B are labeled in $\mathtt{in}$, a contradiction. Thus we obtain that there is at least one argument a with $\lambda(a) = \mathtt{in}$. For Condition (3), towards a contradiction assume that for all arguments a with $\lambda_1(a) = \mathtt{in}$ also $\lambda_2(a) = \mathtt{in}$ holds. As $\lambda_1 \neq \lambda_2$ there is an a with $\lambda_2(a) = \mathtt{in}$ and $\lambda_1(a) = \mathtt{out}$. That is, there is an attack (B, a) such that $\lambda_1(b) = \mathtt{in}$ for all $b \in B$. But then also $\lambda_2(b) = \mathtt{in}$ for all $b \in B$ and by $\lambda_2(a) = \mathtt{in}$ we obtain that $\lambda_2 \notin cf_{\mathscr{L}}(F)$, a contradiction.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $A_{\mathbb{L}} = \textsc{Args}_{\mathbb{L}}$ and $R_{\mathbb{L}} = \{(\lambda_{\mathtt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{out}\}$. We show that $stb_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$.

To this end we first show $stb_{\mathcal{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$. Consider an arbitrary $\lambda \in \mathbb{L}$: By Condition (1) there is no $a \in \textsc{Args}_{\mathbb{L}}$ with $\lambda(a) = \mathtt{undec}$ and it only remains to show $\lambda \in cf_{\mathcal{L}}(F_{\mathbb{L}})$. First, if $\lambda(a) = \mathtt{out}$ for some argument a then by construction of $R_{\mathbb{L}}$ and Condition (2) we have an attack $(\lambda_{\mathtt{in}}, a)$ and thus a is legally labeled $\mathtt{out}$. Now towards a contradiction assume there is a conflict (B,a) such that $B \cup \{a\} \subseteq \lambda_{\mathtt{in}}$. Then, by construction of $R_{\mathbb{L}}$ there is a $\lambda' \in \mathbb{L}$ with $\lambda'_{\mathtt{in}} = B$ and $\lambda_{\mathtt{in}} \neq B$ (as $a \in \lambda_{\mathtt{in}}$). That is, $\lambda'_{\mathtt{in}} \subset \lambda_{\mathtt{in}}$, a contradiction to Condition (3). Thus, $\lambda \in cf_{\mathcal{L}}(F_{\mathbb{L}})$ and therefore $\lambda \in stb_{\mathcal{L}}(F_{\mathbb{L}})$.

To show $stb_{\mathcal{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$, consider $\lambda \in stb_{\mathcal{L}}(F_{\mathbb{L}})$. If λ maps all arguments to $\mathtt{in}$ then there is no attack in $R_{\mathbb{L}}$ which means that $\mathbb{L}$ contains only the labelling λ. Thus, we assume that there is a with $\lambda(a) = \mathtt{out}$ and there is $(B,a) \in R_{\mathbb{L}}$ with $B \subseteq \lambda_{\mathtt{in}}$. By construction there is $\lambda' \in \mathbb{L}$ such that $\lambda'_{\mathtt{in}} = B$. Then by construction we have $(B,c) \in R_{\mathbb{L}}$ for all $c \notin B$ and thus $\lambda'_{\mathtt{in}} = B = \lambda_{\mathtt{in}}$ and moreover $\lambda'_{\mathtt{out}} = \lambda_{\mathtt{out}}$ and thus $\lambda = \lambda'$. □

We now turn to the signature for preferred semantics. Compared to the conditions for stable semantics, labelling may now assign $\mathtt{undec}$ to arguments. Note that stable is the only semantics allowing for an empty labelling set.

Proposition 4. *The signature $\Sigma^{prf_{\mathcal{L}}}_{SETAF}$ is given by all non-empty sets $\mathbb{L}$ of labellings s.t.*

1. *all labellings $\lambda \in \mathbb{L}$ have the same domain $\textsc{Args}_{\mathbb{L}}$.*
2. *If $\lambda \in \mathbb{L}$ assigns one argument to $\mathtt{out}$ then it also assigns an argument to $\mathtt{in}$.*
3. *For arbitrary $\lambda_1, \lambda_2 \in \mathbb{L}$ with $\lambda_1 \neq \lambda_2$ there is an argument a such $\lambda_1(a) = \mathtt{in}$ and $\lambda_2(a) = \mathtt{out}$.*

Proof sketch. We first show that for each SETAF F the set $prf_{\mathcal{L}}(F)$ satisfies the conditions of the proposition. The first condition is satisfied as all $\lambda \in prf_{\mathcal{L}}(F)$ have the same domain. The second condition is satisfied by the definition of conflict-free labellings. Condition (3) is by the $\subseteq$-maximality of $\lambda_{\mathtt{in}}$ which implies that there is a conflict between each two preferred extensions.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $A_{\mathbb{L}} = \textsc{Args}_{\mathbb{L}}$ and $R_{\mathbb{L}} = \{(\lambda_{\mathtt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{out}\} \cup \{(\lambda_{\mathtt{in}} \cup \{a\}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{undec}\}$. It remains to show that $prf_{\mathcal{L}}(F_{\mathbb{L}}) = \mathbb{L}$. To show $prf_{\mathcal{L}}(F_{\mathbb{L}}) \supseteq \mathbb{L}$, consider an arbitrary $\lambda \in \mathbb{L}$. $\lambda \in cf_{\mathcal{L}}(F_{\mathbb{L}})$ can be seen by construction, and $\lambda \in adm_{\mathcal{L}}(F_{\mathbb{L}})$ since argument labelled out is attacked by λ; finally $\lambda \in prf_{\mathcal{L}}(F_{\mathbb{L}})$ is guaranteed since the arguments a with $\lambda(a) = \mathtt{undec}$ are involved in self-attacks. To show $prf_{\mathcal{L}}(F_{\mathbb{L}}) \subseteq \mathbb{L}$ consider $\lambda \in prf_{\mathcal{L}}(F_{\mathbb{L}})$. It can be checked that λ satisfies all the conditions of the proposition. □

Proposition 5. *The signature $\Sigma^{cf_{\mathcal{L}}}_{SETAF}$ is given by all non-empty sets $\mathbb{L}$ of labellings s.t.*

1. *all $\lambda \in \mathbb{L}$ have the same domain $\textsc{Args}_{\mathbb{L}}$.*
2. *If $\lambda \in \mathbb{L}$ assigns one argument to $\mathtt{out}$ then it also assigns an argument to $\mathtt{in}$.*
3. *For $\lambda \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{in}}$ also $(C, \emptyset, \textsc{Args}_{\mathbb{L}} \setminus C) \in \mathbb{L}$.*
4. *For $\lambda \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ also $(\lambda_{\mathtt{in}}, \lambda_{\mathtt{out}} \setminus C, \lambda_{\mathtt{undec}} \cup C) \in \mathbb{L}$.*
5. *For $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ also $(\lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}}) \in \mathbb{L}$.*
6. *For $\lambda, \lambda' \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ (s.t. $C \neq \emptyset$) we have $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'_{\mathtt{in}}$.*

Proof sketch. Let F be an arbitrary SETAF we show that $cf_{\mathcal{L}}(F)$ satisfies the conditions of the proposition. The first two conditions are clearly satisfied by the definition of conflict-free labelling. For Condition (3), towards a contradiction assume that $(C, \emptyset, \textsc{Args}_{\mathbb{L}} \setminus C)$ is not conflict-free. Then there is an attack (B,a) such that $B \cup \{a\} \subseteq C \subseteq \lambda_{\mathtt{in}}$, and thus $\lambda \notin \mathit{cf}_{\mathscr{L}}(F)$, a contradiction. Condition (4) is satisfied as in the definition of conflict-free labellings there are no conditions for labeling an argument $\mathtt{undec}$. Further, the conditions that allow to label an argument $\mathtt{out}$ solely depend on the $\mathtt{in}$ labeled arguments. For Condition (5), consider $\lambda, \lambda' \in \mathit{cf}_{\mathscr{L}}(F)$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ and $\lambda^* = (\lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}})$. Since $\lambda, \lambda' \in \mathbb{L}$, it is easy to check that λ^* is a well-founded labelling and $\lambda^* \in \mathit{cf}_{\mathscr{L}}(F)$. For Condition (6), consider $\lambda, \lambda' \in \mathit{cf}_{\mathscr{L}}(F)$ and a set $C \subseteq \lambda_{\mathtt{out}}$ containing an argument a such that $\lambda(a) = \mathtt{out}$. That is, there is an attack (B,a) with $B \subseteq \lambda_{\mathtt{in}}$ and thus $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'_{\mathtt{in}}$. That is, Condition (6) is satisfied.

Now assume that $\mathbb{L}$ satisfies all the conditions. We give a SETAF $F_{\mathbb{L}} = (A_{\mathbb{L}}, R_{\mathbb{L}})$ with $A_{\mathbb{L}} = \textsc{Args}_{\mathbb{L}}$ and $R_{\mathbb{L}} = \{(\lambda_{\mathtt{in}}, a) \mid \lambda \in \mathbb{L}, \lambda(a) = \mathtt{out}\} \cup \{(B,b) \mid b \in B, \nexists \lambda \in \mathbb{L} : \lambda_{\mathtt{in}} = B\}$. To complete the proof it remains to show that $\mathit{cf}_{\mathscr{L}}(F_{\mathbb{L}}) = \mathbb{L}$. □

Finally, we give an exact characterisation of the signature of grounded semantics.

Proposition 6. *The signature $\Sigma^{grd_{\mathscr{L}}}_{SETAF}$ is given by sets $\mathbb{L}$ of labellings such that $|\mathbb{L}| = 1$, and if $\lambda \in \mathbb{L}$ assigns one argument to* $\mathtt{out}$ *then $\lambda_{\mathtt{in}} \neq \emptyset$.*

Notice that Proposition 6 basically exploits that grounded semantics is a unique status semantics based on admissibility. The result thus immediately extends to other semantics satisfying these two properties, e.g. to ideal or eager semantics [5].

So far, we have provided characterisations for the signatures $\Sigma^{stb_{\mathscr{L}}}_{\text{SETAF}}$, $\Sigma^{prf_{\mathscr{L}}}_{\text{SETAF}}$, $\Sigma^{cf_{\mathscr{L}}}_{\text{SETAF}}$, $\Sigma^{grd_{\mathscr{L}}}_{\text{SETAF}}$. By Theorem 2 we get analogous characterizations of $\Sigma^{\sigma}_{\text{SETADF}}$ for the corresponding ADF semantics.

We have not yet touched admissible and complete semantics. Here, the exact characterisations seem to be more cumbersome and are left for future work. However, for admissible semantics the following proposition provides necessary conditions for an labelling-set to be *adm*-realizable, but it remains open whether they are also sufficient.

Proposition 7. *For each $\mathbb{L} \in \Sigma^{adm_{\mathscr{L}}}_{SETAF}$ we have:*

1. *all $\lambda \in \mathbb{L}$ have the same domain* $\textsc{Args}_{\mathbb{L}}$.
2. *If $\lambda \in \mathbb{L}$ assigns one argument to* $\mathtt{out}$ *then it also assigns an argument to* $\mathtt{in}$.
3. *For $\lambda, \lambda' \in \mathbb{L}$ and $C \subseteq \lambda_{\mathtt{out}}$ (s.t. $C \neq \emptyset$) we have $\lambda_{\mathtt{in}} \cup C \not\subseteq \lambda'_{\mathtt{in}}$.*
4. *For arbitrary $\lambda, \lambda' \in \mathbb{L}$ either (a) $(\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}}) \in \mathbb{L}$ or (b) there is an argument a such $\lambda(a) = \mathtt{in}$ and $\lambda'(a) = \mathtt{out}$.*
5. *For $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{out}} \subseteq \lambda'_{\mathtt{out}}$, and $C \subseteq \lambda_{\mathtt{in}} \setminus \bigcup_{\lambda^* \in \mathbb{L}:\, \lambda^*_{\mathtt{in}} = \lambda'_{\mathtt{in}}} \lambda^*_{\mathtt{out}}$ we have $(\lambda'_{\mathtt{in}} \cup C, \lambda'_{\mathtt{out}}, \lambda'_{\mathtt{undec}} \setminus C) \in \mathbb{L}$.*
6. *For $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$, and $C \subseteq \lambda_{\mathtt{out}}$ we have $(\lambda'_{\mathtt{in}}, \lambda'_{\mathtt{out}} \cup C, \lambda'_{\mathtt{undec}} \setminus C) \in \mathbb{L}$.*
7. *For $\lambda, \lambda' \in \mathbb{L}$ with $\lambda_{\mathtt{in}} \subseteq \lambda'_{\mathtt{in}}$ and $\lambda_{\mathtt{out}} \supseteq \lambda'_{\mathtt{out}}$ we have $(\lambda_{\mathtt{in}}, \lambda'_{\mathtt{out}}, \textsc{Args}_{\mathbb{L}} \setminus (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{out}})) \in \mathbb{L}$.*
8. $(\emptyset, \emptyset, \textsc{Args}_{\mathbb{L}}) \in \mathbb{L}$.

Proof. We show that for each SETAF F the set $\mathit{adm}_{\mathscr{L}}(F)$ satisfies the conditions of the proposition. Conditions (1)–(3) are by the fact that $\mathit{adm}_{\mathscr{L}}(F) \subseteq \mathit{cf}_{\mathscr{L}}(F)$. For Condition (4), let $\lambda, \lambda' \in \mathit{adm}_{\mathscr{L}}(F)$ with $\lambda_{\mathtt{in}} \cap \lambda'_{out} = \{\}$ (since each admissible labelling defends itself, $\lambda'_{\mathtt{in}} \cap \lambda_{out} = \{\}$). Thus, $\lambda^* = (\lambda_{\mathtt{in}} \cup \lambda'_{\mathtt{in}}, \lambda_{\mathtt{out}} \cup \lambda'_{\mathtt{out}}, \lambda_{\mathtt{undec}} \cap \lambda'_{\mathtt{undec}})$ is a well-defined labelling. Further, since $\lambda, \lambda' \in \mathit{adm}_{\mathscr{L}}(F)$ it is easy to check that $\lambda^* \in \mathit{adm}_{\mathscr{L}}(F)$. For Condition (5), let $\lambda^* = (\lambda'_{\mathtt{in}} \cup C, \lambda'_{\mathtt{out}}, \lambda'_{\mathtt{undec}} \setminus C)$. First, λ^* is a well-defined labelling.

Notice that the set C contains arguments defended by λ and not attacked by $\lambda'_{\texttt{in}}$. Now, it is easy to check that λ^* meets the condition for being an admissible labelling. For Condition (6), let $\lambda^* = (\lambda'_{\texttt{in}}, \lambda'_{\texttt{out}} \cup C, \lambda'_{\texttt{undec}} \setminus C)$. Notice that the set C contains only arguments attacked by $\lambda_{\texttt{in}}$ and thus are also attacked by $\lambda'_{\texttt{in}}$. Thus, starting from the admissible labelling λ' we can relabel arguments in C to $\texttt{out}$ and obtain that λ^* is also an admissible labelling. For Condition (7), let $\lambda^* = (\lambda_{\texttt{in}}, \lambda'_{\texttt{out}}, \text{ARGS}_{\mathbb{L}} \setminus (\lambda_{\texttt{in}} \cup \lambda'_{\texttt{out}}))$. First, λ^* is a well-defined labelling. We have that setting $\lambda'_{\texttt{out}}$ to $\texttt{out}$ is sufficient to make all the $\texttt{in}$ labels for arguments in $\lambda'_{\texttt{in}}$ valid and thus are also sufficient to make the $\texttt{in}$ labels for arguments $\lambda_{\texttt{in}} \subseteq \lambda'_{\texttt{in}}$ valid. Moreover, as $\lambda_{\texttt{out}} \supseteq \lambda'_{\texttt{out}}$ also labelling arguments $\lambda_{\texttt{in}}$ with $\texttt{in}$ is sufficient to make the $\texttt{out}$ labels for $\lambda'_{\texttt{out}}$ valid. Hence, λ^* is admissible. For Condition (8), the conditions of admissible labelling for arguments labelled $\texttt{in}$ or $\texttt{out}$ in $(\emptyset, \emptyset, \text{ARGS}_{\mathbb{L}})$ are clearly met, since there are no such arguments. □

5. On the Relation between SETAFs and Support-Free ADFs

In order to compare SETAFs with SFADFs, we can rely on SETADFs (recall Theorem 2). In particular, we will compare the signatures Σ^{σ}_{SETADF} and Σ^{σ}_{SFADF}, cf. Definition 10. We start with the observation that each SETADF can be rewritten as an equivalent SETADF that is also a SFADF.[1]

Lemma 8. *For each SETADF $D = (S, L, C)$ there is an equivalent SETADF $D' = (S, L', C')$ that is also a SFADF, i.e. for each $s \in S$, $\varphi_s \in C$, $\varphi'_s \in C'$ we have $\varphi_s \equiv \varphi'_s$.*

Proof. Given a SETADF D, by Definition 8, each acceptance condition is a CNF over negative literals and thus does not have any support link which is not redundant. We can thus obtain L' by removing the redundant links from L and C' by, in each acceptance condition, deleting the clauses that are super-sets of other clauses. □

By the above we have that $\Sigma^{\sigma}_{\text{SETADF}} \subseteq \Sigma^{\sigma}_{\text{SFADF}}$. Now consider the interpretation $v = \{a \mapsto \mathbf{f}\}$. We have that for all considered semantics σ, v is a σ-interpretation of the SFADF $D = (\{a\}, \{\varphi_a = \bot\})$ but there is no SETADF with v being a σ-interpretation. We thus obtain $\Sigma^{\sigma}_{\text{SETADF}} \subsetneq \Sigma^{\sigma}_{SFADF}$.

Theorem 9. $\Sigma^{\sigma}_{SETADF} \subsetneq \Sigma^{\sigma}_{SFADF}$, *for $\sigma \in \{cf, adm, stb, mod, com, prf, grd\}$.*

In the remainder of this section we aim to characterise the difference between $\Sigma^{\sigma}_{\text{SETADF}}$ and Σ^{σ}_{SFADF}. To this end we first recall a characterisation of the acceptance conditions of SFADF that can be rewritten as collective attacks.

Lemma 10. *[16] Let $D = (S, L, C)$ be a SFADF. If $s \in S$ has at least one incoming link then the acceptance condition φ_s can be written in CNF containing only negative literals.*

It remains to consider those arguments in an SFADF with no incoming links. Such arguments allow for only two acceptance conditions $\top$ and $\bot$. While condition $\top$ is unproblematic (it refers to an initial argument in a SETAF), an argument with unsatisfiable acceptance condition cannot be modeled in a SETADF. In fact, the different expressive-

[1] As discussed in [6], in general, SETAFs translate to bipolar ADFs that contain attacking and redundant links. However, when we first remove redundant attacks from the SETAF we obtain a SFADF.

ness of SETADFs and SFADFs is solely rooted in the capability of SFADFs to set an argument to **f** via a $\bot$ acceptance condition.

We next give a generic characterisations of the difference between $\Sigma^{\sigma}_{\mathrm{SETADF}}$ and Σ^{σ}_{SFADF}.

Theorem 11. *For $\sigma \in \{cf, adm, stb, mod, com, prf, grd\}$, we have $\Delta_\sigma = \Sigma^{\sigma}_{SFADF} \setminus \Sigma^{\sigma}_{SETADF}$ with*

$$\Delta_\sigma = \{\mathbb{V} \in \Sigma^{\sigma}_{SFADF} \mid \exists v \in \mathbb{V} \ s.t. \ \forall a : v(a) \in \{\mathbf{f}, \mathbf{u}\} \wedge \exists a : v(a) = \mathbf{f}\}.$$

Proof sketch. First for $\mathbb{V} \in \Delta_\sigma$ the interpretation v cannot be realized in a SETADF as we cannot have $v(a) \in \mathbf{f}$ without $v(b) \in \mathbf{t}$ for some other argument b. On the other hand one can show that when $\mathbb{V} \in \Sigma^{\sigma}_{\mathrm{SFADF}}$ is such that each $v \in \mathbb{V}$ assigns some argument to **t** one can construct a SETADF D with $\sigma(D) = \mathbb{V}$. This is by the fact that we can rewrite acceptance conditions via Lemma 10 and replace $\bot$ acceptance conditions by collective attacks, i.e. for each interpretation we add collective attacks from the arguments set to **t** to all argument with $\bot$ acceptance condition. □

Next, we provide stronger characterisations of Δ_σ for preferred and stable semantics.

Proposition 12. *For $\mathbb{V} \in \Delta_\sigma$ and $\sigma \in \{stb, mod, prf\}$ we have $|\mathbb{V}| = 1$. For $\sigma \in \{stb, mod\}$ the unique $v \in \mathbb{V}$ assigns all arguments to* **f**.

Proof sketch. If a SFADF has a σ-interpretation v that assigns some arguments to **f** without assigning an argument to **t** then we have that the arguments assigned to **f** are exactly the arguments with acceptance condition $\bot$. For *stb* and *mod* semantics this means all arguments have acceptance condition $\bot$ and the result follows. Each preferred interpretation assigns arguments with acceptance condition $\bot$ to **f** and thus the existence of another preferred interpretation would violate the $\leq_i$-maximality of v. □

In other words each interpretation-set which is σ-realizable in SFADFs and contains at least two interpretations can be realized in SETADFs, for $\sigma \in \{stb, prf, mod\}$. We close this section with an example illustrating that the above characterisation thus not hold for *cf*, *adm*, and *com*.

Example 3. Let $D = (\{a, b, c\}, \{\varphi_a = \bot, \varphi_b = \neg c, \varphi_c = \neg b\})$. We have $com(D) = \{\{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}, \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}\}$. By Theorem 11, $com(D)$ cannot be realized as SETADF. Moreover, as $com(D) \subseteq adm(D) \subseteq cf(D)$ for every ADF D, we have that, despite all three contain more than one interpretation, none of them can be realized via a SETADF.

6. Discussion

In this paper, we have characterised the expressiveness of SETAFs under 3-valued signatures. The more fine-grained notion of 3-valued signatures reveals subtle differences of the expressiveness of stable and preferred semantics which are not present in the 2-valued setting [4] and enabled us to compare the expressive power of SETAFs and SFADFs, a subclass of ADFs that allows only for attacking links. In particular, we have exactly characterized the difference for conflict-free, admissible, complete, stable, preferred, and grounded semantics; this difference is rooted in the capability of SFADFs to set an initial argument to false. Together with our exact characterisations on signatures of SETAFs for stable, preferred, grounded, and conflict-free semantics, this also yields the corresponding results for SFADFs. Exact characterisations for admissible and complete semantics are subject of future work. Another aspect to be investigated is to which extent our insights on labelling-based semantics for SETAFs and SFADFs can help to improve the performance of reasoning systems.

Acknowledgments This research has been supported by FWF through projects I2854, P30168. The second researcher is currently embedded in the Center of Data Science & Systems Complexity (DSSC) Doctoral Programme, at the University of Groningen.

References

[1] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.

[2] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

[3] Søren Holbech Nielsen and Simon Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Proc. ArgMAS*, LNCS 4766, pages 54–73, 2006.

[4] Wolfgang Dvořák, Jorge Fandinno, and Stefan Woltran. On the expressive power of collective attacks. *Argument & Computation*, 10(2):191–230, 2019.

[5] Giorgos Flouris and Antonis Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *Int. J. Approx. Reason.*, 109:55–86, 2019.

[6] Sylwia Polberg. *Developing the abstract dialectical framework*. PhD thesis, TU Wien, Institute of Information Systems, 2017.

[7] Bruno Yun, Srdjan Vesic, and Madalina Croitoru. Toward a more efficient generation of structured argumentation graphs. In *Proc. COMMA*, pages 205–212. IOS Press, 2018.

[8] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178, 2015.

[9] Ringo Baumann and Gerhard Brewka. Extension removal in abstract argumentation - an axiomatic approach. In *Proc. AAAI*, pages 2670–2677. AAAI Press, 2019.

[10] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract Dialectical Frameworks: An Overview. In *Handbook of Formal Argumentation*, chapter 5. College Publications, February 2018.

[11] Martin Diller, Atefeh Keshavarzi Zafarghandi, Thomas Linsbichler, and Stefan Woltran. Investigating subclasses of abstract dialectical frameworks. *Argument & Computation*, 11(1), 2020.

[12] Bart Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. *Proc. NAIC*, 96:357–368, 1996.

[13] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[14] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In *Proc. ECAI*, pages 252–260. IOS Press, 2016.

[15] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. *Artif. Intell.*, 278, 2020.

[16] Johannes Peter Wallner. Structural constraints for dynamic operators in abstract argumentation. *Argument & Computation*, 11(1-2): 151-190, 2020.

[17] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proc. IJCAI*, pages 803–809, 2013.

[18] Atefeh Keshavarzi Zafarghandi. Investigating subclasses of abstract dialectical frameworks. Master's thesis, TU Wien, 2017.

[19] Sylwia Polberg. Understanding the abstract dialectical framework. In *Proc. JELIA*, LNCS 10021, pages 430–446, 2016.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200505

Structure or Content? Towards Assessing Argument Relevance

Marc FEGER [1], Jan STEIMANN[1] and Christian METER
Computer Networks Department
Heinrich-Heine-University Düsseldorf, Germany
firstname.lastname@hhu.de

Abstract. In this paper, we provide a detailed analysis of PageRank to determine the relevance of arguments along with content- and knowledge-based methods from the field of natural language processing. We do not only show how the cross-linking of arguments is only slightly involved in the recognition of relevance, we rather show how basic common knowledge and reader-involving methods outperform the purely structure-related PageRank. The methods we propose are based on the latest research and correlate strongly with human awareness regarding the relevance of arguments. Altogether, we show that PageRank does not fully capture the relevance of arguments and must be extended by a contextual level in order to take concepts of natural language into account at the web level, as they are unavoidably involved in argumentation.

Keywords. Argument relevance, Natural language processing, Argumentation and computational linguistics, Computational properties of argumentation, Argumentation and human-computer interaction

1. Introduction

What would you like to search for? What would you like to know? Whether simply surfing the web, doing research, shopping online, or having a discussion about if the new Star Wars film is a success, these questions are always present in our everyday lives. It is not surprising, especially in times when everyday life is more and more transferred to the digital space, that this space also adapts to the needs of the users. The web is becoming a public sphere in which opinions are sometimes shaped by objective and rational debates [1]. As a result, more and more projects like [2,3] are concerned with the systematic preparation and analysis of arguments within this digital space. The sheer unmanageable amount of data also increases the need to filter the most relevant information. For argumentation this means, similar to web documents, the embedded arguments have to be evaluated with respect to their relevance. [4] already suggested a modified version of PageRank by [5], which is supposed to abstract the relevance of arguments, similar to web pages, via their networking, possibly within the web, purely objectively and without consideration of content and logical aspects. As an example, the selection of arguments might look like the following (taken from the dataset used in [4]):

[1] Both authors contributed in equal parts to this work.

Question: *Why should peanuts be banned on board aircraft?*

Answer 1 (a_1)**:** *Peanut reactions can be life threatening. An individual doesn't have to consume the product to have a life threatening reaction. They can have contact or inhalation reactions.*

Answer 2 (a_2)**:** *Providing buffer zones to avoid contact with peanuts is a thoughtful gesture. But from a practical point of view, it does not work.*

Answer 3 (a_3)**:** *With so many food choices available, why are peanuts a necessary choice?*

Answer 4 (a_4)**:** *Restricting the ban of peanut products to certain flights is not enough.*

Although all answers take up the topic of the question, they are still of varying relevance. Compared to a_2, a_1 is more topic-related, informative and meaningful, as it addresses not only direct dangers but also implicit knowledge such as the restricted space inside aircraft. However, the comparison between a_2 and a_3 is highly subjective. Nonetheless, it can be assumed that a_2 is more relevant than a_3, since it does not contain a question and is therefore more concrete. Clearly, despite its formulation as a question, a_3 appears more informative and more concrete than a_4, since it contains a general conclusion.

In this paper we use the Webis-ArgRank-2017 data by [4] (Section 3) to investigate the impact of content- and knowledge-based methods (Section 4) on perception regarding the relevance of arguments. For this purpose the results of the pioneering work of [4] were reproduced. Besides the influence of PageRank on the relevance of the arguments, we compare them with the results obtained by using more recent and knowledge-based methods (Section 5). Our results show that PageRank and especially the evaluation of relevance exclusively by linking up arguments is not yet satisfactory. Rather, we show that simple content-based methods working with a general conceptual knowledge can achieve significantly better results (Section 6).

Consequently, our work takes up the thesis of [4] suggesting that relevance is structurally induced, and demonstrates how content-based properties co-determine inevitably the relevance of arguments, as these are unavoidably taken into account by a rational reader.

2. Related Work

In argumentation theory, relevance is considered in two parts. Local relevance describes the extent to which the premises of an argument contribute to the acceptance or rejection of the corresponding conclusion [6]. In contrast, global relevance describes the extent to which the argument contributes to the understanding of a topic [6]. [7] examined the influence of trust in an argument. Subsequently, supported by [8], it was stated that a globally relevant argument must necessarily also be locally relevant. However, a locally relevant argument need not necessarily be globally relevant. Despite the differences between the two dimensions, no sharp distinction is made in this work. Rather, first investigations of different methods are carried out to classify the methods proposed by [4]. Moreover, the relevance of an argument must be understood as a dimension of the quality of the argument itself. [9] proposed a tautological division of quality into the three main components: cogency, effectiveness and reasonableness. Whereby the two forms of relevance appear in the dimensions cogency and reasonableness. In addition, [9] notes that these dimensions can be interdependent and also branch out into sub-dimensions. Thus, global relevance is represented as a branch of reasonableness and local relevance as another branch of cogency. Reasonableness describes the extent to which an argument contributes to the understanding of a problem. Furthermore, cogency describes the extent to which the premise relevance contributes to a coherent understanding of the conclusion of an argument. Furthermore, the dependence of reasonableness on cogency is not only supported by the correlation of local and global relevance. The dependence of reasonableness and cogency is in line with the observation of [7] and [8].

In practice, a cornerstone of argument mining is understood to be the work written by [10] dealing with the collection of arguments within documents. In this context, [11] introduced the topic of argument recognition. Comments and the arguments contained therein are recognized by assigning arguments from a predefined set to the corresponding comment using similarity- and entailment-based properties. In addition, [12] identified prominent arguments in online debates by clustering using semantic- and text-based methods. Similarly, [13] used SVM and BLSTM with GloVe input layer to investigate whether the persuasiveness of an argument can be systematically captured. It was found that PageRank does not induce the persuasiveness of arguments and that a higher PageRank is associated with a lower convincingness. Likewise, [14] sketched the PageRank for capturing the global relevance of arguments. This sketch was performed by [4]. In doing so, the modified PageRank method beats a variety of intuitive comparison methods. These comparison methods covered the topics: semantics, similarity and graph structure. In addition, [15] achieved significant results regarding the finding of good counterarguments by using word- and embedding-based methods. Contiguous to this, [16] carried out an analysis of the relevance of arguments using four basic information retrieval methods: DirichletLM, DPH, BM24 and TFIDF. The results showed how the more modern methods performed significantly better and were able to capture the correlation between the relevance and the general quality of an argument.

3. Corpus

We conduct our study on the Webis-ArgRank-2017 dataset. In this dataset [4] constructed a large ground-truth argument graph as well as a ranking of a subset of arguments within this graph. This dataset serves as a first benchmark for evaluating argument relevance assessments. [4] acquired the data for constructing the graph from the Argument Web [17] storing the arguments in the Argument Interchange Format [18]. On June 2, 2016, when [4] accessed the data, the Argument Web was the largest existing argument database with structured argument corpora. It consisted of 57 corpora with 8479 argument-maps storing all information about the arguments, summing up to 49.504 argument units, describing either a premise or a conclusion, and 26.012 arguments. Duplicates and nodes which were not connected to any inference node were removed by the authors. This lead to the resulting graph with 31.080 different argument units of which 28.795 participated in 17.877 arguments. Altogether the arguments can be combined to a not necessarily coherent graph $G = (A, E)$. Each node $a_i \in A$ of the graph G describes an argument consisting of a conclusion c_i and a non-empty set of premises P_i. Thus, an argument $a_i \in A$ is represented with $a_i = \langle c_i, P_i \rangle$. An edge $(a_j, a_i) \in E \subseteq A \times A$ is given if the conclusion of a_j is used as a premise of a_i. Consequently, $P_i = \{c_1, \cdots, c_k\}, k \geq 1$.

In our work we were able to reproduce the resulting argument graph and numbers which [4] stated out in their paper. The authors made the 28.795 argument units and all following data available. We found slightly different numbers for the ground-truth-argument graph and the argument relevance benchmark dataset. Thus, [4] found 17.372 of all 31.080 argument units never to be used as a conclusion. Whereas our reproduction showed the same circumstance for 17.370 argument units. Building on this we came up with 17.096 argument units while [4] findings showed 17.093 argument to be used only once as a premise. Nevertheless, we consider the differences to be negligible since the difference is too small to have a deeper impact to a general assumption.

3.1. Benchmark Argument Ranking

In the constructed graph 3113 conclusions were part of more than one argument. Therefore, they were candidates for ranking. [4] selected 498 conclusions to be classified by two experts from computational linguistics. These experts decided for each conclusion if it contains a real claim or, e.g., if it has a personal context. Only if both experts saw a real claim the arguments has been kept. The remaining arguments were examined whether they allowed a logical inference to be drawn, if they form a valid counter-argument or if they were based on reasonable premises. The resulting benchmark dataset consists of 32 conclusions which participate in 110 arguments. These 110 arguments were then ranked by seven experts from computational linguistics and information retrieval. Each argument was ranked by how much each of its premises contributes to the acceptance or rejection of the conclusion. In terms of Kendall's τ the mean over all agreement was 0.36. The authors explain this low agreement with the general high subjectivity of argument relevance.

4. Methods

Basically, the original form of PageRank, as applied by [5], is based on the popularity of cross-linked websites. Accordingly, the relevance of a website depends on how many other relevant websites offer direct linking to this page. Furthermore, this procedure can be transferred to the argumentation graph G by replacing the web pages with argument units representing either a premise or a conclusion. Thus, the relevance of an argument, indicated by its conclusion, at a certain point in time t results from its premises and their interconnection according to G:

$$p_t(c_i) = \begin{cases} (1-\alpha)\frac{1}{|D|} + \alpha \sum_j \frac{p_{t-1}(c_j)}{|P_j'|} & : t > 0 \\ \frac{1}{|D|} & : t = 0 \end{cases} \tag{1}$$

For the initialization it should be considered that each argument is assigned the same relevance $\frac{1}{|D|}$. $|D|$ describes the number of all unique argument units in G. For each subsequent point in time $t > 0$ the relevance results from the α-weighted sum of the ground relevance $\frac{1}{|D|}$ and the linking relevance $\sum_j \frac{p_{t-1}(c_j)}{|P_j'|}$. The linking relevance reflects the importance of those arguments a_j whose conclusion is used as premise by a_i. If the conclusion c_j of a_j is used in $|P'_j|$ cases as a premise of further arguments, its relevance $p_{t-1}(c_j)$ is distributed accordingly. In addition to the *custom-made PageRank* (CPR) developed in this paper, we use the implementation of NetworkX [19] and its extension via Scipy [20] for control purposes.

4.1. Baselines Applied

Just like the baselines presented by [4], our approaches emphasize the collaboration of the premises for the respective conclusion. However, our approaches differ, not only because they user newer methods, but also because they take up aspects of content and language. For example, the *Similarity* used by [4] is intuitive, but it only covers strict word similarity. Furthermore, the *Frequency* of a premise across the arguments is used to measure relevance. Although both methods are intuitively calculable, they do not really match the judgement of a human reader of an argument. On the other hand, the methods we use are more focused on the reader and the way in which the viewer perceives an argument. Therefore, our methods take up concepts and linguistic constructs which are superior to the text as they occur in natural language. By deliberately emphasizing the dependencies within an argument, we want to tighten the baselines for the PageRank approach introduced by [4]. This is necessary because PageRank alone, through structural and without content aspects, is supposed to induce relevance in a linguistic environment, as it is the case in a debate. To make our work comparable, we adopted the intuitive baselines *Similarity*, and *Sentiment* and oriented our methods accordingly. Corresponding, the new baselines used in this paper are listed[2].

4.2. Similarity

An important aspect when comparing non-content-based methods like PageRank is the collection of content-related aspects of the data. Therefore, our methods address the so-called *semantic similarity*. Conversely, this means that the components of an argument at the word or sentence level are transferred into a corresponding vector representation. In this paper we have used Flair [21] to calculate the respective embedding in vectors.

4.3. Vector Space Models

In total, we have investigated three different ways to embed words or sentences within this work. We used GloVe [22] as a first vector representation for our model. This method produces unsupervised vector-space representations of words. Thereby a global word-word-co-occurrence statistic is learned. This kind of learning is based on linear relationships of words which correspond to a semantic similarity. Thus, vector relations as given by [23] can be described using a corresponding aggregation with, e.g., $king - man + women \approx queen$. In this paper both *GloVe with punctuation* (GWP) and *GloVe without punctuation* (GWOP) were investigated. ELMo [24] was used as the second method. Here, the embeddings are learned on the basis of a bidirectional language model. Thus, linguistic contextual properties are learned in addition to syntactic and semantic properties of the words. Apart from semantic analysis, these can also be used to answer questions and the associated textual inference. Thus, it can be determined to

[2] The code is located at: `https://github.com/hhucn/argument-relevance-paper-results`.

what extent a premise indicates a logical conclusion. Similar to GloVe, both *ELMo with punctuation* (EWP) and *ELMo without punctuation* (EWOP) were used. The third approach used was BERT [25]. This procedure also develops embeddings via bidirectional language models. However, newer techniques such as transformers etc. are added, making the embeddings given even more detailed. Similar to the previous models, we used *BERT with punctuation* (BWP) and *BERT without punctuation* (BWOP). To determine the resemblance, the Cosine-Similarity was used for each of the models mentioned, since this provided by far the most favorable results.

4.3.1. WordNet

Additionally, we have used the knowledge-based similarity function $Sim(T_1,T_2)$ introduced by [26]. This method determines the semantic similarity of two input texts T_1,T_2 by mutually picking up similar concepts. Vice-versa each individual word w of one text is compared, over the highest conceptual similarity, with the entire word concepts of the others by using the weighting $idf(w)$. Analogously, we considered in this paper a weakened variant limited to similarity from T_1 to T_2 via the average conceptual similarity across the words of T_1 to the totality of word concepts occurring in T_2. For both the *mutual knowledge-based method* (MKBM) and for the *average knowledge-based method* (AKBM), the implementation of the thesaurus WordNet [27] by NLTK [28] was used to identify the word concepts. In this thesaurus words are connected with respect to their synonyms in the form of synsets. To determine the similarity of words, the Wu-Palmer-Similarity $CoSim(c_1,c_2)$ [29] was used, which takes into account the depth of the concepts c_1,c_2 to be compared as well as the least common ancestor of both. We have made use of this similarity because, analogous to a family tree, it picks up the origin of two concepts and thus connects them to superordinate knowledge.

4.4. Sentiment

Just like [4], we have taken up the subject of sentiment, as it can certainly contribute to the persuasive power of an argument. Sentiment uses the positive tone of the premises to calculate a score for the argument. Unlike [4], a *sentiment neuronal network* (SNN) based on FastText [30], which was trained on the film ratings of IMDb, was used instead of SentiWordNet [31]. The advantage of this model architecture is its speed and simplicity. Whereby it merely consists of an input layer, which then passes the averaged feature vectors, using GloVe embeddings, to a linear classifier.

5. Results

In the subsequent section, besides the detailed analysis regarding the different implementations of PageRank and their dependency on α, the baselines given by [4] are presented in comparison with our results. It should be mentioned that the relevance of a conclusion can always be derived from the premises. For this purpose, the four different aggregations of the premise values *min*, *average*, *max*, *sum* are listed, resulting in an overall relevance of the conclusion. The relevance of *min* and *max* is determined by the smallest and the largest value of the premises. Similarly, the *sum* and *average* are used to determine the relevance of the entirety of the premises.

5.1. PageRank Comparison

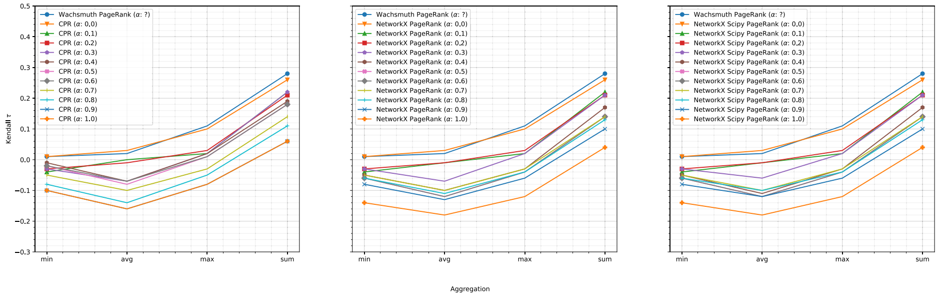


Figure 1. Development of the perception regarding the argument relevance induced by PageRank regarding all possible aggregations. CPR, NetworkX and NetworkX using Scipy were plotted against the result obtained by [4] for different α values, which regulates the influence of linking the arguments.

Since PageRank in its modified and unmodified variants always operates structurally and not content-related, a more in-depth investigation of the method as such is necessary. In the context of this, a more precise illumination of the influence of the parameter α on the supposedly induced relevance is also required. Figure 1 shows the results of the different implementations of PageRank on the argumentation graph G. The results were plotted according to the aggregations against the results obtained by [4].

As a first observation, it can be stated that the different implementations for $\alpha = 0$ achieve comparable results and, moreover, can easily keep up with the variant presented by [4]. Due to the different setups and sometimes different implementation details, slightly different results regarding the achieved τ values per aggregation are obtained. Furthermore, it can be seen that for an increasing α, which is accompanied by a higher influence of the cross-linking of arguments, the general agreement regarding relevance decreases over each aggregation, resulting in a low point for $\alpha = 1$.

5.2. Extended Baselines

As one part of this paper the baselines given by [4] were reproduced: *Random*, *Most premises*, *Frequency*, *Sentiment*, *Similarity* and *PageRank*. Their results were plotted in Figure 2 against the values obtained by the methods shown in Section 4. In contrast to the thesis stated by [4] according to which PageRank induces relevance better than frequency-based and simple content-based methods, it can be seen how methods that emphasize linguistic aspects through content-related and contextual properties achieve significantly better results in the assessment of relevance. The results achieved by using WordNet and GloVe are particularly striking.

Table 1 shows the direct numerical comparison. Altogether GWP performs with $\tau \approx 0.47$. Moreover, AKBM achieves a correlation of $\tau \approx 0.4$ and MKBM of $\tau \approx 0.34$. Likewise, the rating of the SNN for determining sentiment is $\tau \approx 0.31$. In contrast, BWP and BWOP achieve only marginally lower results than PageRank, whereas EWP performs $\tau \approx 0.28$. Therefore, results similar to PageRank are achieved. Despite the strong subjectivity of the task, the approaches mentioned above perform rather strongly compared to the average agreement among all annotators of $\tau \approx 0.36$. Similarly, GWP performs best in 16 out of 32 cases and worst in only 1 case. Not only does this result out-

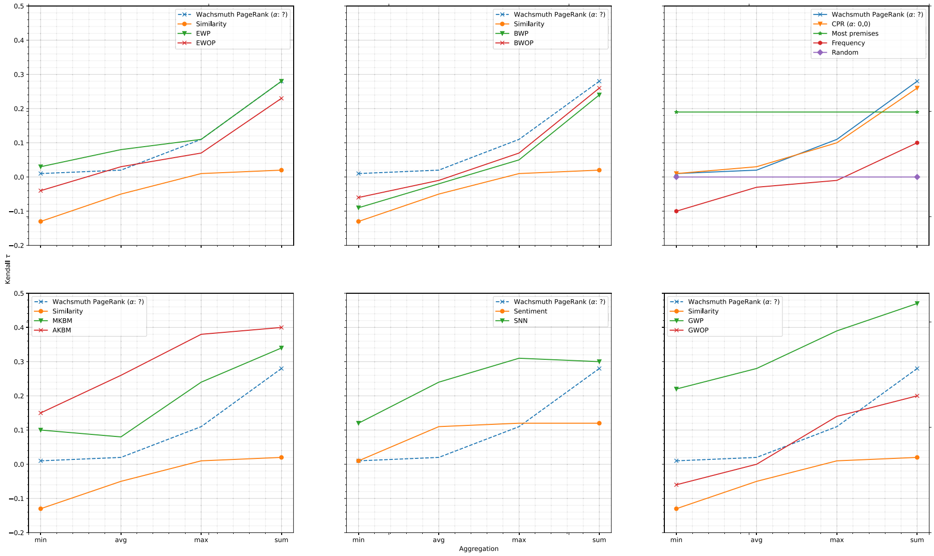


Figure 2. Direct comparison of the awareness regarding the relevance of arguments of all baseline values reported by [4] with all results obtained in this paper. *Most premises* and *Random*, which achieved slightly different values due to unequal random procedures, were adopted across all aggregations, as these can only be applied in *sum*.

perform the results of the modified PageRank reported by us, which is better in 11 cases and worse in 5 cases, but it also exceeds the results reported by [4], with 15 best and 3 worst. Furthermore, AKBM comes second with 14 best and 4 worst results followed by MKBM with 13 best and 6 worst cases. Also, SNN is slightly better than MKBM with 13 best and 5 bad cases. Likewise, EWP, EWOP, BWP and BWOP are similar to the PageRank results we previously reported.

#	Approach	(a) Minimum			(b) Average			(c) Maximum			(d) Sum			(e) Best results		
		τ	*best*	*worst*	τ	*best*	*worst*	τ	*best*	*worst*	τ	*best*	*worst*	τ	*best*	*worst*
1	PageRank	0.01	8	6	0.02	9	7	0.11	8	6	0.28	11	5	0.28	11	5
2	Frequency	-0.10	2	8	-0.03	3	9	-0.01	2	8	0.10	6	8	0.10	6	8
3	Similarity	-0.13	4	11	-0.05	5	11	0.01	6	10	0.02	6	10	0.02	6	10
4	Sentiment	0.01	6	7	0.11	9	4	0.12	6	4	0.12	9	4	0.12	9	4
5	Most premises	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	0.19	3	3	0.19	3	3
6	Random	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	0.00	5	7	0.00	5	7
7	SNN	0.12	10	6	0.24	11	5	0.31	12	5	0.30	13	5	0.31	13	5
8	GWP	**0.22**	12	5	**0.28**	13	**3**	**0.39**	**14**	**2**	**0.47**	**16**	**1**	**0.47**	**16**	**1**
9	GWOP	-0.06	5	9	0.00	6	7	0.14	8	6	0.20	8	4	0.20	8	4
10	EWP	0.03	6	9	0.08	7	8	0.11	8	8	0.28	9	5	0.28	9	5
11	EWOP	-0.04	5	9	0.03	6	8	0.07	7	8	0.23	9	6	0.23	9	6
12	BWP	-0.09	6	9	-0.02	7	8	0.05	9	8	0.24	10	5	0.24	10	5
13	BWOP	-0.06	6	9	-0.01	7	8	0.07	9	8	0.26	10	5	0.26	10	5
14	MKBM	0.10	5	7	0.08	13	6	0.24	12	8	0.34	11	9	0.34	13	6
15	AKBM	0.15	**14**	**4**	0.26	**14**	4	0.38	11	7	0.40	13	7	0.40	14	4

Table 1. Comparison of the approaches of [4] (1-6) with those used in this study (7-15). For each aggregation (a-d) the average agreement τ and the cases in which the respective approach performed best or worst over the 32 conclusions of the 110 arguments are given. (e) shows the best results of an aggregation.

6. Interpretation

Based on these results, we can embed the PageRank for the purpose of argument relevance into the current state of research. In order to achieve a better comparability of the results and procedures, we have divided these underlying methods into two categories. Thus, the vector space models working with ELMo and BERT belong to the group of direct contextual methods, which require expert knowledge of the discussion and its language usage. In this group we also include PageRank, because it includes a structural context. On the other hand, we consider those approaches working with WordNet, GloVe and Sentiment to be part of the group of indirect contextual methods. This does not mean that the mentioned methods of this group miss the underlying context completely. WordNet and GloVe, for example, take up linguistic similarity as well as higher-level concepts and work with them in the context of local evaluation of arguments. Likewise, the positive sentiment appeals to such a local context and thus emphasizes respectful interaction and the constructive effect resulting from this.

As a first finding, the methods of the directly context-related methods obtain comparable results. In some cases, PageRank even outperforms vector space models using ELMo and BERT, which could be due to the fact that the training data did not contain sufficient argumentative data, which occasionally also needs to be included in the ground-truth. However, the overall course of the results is mostly identical, indicating that similar underlying properties have been captured. We assume that the PageRank given by [4] used a α-weighting around $\alpha \approx 0$, since CPR achieved the highest agreement for $\alpha = 0$. Thus, the actual advantage of PageRank, which is to determine relevance through structure, is nearly absolutely lost, since the linking relevance is only included very poorly in the computation. Therefore, based on the results of CPR, none or at least very small portions of the structure underlying the argumentation are included in the assessment of argument relevance. The fact that the results of these methods perform similarly well accentuates the quality of the PageRank as well as the complexity of the task due to the diversity of approaches.

The methods of the indirect context-related group behave differently. Thus, all the methods mentioned perform significantly higher in terms of the awareness of relevance. GWOP, for example, without considering sentence structures, is comparable to the previous results of the context-related methods. However, GWP clearly stands out from all results by using sentence structures. The same behavior is observed for MKBM, AKBM and SNN. The better performance of those methods using WordNet and GloVe with respect to the use of sentence structures can be attributed to the local approach. Besides punctuation, both methods only consider word analogies that result from the local context. Thus, the sentence structure for the WordNet methods is taken up through idf, since the premises are considered in sentences where each sentence represents a document. For GloVe, word similarities result from their local combinations and for the WordNet methods from the synergy of the superordinate concepts. Furthermore, the positivity for Sentiment is restricted to the local context. Thus, it is not surprising that the best result for the aggregation *max* is achieved. This corresponds to the view that the most constructive premise contributes to the relevance of the argument. Overall, there seems to be an advantage of those methods which take the local context into account and thus emphasize the local relevance much better, as this reflects the reader of an argument better.

7. Conclusion

In this paper, we have shown how already existing content- and knowledge-based methods clearly exceed the PageRank as modified by [4] for determining the relevance of arguments. We were also the first to transfer the latest approaches from the field of natural language processing to the assessment of argument relevance. Additionally, we were able to embed the modified PageRank into the current state of research. We provide evidence that superordinate knowledge and concepts of natural language are more important for relevance than structural methods like PageRank, because they are more likely to be involved in convincingness. Thus, the observation of [13] suggesting counter-productive effects of PageRank on convincingness can be confirmed. Nevertheless, the PageRank should still be investigated, as it can achieve meaningful results despite its low degree of interconnectedness. Even if the properties of the arguments can be precalculated by the presented methods, their scalability on the web should still be investigated in more detail. The precalculation allows the properties to be uniquely assigned to an argument. Thus, the scaleability in the web is not limited by expensive calculations. Therefore, the presented methods could possibly keep up with the already well investigated scalability of PageRank, which also involves a precalculation phase. We therefore propose to combine content- and knowledge-based approaches with structure-emphasizing methods similar to the Hummingbird [32] algorithm used by Google, which replaced PageRank in 2013 and only partially integrates it into search queries. We are looking forward to jointly solve the existing problems and thereby paving the way for search engines to consider arguments and especially their relevance.

References

[1] D. Rasmussen, J. Habermas, C. Lenhardt, and S. Nicholsen, “Moral consciousness and communicative action.” *The Philosophical Quarterly*, vol. 43, no. 173, p. 571, 10 1993. [Online]. Available: https://doi.org/10.2307/2220013

[2] C. Meter and A. Schneider, “Various Efforts of Enhancing Real World Online Discussions,” in *ECA 2019: Proceedings of the 3rd European Conference on Argumentation*, June 2019.

[3] T. Krauthoff, C. Meter, M. Baurmann, G. Betz, and M. Mauve, “D-BAS âĂŞ A Dialog-Based Online Argumentation System,” in *Computational Models of Argument*, September 2018, pp. 325–336. [Online]. Available: http://doi.org/10.3233/978-1-61499-906-5-325

[4] H. Wachsmuth, B. Stein, and Y. Ajjour, “"PageRank" for argument relevance,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017, pp. 1117–1127. [Online]. Available: http://aclweb.org/anthology/E17-1105

[5] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[6] D. Walton, “Informal logic: A pragmatic approach, second edition,” *Informal Logic: A Pragmatic Approach, Second Edition*, pp. 1–347, 01 2008. [Online]. Available: https://doi.org/10.1017/CBO9780511808630

[7] F. Paglieri and C. Castelfranchi, “Trust, relevance, and arguments,” *Argument and Computation*, vol. 5, pp. 216–236, 2014. [Online]. Available: https://doi.org/10.1080/19462166.2014.899270

[8] C. Jacobs and S. Jackson, “Relevance and digressions in argumentative discussion: A pragmatic approach,” *Argumentation*, vol. 6, no. 2, pp. 161–176, 05 1992. [Online]. Available: https://doi.org/10.1007/BF00154323

[9] H. Wachsmuth, N. Naderi, Y. Hou, Y. Bilu, V. Prabhakaran, T. A. Thijm, G. Hirst, and B. Stein, “Computational argumentation quality assessment in natural language,” in *Proceedings of the 15th*

Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Valencia, Spain: Association for Computational Linguistics, 04 2017, pp. 176–187. [Online]. Available: https://doi.org/10.18653/v1/E17-1017

[10] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ser. ICAIL âĂŹ07. New York, NY, USA: Association for Computing Machinery, 2007, p. 225âĂŞ230. [Online]. Available: https://doi.org/10.1145/1276318.1276362

[11] F. Boltužić and J. Šnajder, "Back up your stance: Recognizing arguments in online discussions," in *Proceedings of the First Workshop on Argumentation Mining*. Baltimore, Maryland: Association for Computational Linguistics, 06 2014, pp. 49–58. [Online]. Available: https://doi.org/10.3115/v1/W14-2107

[12] ——, "Identifying prominent arguments in online debates using semantic textual similarity," in *Proceedings of the 2nd Workshop on Argumentation Mining*. Denver, CO: Association for Computational Linguistics, 06 2015, pp. 110–115. [Online]. Available: https://www.aclweb.org/anthology/W15-0514

[13] I. Habernal and I. Gurevych, "Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany: Association for Computational Linguistics, 08 2016, pp. 1589–1599. [Online]. Available: https://doi.org/10.18653/v1/P16-1150

[14] K. Al-Khatib, H. Wachsmuth, M. Hagen, J. Köhler, and B. Stein, "Cross-domain mining of argumentative text through distant supervision," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 06 2016, pp. 1395–1404. [Online]. Available: https://doi.org/10.18653/v1/N16-1165

[15] H. Wachsmuth, S. Syed, and B. Stein, "Retrieval of the best counterargument without prior topic knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational*. Melbourne, Australia: Association for Computational Linguistics, 07 2018, pp. 241–251. [Online]. Available: https://doi.org/10.18653/v1/P18-1023

[16] M. Potthast, L. Gienapp, F. Euchner, N. Heilenkötter, N. Weidmann, H. Wachsmuth, B. Stein, and M. Hagen, "Argument search: Assessing argument relevance," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIRâĂŹ19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1117âĂŞ1120. [Online]. Available: https://doi.org/10.1145/3331184.3331327

[17] F. Bex, J. Lawrence, M. Snaith, and C. Reed, "Implementing the argument web," *Commun. ACM*, vol. 56, no. 10, p. 66âĂŞ73, Oct. 2013. [Online]. Available: https://doi.org/10.1145/2500891

[18] C. Chesnevar, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, S. Willmott *et al.*, "Towards an argument interchange format," *Knowl. Eng. Rev.*, vol. 21, no. 4, p. 293âĂŞ316, Dec. 2006. [Online]. Available: https://doi.org/10.1017/S0269888906001044

[19] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15. [Online]. Available: http://conference.scipy.org/proceedings/SciPy2008/paper_2/full_text.pdf

[20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," pp. 261–272, 2020. [Online]. Available: https://doi.org/10.1038/s41592-019-0686-2

[21] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 08 2018, pp. 1638–1649. [Online]. Available: https://www.aclweb.org/anthology/C18-1139

[22] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Doha, Qatar: Association for Computational Linguistics, 10 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/D14-1162

[23] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPSâĂŹ16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4356âĂŞ4364. [Online]. Available: http://arxiv.org/abs/1607.06520

[24] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[25] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[26] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, ser. AAAIâĂŹ06. AAAI Press, 2006, p. 775âĂŞ780.

[27] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 05 1998, isbn: 9780262061971.

[28] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'REILLY, 01 2009, isbn: 978-0-596-51649-9.

[29] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," USA, p. 133âĂŞ138, 1994. [Online]. Available: https://doi.org/10.3115/981732.981751

[30] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, 04 2017, pp. 427–431. [Online]. Available: https://doi.org/10.18653/v1/E17-2068

[31] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), 05 2010. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

[32] A. P. D. R. Yazdanifard, "How googleâĂŹs new algorithm, hummingbird, promotes content and inbound marketing," *American Journal of Industrial and Business Management*, vol. 4, pp. 51–57, 01 2014. [Online]. Available: https://doi.org/10.4236/ajibm.2014.41009

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200506

Argumentative Reflections of Approximation Fixpoint Theory

Jesse HEYNINCK [a,1] and Ofer ARIELI [b,2]
[a] *Department of Computer Science, TU Dortmund, Germany*
[b] *School of Computer Science, The Academic College of Tel-Aviv, Israel*

Abstract. In this paper we show that non-monotonic formalisms that are represented by approximation fixpoint theory can also be represented by formal argumentation frameworks. By this, we are able not only to recapture and generalize many forms of non-monotonic reasoning in the context of argumentation theory, but also introduce new argumentative representations that have not been considered so far.

Keywords. Non-monotonic reasoning, approximation fixpoint theory, assumption-based argumentation.

1. Introduction

In a series of papers (e.g., [7,8]) Denecker, Marek, and Truszczyński introduced a general technique, using approximating fixpoint computations, for constructively characterizing a variety of non-monotonic formalism. In [17] this method was further applied to a variety of logic programs (including normal logic programs, first order logic programs, and logic programming with aggregates), and in [1] it has been applied to HEX programs. In this paper, we show how fixpoints of approximating operators can be represented by (extensions of) the 'reflecting' assumption-based argumentation framework, thus allowing for argumentative counterparts of corresponding characterizations that were provided in terms of approximation fixpoint theory. These alternative argumentative characterizations generalize known characterizations of semantics of non-monotonic formalisms such as default logic, logic programming and autoepistemic logic that are introduced in, e.g., [3,5], and allow for new argumentative representations of other formalisms for non-monotonic reasoning.

The paper is organized as follows: in the next section we recall some basic notions from assumption-based argumentation, approximation fixpoint theory, and semantics for logic programing. In Section 3 we show how argumentation theory can be used for characterizing semantics of propositional logic programs and how this can be generalized to reflections of approximated fixpoint concepts. In Section 4 we give some applications of our results, and in Section 5 we conclude.

[1] Corresponding Author. Email: jesse.heyninck@tu-dortmund.de. Supported by the German National Science Foundation, DFG-project KE-1413/11-1.

[2] Supported by the Israel Science Foundation, grant No. 550/19.

2. Preliminaries

2.1. Assumption-Based Argumentation (ABA)

Definition 1. A *reasoning frame* for a propositional language $\mathcal{L}$ is a pair $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$, where $\vdash$ is a monotonic binary relation between sets of formulas and formulas in $\mathcal{L}$ (so if $\Gamma \vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \psi$).

The next definition, which is a variation of that in [13], generalizes the definition in [3] of assumption-based frameworks.

Definition 2. An *assumption-based framework* (ABF, for short) is a triple $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$, where:

- $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ is a reasoning frame,
- Λ (the *defeasible assumptions*) is a non-empty, countable set of $\mathcal{L}$-formulas,
- $- : \Lambda \rightarrow \wp(\mathcal{L})$ is a contrariness operator, assigning a finite set of $\mathcal{L}$-formulas to every defeasible assumption in Λ.[3]

Remark 1. In [3,13] ABFs are in fact quadruples, where the assumptions are divided to strict and defeasible ones. In what follows strict assumptions will not be needed, so they are removed from the definition. Yet, our results can be easily adjusted to ABFs with a set Γ of strict premises, defined e.g. by $\Gamma = \{\psi \mid \emptyset \vdash \psi\}$.

Remark 2. In this paper we shall concentrate on *flat* ABFs. In such ABFs the sets of assumptions are always closed, i.e., they contain any assumption they imply. Non-flat characterizations of approximate fixpoint theory will be investigated in future work.

Attacks in ABFs of assertions by counter-assertions are defined as follows:

Definition 3. Let $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$ be an assumption-based framework, $\Delta, \Theta \subseteq \Lambda$, and $\psi \in \Lambda$. We say that Δ *attacks* ψ iff $\Delta \vdash \phi$ for some $\phi \in -\psi$. Accordingly, Δ attacks Θ if Δ attacks some $\psi \in \Theta$.

The last definition gives rise to the following adaptation to ABFs of the usual semantics for abstract argumentation frameworks [9].

Definition 4. [3] Given $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$, we denote $\mathcal{AF}(\mathsf{ABF}) = (\wp(\Lambda), \rightsquigarrow)$ where $(\Delta, \Theta) \in \rightsquigarrow$ for some $\Delta, \Theta \subseteq \Lambda$ iff Δ attacks Θ. We denote $\Delta^+ = \{\phi \in \Lambda \mid \Delta \rightsquigarrow \phi\}$.

For $\Delta \subseteq \Lambda$, we say that

Δ is *conflict-free* iff there is no $\Delta' \subseteq \Delta$ that attacks some $\psi \in \Delta$. Δ *defends* a set $\Delta' \subseteq \Lambda$ iff for every set Θ that attacks Δ' there is a set $\Delta'' \subseteq \Delta$ that attacks Θ. Δ is *admissible* iff it is conflict-free and defends every $\Delta' \subseteq \Delta$. Δ is *complete* iff it is admissible and contains every $\Delta' \subseteq \Lambda$ that it defends. Δ is *grounded* iff it is minimally complete.[4] Δ is *preferred* iff it is maximally complete.[5] Δ is *stable* iff it is conflict-free and $\Delta^+ = \Lambda \setminus \Delta$.

[3] Here and in what follows $\wp(\mathcal{L})$ denotes the powerset of $\mathcal{L}$.

[4] For flat ABFs the grounded extension always exists and it is unique.

[5] Often, preferred extensions are defined as maximally admissible. However, for flat ABFs, these definitions are equivalent.

Example 1. Let $\mathsf{ABF}_{\mathcal{P}} = \langle \mathfrak{L}_{\mathsf{MP}}, \{\neg p, \neg q, \neg r\}, -\rangle$ be an assumption-based argumentation framework in which $-\neg x = \{x\}$ for any $x \in \{p,q,r\}$, and where $\mathfrak{L}_{\mathsf{MP}}$ is a reasoning setting whose entailment relation is based on Modus Ponens as its single rule, i.e.: $\Delta \vdash \psi$ if there is a derivation of ψ based on the formulas in Δ and the inference rule

$$[\mathsf{MP}_{\mathcal{P}}] \quad \frac{\phi_1 \quad \phi_2 \quad \cdots \quad \phi_n \quad \psi \leftarrow \phi_1, \ldots, \phi_n \in \mathcal{P}}{\psi}$$

where in this case we take: $\mathcal{P} = \{q \leftarrow \neg p;\ p \leftarrow \neg q;\ r \leftarrow \neg q;\ r \leftarrow \neg r\}$. Figure 1 below is a schematic representation of a fragment of the attack relation in ABF.

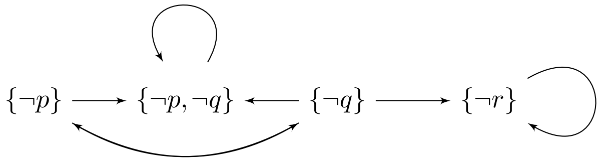


Figure 1. The ABF of Example 1

The sets $\emptyset, \{\neg p\}, \{\neg q\}$ are admissible (and complete) in ABF. The latter two are also preferred, and $\{\neg q\}$ is also stable. The grounded extension here is $\emptyset$.

2.2. Approximation Fixpoint Theory (AFT)

Next, we review the basics notions of approximation fixpoint theory (AFT, [7]). Its main purpose is to find constructive techniques for approximating the fixpoints of an operator O over a lattice L. For this, the following structure (known as a *bilattice*, see [11,12]) is useful:

Definition 5. *Given a lattice $L = \langle \mathcal{L}, \leq \rangle$, we let $L^2 = \langle \mathcal{L}^2, \leq_i, \leq_t \rangle$ be a structure in which $\mathcal{L}^2 = \mathcal{L} \times \mathcal{L}$, and for every $x_1, y_1, x_2, y_2 \in \mathcal{L}$,*

- *$(x_1, y_1) \leq_i (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \geq y_2$,*
- *$(x_1, y_1) \leq_t (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \leq y_2$.*

An approximation operator $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ of $O_{\mathcal{L}} : \mathcal{L} \to \mathcal{L}$ is defined by specifying two operators $\mathcal{O}_l$ and $\mathcal{O}_u$ which calculate a *lower* and an *upper bound* for the value of $O_{\mathcal{L}}$. It is observed in [7] that many formalisms can be characterized by a symmetric operator where the upper bound can be calculated by “inversing” the lower bound (and vice versa). This is formalized next.

Definition 6. Let $O_{\mathcal{L}} : \mathcal{L} \to \mathcal{L}$ and $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$.

- $\mathcal{O}$ is an *approximation* of $O_{\mathcal{L}}$, if $\forall x, y, \in \mathcal{L}$, $\mathcal{O}(x,y) = (\mathcal{O}_l(x,y), \mathcal{O}_u(x,y))$, where $\mathcal{O}_l : \mathcal{L}^2 \to \mathcal{L}$ and $\mathcal{O}_u : \mathcal{L}^2 \to \mathcal{L}$ are a lower and upper bound, respectively, of $O_{\mathcal{L}}(x)$ and $O_{\mathcal{L}}(y)$, namely: $\mathcal{O}_l(x,y) \leq O_{\mathcal{L}}(x)$ and $\mathcal{O}_u(x,y) \geq O_{\mathcal{L}}(y)$.

- $\mathcal{O}$ is *symmetric* if $\mathcal{O}(x,y) = (\mathcal{O}_l(x,y), \mathcal{O}_l(y,x))$ for some $\mathcal{O}_l : \mathcal{L}^2 \to \mathcal{L}$; $\mathcal{O}$ is $\leq_i$*-monotone*, if whenever $(x_1,y_1) \leq_i (x_2,y_2)$, also $\mathcal{O}(x_1,y_1) \leq_i \mathcal{O}(x_2,y_2)$; and $\mathcal{O}$ is *approximating*, if it is both symmetric and $\leq_i$-monotone.

In [7] it is shown that the *stable operator*, as defined next, can be used for expressing the semantics of many non-monotonic formalisms.

Definition 7. Let $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ be an approximation operator.

- $\mathcal{O}_l(\cdot,y) = \lambda x.\mathcal{O}_l(x,y)$, i.e., for $x \in \mathcal{L}$, $\mathcal{O}_l(\cdot,y)(x) = \mathcal{O}_l(x,y)$.
- $C(\mathcal{O}) : \mathcal{L} \to \mathcal{L}$, *the complete stable operator for* $\mathcal{O}$, is defined, for every $y \in \mathcal{L}$, by: $C(\mathcal{O})(y) = lfp(\mathcal{O}_l(.,y)) = \min_{\leq}\{x \in \mathcal{L} \mid x = \mathcal{O}_l(x,y)\}$.
- $S(\mathcal{O}) : \mathcal{L}^2 \to \mathcal{L}^2$, *the stable operator for* $\mathcal{O}$, is $S(\mathcal{O})(x,y) = (C(\mathcal{O})(y), C(\mathcal{O})(x))$.

Stable operators capture the idea of minimizing truth in the sense that for any $\leq_i$-monotone operator $\mathcal{O}$ on $\mathcal{L}^2$, fixpoints of the stable operator $S(\mathcal{O})$ are $\leq_t$-minimal fixpoints of $\mathcal{O}$ (see [7, Theorem 4]).

Accordingly, the following notions are defined in [7]:

Kripke-Kleene fixpoint of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid (x,y) = lfp_{\leq_i}(\mathcal{O}(x,y))\}$
well-founded semantics of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid (x,y) = lfp_{\leq_i}(S(\mathcal{O})(x,y))\}$
three-valued stable models of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid S(\mathcal{O})(x,y) = (x,y)\}$
two-valued stable models of $\mathcal{O}$:	$\{(x,x) \in \mathcal{L}^2 \mid S(\mathcal{O})(x,x) = (x,x)\}$

These semantical notions have been shown to provide a uniform framework for the mechanisms underlying many major knowledge representation formalisms, such as logic programming [17], autoepistemic logic [8], default logic [8], abstract argumentation [27] and abstract dialectical frameworks [27]. In more detail, for autoepistemic and default logics, the lattice of possible-world structures is investigated, and operators for autoepistemic logic and default logic give rise to characterizations of various formalisms, including autoepistemic expansions [16], well-founded semantics for default logic [2], weak default extensions [15], and Reiter's default extensions [20]. Full details can be found in [8].

2.3. Propositional Logic Programming (LP)

We now review some notions from logic programming theory that are needed in what follows. For simplicity and due to lack of space, we restrict our attentions to the propositional case. Following [17,25], a *generalized logic program* $\mathcal{P}$ is a finite set of the rules of the form $p \leftarrow \psi$, where p is an atom and ψ is a formula. A rule is *normal* if ψ is a conjunction of literals (that is, a conjunction of atomic formulas or negated atoms). A program is normal if it consists only of normal rules.

Given a four-valued lattice $\mathsf{F} \leq_t \mathsf{U}, \mathsf{B} \leq_t \mathsf{T}$ and a $\leq_t$-involution $-$ on it (i.e, $-\mathsf{F} = \mathsf{T}$, $-\mathsf{T} = \mathsf{F}$, $-\mathsf{U} = \mathsf{U}$ and $-\mathsf{B} = \mathsf{B}$), a four-valued *interpretation* of a generalized program $\mathcal{P}$ is a pair (x,y), where x is the set of the atoms that are assigned a value in $\{\mathsf{T},\mathsf{B}\}$ and y is the set of atoms assigned a value in $\{\mathsf{T},\mathsf{U}\}$. An interpretation (x,y) is *consistent* if $x \subseteq y$ (i.e., it doesn't have B-assignments). Truth assignments to complex formulas are then recursively defined as follows:

- $(x,y)(\phi) = \begin{cases} \mathsf{T} & \text{if } \phi \in x \text{ and } \phi \in y \\ \mathsf{U} & \text{if } \phi \notin x \text{ and } \phi \in y \\ \mathsf{F} & \text{if } \phi \notin x \text{ and } \phi \notin y \\ \mathsf{B} & \text{if } \phi \in x \text{ and } \phi \notin y \end{cases}$
- $(x,y)(\neg\phi) = -(x,y)(\phi)$
- $(x,y)(\psi \wedge \phi) = \min_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$
- $(x,y)(\psi \vee \phi) = \max_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$

The *immediate consequence operator* $\Phi_{\mathcal{P}}$ of $\mathcal{P}$ is now defined as follows:

$$\Phi_{\mathcal{P}}(x,y) = (\Phi^l_{\mathcal{P}}(x,y), \Phi^u_{\mathcal{P}}(x,y))$$

- $\Phi^l_{\mathcal{P}}(x,y) = \{\phi \in \mathsf{Atoms} \mid \text{ there is some } \phi \leftarrow \psi \in \mathcal{P}, (x,y)(\psi) \in \{\mathsf{T},\mathsf{B}\}\}$,
- $\Phi^u_{\mathcal{P}}(x,y) = \{\phi \in \mathsf{Atoms} \mid \text{ there is some } \phi \leftarrow \psi \in \mathcal{P}, (x,y)(\psi) \in \{\mathsf{T},\mathsf{U}\}\}$.

Remark 3. It can be easily seen that equivalently, one can define the immediate consequence operator $\Phi_{\mathcal{P}}$ by: $\Phi_{\mathcal{P}}(x,y) = (x',y')$, where for any atom ϕ,

$$(x',y')(\phi) = \max_{\leq_t}\{(x,y)(\psi) \mid \phi \leftarrow \psi \in \mathcal{P}\}.$$

We furthermore note that, alternatively, $\Phi^u_{\mathcal{P}}(x,y)$ can be taken as $\Phi^l_{\mathcal{P}}(y,x)$.

Note that $\Phi_{\mathcal{P}}$ is an operator on the lattice of the four-valued interpretations of $\mathcal{P}$. We therefore can define the following semantics for $\mathcal{P}$ in terms of the fixpoint notions considered in the previous section:

Definition 8. Given a generalized program $\mathcal{P}$, we say that a consistent interpretation (x,y) is:

- a *partial stable model* of $\mathcal{P}$, iff (x,y) is a three-valued stable model of $\Phi_{\mathcal{P}}$.
- a *total stable model* of $\mathcal{P}$, iff (x,y) is a two-valued stable model of $\Phi_{\mathcal{P}}$.
- the *well-founded model* of $\mathcal{P}$, iff (x,y) is the well-founded model of $\Phi_{\mathcal{P}}$.

In [17] it is shown that for normal logic programs the partial stable models coincide with the three-valued semantics as defined by [19], the well-founded model coincides with the homonymous semantics as defined by [19,28], and the total stable models coincide with the two-valued (or total) stable models of $\mathcal{P}$.

Example 2. Consider the program $\mathcal{P} = \{q \leftarrow \neg p;\ p \leftarrow \neg q;\ r \leftarrow \neg q;\ r \leftarrow \neg r\}$ (see Example 1). The bilattice of interest is formed by all pairs of subsets of $\{p,q,r\}$.

The partial stable models of $\mathcal{P}$ are $(\emptyset, \{p,q,r\})$, $(\{q\},\{q,r\})$, and $(\{p,r\},\{p,r\})$. In this case, $(\emptyset,\{p,q,r\})$ is well-founded and $(\{p,r\},\{p,r\})$ is total stable.

3. Argumentative Reflections

We now show how non-monotonic formalisms in general, and LP in particular, may be reflected by argumentation frameworks. First, we review some existing results concerning the correspondence between semantical notations in LP and ABA, and then we show how these results may be carried on to further types of LP semantics and other forms of nonmonotonic formalisms, using argumentative reflections of approximated fixpoint concepts.

3.1. Argumentative Characterizations of Logic Programs

The translation of logic programs into assumption-based argumentation has been the subject of several publications (e.g., [5,10,14,23]). The basic idea underlying all of these works is the same: the set of assumptions is made up of negated atoms and the contrary of a negated atom is the positive atom. For such translations it is shown that several argumentation semantics can characterize LP models. For instance, in [5] it is shown that for normal logic programs, complete extensions correspond to partial stable models, the grounded extension corresponds to the well-founded model, preferred extensions correspond to $\leq_i$-maximal partial stable models (also called 'regular'), and stable extensions correspond to two-valued stable models.

The results above are extended in [14] to disjunctive logic programming under stable model semantics. Furthermore, argumentative characterizations of the so-called well-justified [25] and well-founded [29] semantics of general or first-order logic programs with aggregates are provided in [10]. These generalizations are again based on similar representation methods: the assumptions consist of negated atoms and attacks are initiated when the attacking set allows to derive the positive version of the attacked (negated) atom. What changes, however, is the reasoning frame used to determine initiation of attacks. For example, in [14] the reasoning frame is supplemented with rules ensuring the adequate treatment of disjunction. Likewise, in [10], any valid first-order deduction rule is applicable.

Example 3. Consider again the assumption-based framework in Example 1. This is in fact a translation, according to the description of [5] above, of the logic program in Example 2. Indeed, the following semantic elements, indicated in Examples 1 and 2, correspond to the equivalences listed below (For instance, the stable extension of ABF is obtained by the (negation of the) complement of the total stable model of $\mathcal{P}$):

ABF		$\mathcal{P}$	
complete	$\emptyset$, $\{\neg p\}$, $\{\neg q\}$	partial stable	$(\emptyset,\{p,q,r\})$, $(\{q\},\{q,r\})$, $(\{p,r\},\{p,r\})$
grounded	$\emptyset$	well-founded	$(\emptyset,\{p,q,r\})$
stable	$\{\neg q\}$	total stable	$(\{p,r\},\{p,r\})$
peferred	$\{\neg p\}$, $\{\neg q\}$	$\leq_i$-max. stb	$(\{q\},\{q,r\})$, $(\{p,r\},\{p,r\})$

Despite these recent efforts, several questions still remain open. For example, several semantics for disjunctive logic programming have not yet been characterized in assumption-based argumentation. Likewise, three-valued stable models have not been characterized for first-order logic programs with aggregates.

3.2. Argumentative Reflections of Approximated Fixpoint Concepts

We now generalize the ideas discussed previously and show that for any operator over an underlying lattice one may compute its *argumentative reflection*. By this, it will be possible to show a correspondence between the semantical notions from approximation fixpoint theory and those of argumentation-based semantics, provided that the attack relation adequately reflects the operator in question.

The lattice under consideration in what follows is of the form $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$, where $\mathcal{A}$ is some nonempty set.[6] We then denote by $\overline{\mathcal{A}} = \{\overline{A} \mid A \in \mathcal{A}\}$ the set of argumentative reflections of the elements in $\mathcal{A}$. Intuitively and in accordance with the argumentative characterizations described above, $\overline{A}$ can be interpreted as some kind of *absence* of A.[7] Depending on the exact context, this absence can be assumption of falsity, failure to prove, etc. Accordingly, we denote:

- if $\Delta \subseteq \mathcal{A}$, then: $\overline{\Delta} = \{\overline{A} \mid A \in \Delta\}$ and $\sim\Delta = \mathcal{A} \setminus \Delta$.
- if $\Delta \subseteq \overline{\mathcal{A}}$, then: $\underline{\Delta} = \{A \in \mathcal{A} \mid \overline{A} \in \Delta\}$ and $\sim\Delta = \overline{\mathcal{A}} \setminus \Delta$.

We shall say that the lattice $L_{\overline{\mathcal{A}}} = (\wp(\overline{\mathcal{A}}), \subseteq)$ is the *reflection* of $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$. In what follows we shall assume that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ and that $\overline{A} \neq \overline{B}$ for every distinct $A, B \in \mathcal{A}$.

Remark 4. The assumption that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ is meant to assure that the resulting reflecting assumption-based frameworks (Definition 10) will be flat. This assumption holds for the translations of LP discussed above, as well as for the translation of default logic in ABA (see [3]). For normal logic programs, such an assumption is automatically satisfied, since heads of rules contain only positive atoms. When moving to general logic programs, [10] introduces for every atom a new element $\overline{A}$ (originally denoted $\mathsf{not}\, A$) such that $\mathsf{not}\, A \vdash \neg A$. Here we follow a similar idea for a more general case. For the translation of autoepistemic logic from [3], the assumption that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ is not warranted. The generalization of the results in this paper for such formalisms is left for future work.

We now turn to the primary concept of representations by argumentation frameworks. The underlying idea is to assume the 'absence' ($\overline{A}$) of any $A \in \mathcal{A}$, unless, on the basis of $C(\mathcal{O})$, some set of assumptions indicates that A holds. Thus, the complete stable operator $C(\mathcal{O})$ should be reflected in the attack relations.

Definition 9. The *argumentative reflection* of an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is given by the framework $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \rightsquigarrow \rangle$, where $\Delta \rightsquigarrow \overline{A}$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$.

Note that since $\overline{\Delta}$ and $\underline{\Delta}$ are complementary operators on Δ (that is, $\overline{\underline{\Delta}} = \Delta$), moving back and forth between a lattice $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$ and an argumentative reflection $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ of an approximation operator $\mathcal{O}$ on $(L_{\mathcal{A}})^2$, is straightforward. In particular, we have:

[6] For instance, $\mathcal{A}$ may be the set of the atomic formulas appearing in a logic program.

[7] For instance, as indicated in the previous section, in LP reflections are the negated atoms of the atoms of the logic program, that is:, $\overline{\mathcal{A}} = \{\neg A \mid A \in \mathcal{A}\}$, where $\neg$ may be a 'negation as failure' connective.

Lemma 1. *Suppose that $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \rightsquigarrow \rangle$ reflects $\mathcal{O} : (L_{\mathcal{A}})^2 \rightarrow (L_{\mathcal{A}})^2$. Then:*

a) for every $\Delta \subseteq \wp(\overline{\mathcal{A}})$ it holds that $\overline{\sim(\underline{\sim\Delta})} = \Delta$ and for every $\Delta \subseteq \wp(\mathcal{A})$ it holds that $\underline{\sim(\overline{\sim\Delta})} = \Delta$,

b) if (x,y) is a three-valued stable model of $\mathcal{O}$, then $x = \underline{(\overline{\sim y})^{+}}$.

Proof. We show the first part of (a) (the other one is similar): Since $\underline{\sim\Delta} = \{A \in \mathcal{A} \mid \overline{A} \notin \Delta\}$, we have that $\overline{\sim(\underline{\sim\Delta})} = \{\overline{A} \mid A \notin \{A \in \mathcal{A} \mid \overline{A} \notin \Delta\}\} = \{\overline{A} \mid \overline{A} \in \Delta\} = \Delta$.

For (b), note that since (x,y) is stable, $x = lfp(\mathcal{O}_l(.,\underline{\sim y}) = C(\mathcal{O})(\underline{\sim y})$. Since $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects $\mathcal{O}$, this means that $(\overline{\sim y})^{+} = \{\overline{A} \mid A \in x\}$. Thus, $\underline{(\overline{\sim y})^{+}} = x$. □

Note that Item (b) of the lemma indicates that when we are given an argumentation framework that reflects $\mathcal{O}$, only the y-component determines the stable models of $\mathcal{O}$.

Now we can state the next result, which is the meta-theoretical basis of all the other propositions that follow.

Proposition 1. *Suppose that $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \rightarrow (L_{\mathcal{A}})^2$. Then Δ is a complete extension of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ iff $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ is a consistent three-valued stable model of $\mathcal{O}$.*

Proof. We show that the condition is necessary for being Δ a complete extension of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$, omitting the other direction due to space restrictions. Suppose that $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ is a three-valued stable model, we show that Δ is complete. First, we note that, by Item (b) of Lemma 1, $\underline{\Delta^{+}} = \underline{(\overline{\sim\Delta})^{+}}$, and so: $(\star)$ $\Delta = \overline{\sim\Delta}$.
(1) Conflict-freeness: Since $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ is consistent, $\underline{\Delta^{+}} \subseteq \underline{\sim\Delta}$, and so $\Delta^{+} \subseteq \sim\Delta$. Thus Δ attacks only elements in its complementary set.
(2) Admissibility: Suppose that there is a set $\Theta \subseteq \overline{\mathcal{A}}$ such that $\Theta \rightsquigarrow \overline{A}$ for some $\overline{A} \in \Delta$. By $(\star)$, $\overline{A} \in \overline{\sim\Delta}$, so in particular, $A \notin \underline{\Delta}$. This means with the stability of $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ that $A \notin lfp(\mathcal{O}_l(.,\underline{\Delta^{+}})$, i.e., $A \notin C(\mathcal{O})(\underline{\Delta^{+}})$. Since $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects $\mathcal{O}$, this means that $\overline{\sim(\underline{\Delta^{+}})} \not\rightsquigarrow \overline{A}$. By definition, $\overline{\sim(\underline{\Delta^{+}})} = \{\overline{A} \mid \overline{A} \notin \Delta^{+}\} = \overline{\mathcal{A}} \setminus \Delta^{+}$. Thus, $\overline{\sim(\underline{\Delta^{+}})} \not\rightsquigarrow \overline{A}$ means that $\overline{\mathcal{A}} \setminus \Delta^{+} \not\rightsquigarrow \overline{A}$, which implies that if for some $\Gamma \subseteq \overline{\mathcal{A}}$, $\Gamma \rightsquigarrow \overline{A}$, then $\Gamma \cap \Delta^{+} \neq \emptyset$. Thus, $\Delta^{+} \cap \Theta \neq \emptyset$, which means that Θ is attacked by Δ, and so Δ defends $\overline{A}$.
(3) Completeness: Similar to the proof of admissibility. □

Proposition 2. *It $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \rightarrow (L_{\mathcal{A}})^2$, then:*

1. *(x,y) is the well-founded model of $\mathcal{O}$ iff $\overline{\sim y}$ is the grounded extension of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.*
2. *(x,y) is a stable model of $\mathcal{O}$ iff $\overline{\sim y}$ is a two-valued stable model of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.*
3. *(x,y) is a $\leq_i$-maximal three-valued stable model of $\mathcal{O}$ iff $\overline{\sim y}$ is a preferred extension of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.*

Proof. We show one direction of the first item. The proofs of the other claims are similar. For the proof we need the following two lemmas:

Lemma 2. *Suppose that $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \rightarrow (L_{\mathcal{A}})^2$. Let (x_1,y_1) and (x_2,y_2) be three-valued stable models of $\mathcal{O}$. Then $(x_1,y_1) \leq_i (x_2,y_2)$ iff $\overline{\sim y_1} \cup (\overline{\sim y_1})^{+} \subseteq \overline{\sim y_2} \cup (\overline{\sim y_2})^{+}$.*

Lemma 3. *Let $\mathcal{O}$ be an approximating operator over the lattice $(L_{\mathcal{A}})^2$ and let $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \rightsquigarrow \rangle$ be the argumentative reflection of $\mathcal{O}$. Then $\rightsquigarrow$ is monotonic: If $\Delta \rightsquigarrow \overline{A}$ and $\Delta \subseteq \Theta$, then $\Theta \rightsquigarrow \overline{A}$.*

Suppose now that (x,y) is the well-founded model of $\mathcal{O}$. By Proposition 1 and since the well-founded model is three-valued stable model, $\overline{\sim y}$ is complete. Suppose now that there is some $\Delta \subset \overline{\sim y}$ such that Δ is complete. By Proposition 1, $(\underline{\Delta^+}, \underline{\sim\Delta})$ is a three-valued stable model of $\mathcal{O}$. By Lemma 3, $\Delta \subset \overline{\sim y}$ implies $\Delta^+ \subseteq (\overline{\sim y})^+$. By Lemma 2, this implies that $(\underline{\Delta^+}, \underline{\sim\Delta}) \leq_i (x,y)$. But since $(\underline{\Delta^+}, \underline{\sim\Delta})$ is stable, (x,y) cannot be well-founded. □

4. Applications

In the first part of this section (Section 4.1) we demonstrate the usefulness of the results in Section 3.2 by showing how to obtain an assumption-based framework whose argumentation framework constitutes an argumentative reflection of a given operator. Then, in Section 4.2 we illustrate this in detail using propositional logic programs as defined in Section 2.3.

4.1. Assumption-based Argumentative Reflection

We show how to obtain an ABF whose argumentation framework is an argumentative reflection (Definition 9) of a given operator $\mathcal{O}$. First, we define an appropriate reasoning frame:

Lemma 4. *Let $\mathcal{O}$ be an approximating operator over the lattice $(L_{\mathcal{A}})^2$. Consider the pair $\mathfrak{L}_{\mathcal{O}} = \langle \mathcal{L}, \vdash \rangle$, where $\mathcal{L}$ includes both $\mathcal{A}$ and $\overline{\mathcal{A}}$, and $\vdash$ is defined for $\Delta \subseteq \overline{\mathcal{A}}$ by $\Delta \vdash A$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. Then $\mathfrak{L}_{\mathcal{O}}$ is a reasoning frame, i.e., $\vdash$ is monotonic.*

Definition 10. The *assumption-based argumentative reflection* of an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is given by the assumption-based argumentation framework $\mathsf{ABF}_{\mathcal{A},\mathcal{O}} = \langle \mathfrak{L}_{\mathcal{O}}, \Lambda, - \rangle$, where $\mathfrak{L}_{\mathcal{O}}$ is the reasoning frame defined in Lemma 4, $\Lambda = \overline{\mathcal{A}}$, and the contrariness operator is defined for every $\overline{A} \in \overline{\mathcal{A}}$ by $-\overline{A} = A$.

Proposition 3. *If $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is approximating (in the sense of Definition 6), then $\mathcal{AF}(\mathsf{ABF}_{\mathcal{A},\mathcal{O}})$ (Definition 4 and 10) reflects $\mathcal{O}$.*

Proof. We have to show that for any $\Delta \cup \{\overline{A}\} \subseteq \overline{\mathcal{A}}$, Δ attacks $\overline{A}$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. By definition, Δ attacks $\overline{A}$ iff $\Delta \vdash -\overline{A}$. Since $-\overline{A} = A$, Δ attacks $\overline{A}$ iff $\Delta \vdash A$. By the definition of $\mathsf{ABF}_{\mathcal{A},\mathcal{O}}$, $\Delta \vdash A$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. □

4.2. Example: Propositional Logic Programming

We now illustrate how the results shown in the previous section can be applied to obtain argumentative characterizations of logic programs as defined in Section 2.3.

Theorem 1. *Let $\mathcal{P}$ be a logic program and let $\overline{\mathsf{Atoms}(\mathcal{P})} = \{\overline{A} \mid A \in \mathsf{Atoms}(\mathcal{P})\}$. Consider the assumption-based framework $\mathsf{ABF}_\mathcal{P} = \langle \mathfrak{L}_\mathcal{P}, \overline{\mathsf{Atoms}(\mathcal{P})}, -\rangle$, where $-\overline{A} = A$ and $\mathfrak{L}_\mathcal{P} = \langle \mathcal{L}_\mathcal{P}, \vdash_\mathcal{P}\rangle$ is a reasoning frame in which $\mathcal{L}_\mathcal{P}$ includes $\overline{\mathsf{Atoms}(\mathcal{P})}$ and the closure under $\neg, \wedge, \vee$ of $\mathsf{Atoms}(\mathcal{P})$, and $\Delta \vdash_\mathcal{P} \phi$ iff $\phi \in C(\Psi^l_\mathcal{P})(\underline{\sim\Delta})$. For a set $\Delta \subseteq \overline{\mathsf{Atoms}(\mathcal{P})}$ we denote $\Delta^* = (\underline{\Delta^+}, \underline{\sim\Delta})$. Then:*

1. *Δ is a complete extension of $\mathsf{ABF}_\mathcal{P}$ iff Δ^* is a partial stable model of $\mathcal{P}$.*
2. *Δ is the grounded extension of $\mathsf{ABF}_\mathcal{P}$ iff Δ^* is the well-founded model of $\mathcal{P}$.*
3. *Δ is a stable extension of $\mathsf{ABF}_\mathcal{P}$ iff Δ^* is a total stable model of $\mathcal{P}$.*
4. *Δ is a preferred extension of $\mathsf{ABF}_\mathcal{P}$ iff Δ^* is a $\leq_i$-maximal partial stable model of $\mathcal{P}$.*

Proof. Since $\mathsf{ABF}_\mathcal{P} = \mathsf{ABF}_{\mathsf{Atoms}(\mathcal{P}), \Psi^l_\mathcal{P}}$, by Proposition 3, $\mathcal{AF}(\mathsf{ABF}_\mathcal{P})$ reflects $\Psi^l_\mathcal{P}$. Thus, Item 1 is obtained by Proposition 1 and Definition 8. Items 2, 3, and 4 are obtained in a similar way, using respectively Items 1, 2 and 3 of Proposition 2. □

Remark 5. The results of Theorem 1 were already shown for grounded and stable extensions (i.e., Items 2 and 3) in [10].[8] On the other hand, the correspondence between complete and partial stable extensions (and the analogous correspondence for preferred extensions) was left in [10] as a conjecture. We are now able to confirm this conjecture with Theorem 1.

Remark 6. For normal logic programs $C(\Psi^l_\mathcal{P}(\sim\Delta))$ is nothing but the consequence operator $\vdash$ based on Modus Ponens from Example 1. Furthermore, for the general case of propositional logic programs, it can be shown that for consistent sets of assumptions, $C(\Psi^l_\mathcal{P}(\sim\Delta))$ is the consequence operator that satisfies Modus Ponens with respect to the rules in the logic program and every valid inference for classical logic (this is the consequence operator used in [10] as described in Section 3.2).

Example 4. Theorem 1 can be illustrated by revisiting Example 1. Indeed, there $\mathsf{ABF}_\mathcal{P}$ is the assumption-based argumentative reflection of $\Psi^l_\mathcal{P}$ (as in Theorem 1), when restricted to normal logic programs. We see, then, that the semantic equivalences observed in Example 3 are not a coincidence, since they follow from Theorem 1.

Argumentative characterizations similar to those in Theorem 1 can be obtained as corollaries of our results for many other variants of logic programs (such as first-order logic programs, logic programing with aggregates and HEX-programs). Furthermore, the generalization of an argumentative characterizations of semantical alternatives to Reiter's extensions for default logic can also be obtained as a corollary of our main results and the characterization of default logic in approximation fixpoint theory from [8], thus significantly extending the argumentative characterization of Reiter's default extensions in [3].

[8] In fact, in [10] this correspondence is shown for first-order logic programs where the head of a rule can be any propositional formula. Due to space limitations we have restricted ourselves to the propositional case where heads of rules are literals.

5. Conclusion, In View of Related Work

Our results allow for translations of any non-monotonic formalism which has received characterizations in approximation fixpoint theory (and gives rise to flat assumption-based reflections). This includes default logic under the semantics discussed in Section 2.2, as well as many families of logic programming languages under various semantics like those in Section 2.3.

To the best of our knowledge, a general methodology for argumentative characterizations of non-monotonic formalisms has not been suggested before. The connections between approximation fixpoint theory and abstract argumentation where investigated already in [26], where it was shown that abstract dialectical frameworks [4] and Dung's abstract argumentation [9] can be characterized using approximation fixpoint theory. Even though the results of this paper are in a sense complementary to those in [26] (that is, that argumentation theory can capture approximation fixpoint theory), the goal of our paper is somewhat orthogonal: the argumentative characterization of approximation fixpoint theory is to be seen as a mean to obtain a multitude of results on argumentative characterizations of non-monotonic formalisms.

The translation of non-monotonic formalisms into assumption-based argumentation is not only interesting from a theoretical point of view, but also allows for importing methods from one formalism to the other. For example, argumentative characterizations of logic programming have been proven useful for explanation [22,24] and visualization [21] of inferences in logic programming. Our results now open the door for the applications of such techniques to any of the formalisms discussed in this paper. Furthermore, extensions of assumption-based argumentation, such as the integration of priorities [6], can now be combined with the translated formalisms.

In future work, we plan to extend our results to approximation operators that give rise to non-flat ABFs (such as those for autoepistemic logic) and consider non-deterministic operators [18].

References

[1] Christian Antić, Thomas Eiter, and Michael Fink. Hex semantics via approximation fixpoint theory. In *Proceedings of LPNMR'13*, LNCS 8148, pages 102–115. Springer, 2013.

[2] Chitta R Baral and V. S. Subrahmanian. Dualities between alternative semantics for logic programming and nonmonotonic reasoning. *Journal of Automated Reasoning*, 10(3):399–420, 1993.

[3] Andrei Bondarenko, Phan Minh Dung, Robert Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63–101, 1997.

[4] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proceedings of KR'10*. AAAI Press, 2010.

[5] Martin Caminada and Claudia Schulz. On the equivalence between assumption-based argumentation and logic programming. *Artificial Intelligence Research*, 60:779–825, 2017.

[6] Kristijonas Cyras and Francesca Toni. ABA+: assumption-based argumentation with preferences. In *Proceedings of KR'16*. AAAI Press, 2016.

[7] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In *Logic-based*

Artificial Intelligence, volume 597 of *The Springer International Series in Engineering and Computer Science*, pages 127–144. Springer, 2000.

[8] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence*, 143(1):79–122, 2003.

[9] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[10] Phan Minh Dung, Tran Cao Son, and Phan Minh Thang. Argumentation-based semantics for logic programs with first-order formulae. In *Proceedings of PRIMA'16*, LNCS 9862, pages 43–60. Springer, 2016.

[11] Melvin Fitting. Bilattices are nice things. In *Self Reference*, volume 178 of *CSLI Lecture Notes*, pages 53–77. CLSI Publications, 2006.

[12] Matthew L. Ginsberg. Multi-valued logics: A uniform approach to reasoning in artificial intelligence. *Computer Intelligence*, 4:256–316, 1988.

[13] Jesse Heyninck and Ofer Arieli. On the semantics of simple contrapositive assumption-based argumentation frameworks. In *Proceedings of COMMA'18*, Frontiers in Artificial Intelligence and Applications 305, pages 9–20. IOS Press, 2018.

[14] Jesse Heyninck and Ofer Arieli. An argumentative characterization of disjunctive logic programming. In *Proceedings of EPIA'19*, LNCS 11805, pages 526–538. Springer, 2019.

[15] Victor Marek and Miroslaw Truszczynski. Relating autoepistemic and default logics. In *Proceedings of KR'89*, pages 276–288. Morgan Kaufmann, 1989.

[16] Robert Moore. Semantic considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[17] Nikolay Pelov, Marc Denecker, and Maurice Bruynooghe. Well-founded and stable semantics of logic programs with aggregates. *Theory and Practice of Logic Programming*, 7(3):301–353, 2007.

[18] Nikolay Pelov and Miroslaw Truszczynski. Semantics of disjunctive programs with monotone aggregates – an operator-based approach. In *Proceedings of NMR'04*, pages 327–334, 2004.

[19] Teodor C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.

[20] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[21] Claudia Schulz. Graphical representation of assumption-based argumentation. In *Proceedings of AAAI'15*, pages 4204–4205. AAAI Press, 2015.

[22] Claudia Schulz, Ken Satoh, and Francesca Toni. Characterising and explaining inconsistency in logic programs. In *Proceedings of LPNMR'15*, LNCS 9345, pages 467–479. Springer, 2015.

[23] Claudia Schulz and Francesca Toni. Logic programming in assumption-based argumentation revisited-semantics and graphical representation. In *Proceedings of AAAI'15*, pages 1569–1575. AAAI Press, 2015.

[24] Claudia Schulz and Francesca Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59âĂŞ110, 2016.

[25] Yi-Dong Shen, Kewen Wang, Thomas Eiter, Michael Fink, Christoph Redl, Thomas Krennwallner, and Jun Deng. FLP answer set semantics without circular justifications for general logic programs. *Artificial Intelligence*, 213:1–41, 2014.

[26] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[27] Hannes Strass and Johannes Peter Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, 226:34–74, 2015.

[28] Allen Van Gelder, Kenneth A Ross, and John S Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, 1991.

[29] Yisong Wang, Fangzhen Lin, Mingyi Zhang, and Jia-Huai You. A well-founded semantics for basic logic programs with arbitrary abstract constraint atoms. In *Proceedings of AAAI'12*, pages 835–841. AAAI Press, 2012.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200507

An Epistemic Interpretation of Abstract Dialectical Argumentation

Jesse HEYNINCK [a,1], and Gabriele KERN-ISBERNER [a]
[a] *Department of Computer Science, TU Dortmund, Germany*

Abstract. Formal argumentation is a well-established and influential knowledge representation formalism that is at the center of recent developments in explainable artificial intelligence. Many extensions to formal argumentation have been proposed, and to cope with the multiplicity of such generalizations, *abstract dialectical frameworks* (in short, ADFs) have been proposed by Brewka and Woltran. This generality comes at a cost, since the semantics underlying ADFs are arguably not as transparent as those of abstract argumentation frameworks. This opacity is witnessed among others by revisions of several of the central semantics for abstract dialectical frameworks. In this paper, we intend to give a clear conceptual foundation of abstract dialectical frameworks by intepreting abstract dialectical frameworks in *epistemic logic*. In particular, we show how interpretations and their refinements can be straightforwardly embedded in epistemic logic as S5-structures that model the interpretation as knowledge. Given such an interpretation, it turns out that all major semantics for ADFs coincide with the possible world structures that are *autoepistemically sound* according to the seminal paper by Moore with respect to the theory expressed by the ADF.

Keywords. Autoepistemic Logic, Computational Argumentation, Abstract Dialectical Frameworks

1. Introduction

Formal argumentation is one of the major approaches to knowledge representation and has been heralded for its potential in explainable artificial intelligence (see e.g. [26]). In the seminal paper [8], *abstract argumentation frameworks* where conceived of as directed graphs where nodes represent arguments and edges between these nodes represent attacks. So-called *argumentation semantics* determine which sets of arguments can be reasonably upheld together given such an argumentation graph. Various authors have remarked that other relations between arguments are worth consideration. For example, in [6], *bipolar argumentation frameworks* are developed, where arguments can support as well as attack each other. The last decades saw a proliferation of such extensions of the original formalism of [8], and it has often proven hard to compare the resulting different dialects of the formal argumentation formalism. To cope with the result-

[1] Corresponding Author: Jesse Heyninck. Email: jesse.heyninck@tu-dortmund.de

ing multiplicity, [5,4] introduced *abstract dialectical argumentation* that aims to unify these different dialects. Just like in [8], *abstract dialectical frameworks* (in short, ADFs) are directed graphs. In contradistinction to abstract argumentation frameworks, however, in ADFs, edges between nodes do not necessarily represent attacks but can encode *any* relationship between arguments. Such a generality is achieved by associating an *acceptance condition* with each argument, which is a boolean formula in terms of the parents of the argument that expresses the conditions under which an argument can be accepted. As such, ADFs are able to capture all of the major extensions of abstract argumentation and offer a general framework for argumentation based inference. This generality arguably results in a loss of transparency of the semantics of ADFs. Such an opacity is witnessed by revisions of several of the central semantics for ADFs. For example, the stable semantics from [5] was revised in [4] because it did not adequately capture the stable model semantics from logic programming. Likewise, the admissible semantics received reformulations in [1] and [22] in view of both reasons of intuitiveness and representational adequacy. Such a lack of transparency is especially worrying given the ambitions of formal argumentation in contributing to explainable AI. Therefore, we make first steps towards a clear conceptual foundations of ADFs by interpreting ADFs in *epistemic logic*. In particular, we show how interpretations can be interpreted as S5-structures for the beliefs in the arguments accepted by the interpretations in question. Under such an interpretation, it turns out that all major semantics for ADFs coincide with the S5-structures that are *autoepistemically sound* according to [21] with respect to the knowledge expressed by the ADF.

Outline of the Paper In Section 2, we give preliminaries on propositional logic (Section 2.1), ADFs (Section 2.2) and epistemic and autoepistemic logic (Section 2.3). In Section 3, we reinterpret interpretations as S5-structures known from epistemic logic, and show that such an interpretation fulfills some basic sanity criteria. In Section 4 we show that such an interpretation can be used to translate ADFs into autoepistemic logic. In Section 5 we make some remarks about translating autoepistemic logic into ADFs. We end the paper by discussing related work (Section 6) and making some concluding remarks (Section 7).

2. Preliminaries

In the following, we briefly recall some general preliminaries on propositional logic as well as technical details on ADFs [4].

2.1. Propositional Logic

For a set At of atoms let $\mathcal{L}(\mathsf{At})$ be the corresponding propositional language constructed using the usual connectives $\wedge$ (*and*), $\vee$ (*or*), $\neg$ (*negation*) and $\rightarrow$ (*material implication*). A (classical) *interpretation* (also called *possible world*) ω for a propositional language $\mathcal{L}(\mathsf{At})$ is a function $\omega : \mathsf{At} \rightarrow \{\top, \bot\}$. Let $\Omega(\mathsf{At})$ denote the set of all interpretations for At. We simply write Ω if the set of atoms is implicitly given. An interpretation ω *satisfies* (or is a *model* of) an atom $a \in \mathsf{At}$, denoted by $\omega \models a$, if and only if $\omega(a) = \top$. The satisfaction relation $\models$ is extended to formulas as usual. As an abbreviation we sometimes identify an interpretation ω with its *complete conjunction*, i. e., if $a_1, \dots, a_n \in \mathsf{At}$ are those atoms that are assigned $\top$ by ω and $a_{n+1}, \dots, a_m \in \mathsf{At}$ are those atoms that are assigned $\bot$ by ω we identify ω by $a_1 \dots a_n \overline{a_{n+1}} \dots \overline{a_m}$ (or any permutation of this). For example, the interpretation ω_1 on $\{a, b, c\}$ with $\omega(a) = \omega(c) = \top$ and $\omega(b) = \bot$ is abbreviated by $a\bar{b}c$. For $\Phi \subseteq \mathcal{L}(\mathsf{At})$ we also define $\omega \models \Phi$ if and only if $\omega \models \phi$ for every $\phi \in \Phi$. Define the set of models $\mathsf{Mod}(X) = \{\omega \in \Omega(\mathsf{At}) \mid \omega \models X\}$ for every formula or set of formulas X. A formula or set of formulas X_1 *entails* another formula or set of formulas X_2, denoted by $X_1 \vdash X_2$, if $\mathsf{Mod}(X_1) \subseteq \mathsf{Mod}(X_2)$.

2.2. Abstract Dialectical Frameworks

We briefly recall some technical details on ADFs following loosely the notation from [4]. An ADF D is a tuple $D = (S, L, C)$ where S is a set of *statements*, $L \subseteq S \times S$ is a set of *links*, and $C = \{C_s\}_{s \in S}$ is a set of *acceptance functions*, which are total functions $C_s : 2^{par_D(s)} \to \{\top, \bot\}$ for each $s \in S$ with $par_D(s) = \{s' \in S \mid (s', s) \in L\}$. An acceptance function C_s defines the cases when the statement s can be accepted (truth value $\top$), depending on the acceptance status of its parents in D. By abuse of notation, we will often identify an acceptance function C_s with its equivalent *acceptance condition* which models the acceptable cases as a propositional formula $\phi \in \mathcal{L}(par_D(s))$.

Example 1. *We consider the following ADF $D_1 = (\{a, b, c\}, L, C)$ with:*
$L = \{(a, b), (b, a), (a, c), (b, c)\}$ *and:* $C_a = \neg b$, $C_b = \neg a$, $C_c = a \vee b$.
Informally, the acceptance conditions can be read as "a is accepted if b is not accepted", "b is accepted if a is not accepted" and "c is accepted if either a is accepted or b is accepted".

An ADF $D = (S, L, C)$ is interpreted through 3-valued interpretations $v : S \to \{\top, \bot, u\}$, which assign to each statement in S either the value $\top$ (true, accepted), $\bot$ (false, rejected), or u (unknown). A 3-valued interpretation v can be extended to arbitrary propositional formulas over S via Kleene semantics: $v(\neg\phi) = \bot[\top]$ iff $v(\phi) = \top[\bot]$, and $v(\neg\phi) = u$ iff $v(\phi) = u$. $v(\phi \wedge \psi) = \top$ iff $v(\phi) = v(\psi) = \top$, $v(\phi \wedge \psi) = \bot$ iff $v(\phi) = \bot$ or $v(\psi) = \bot$, and $v(\phi \wedge \psi) = u$ otherwise, and similarly for disjunction. $\mathcal{V}$ is the set of all three-valued interpretations.

Then $v \in \mathcal{V}$ is a *model* of D if for all $s \in S$, if $v(s) \neq u$ then $v(s) = v(C_s)$.

We define an order $\leq_i$ over $\{\top, \bot, u\}$ by making u the minimal element: $u <_i \top$ and $u <_i \bot$, and this order is lifted pointwise as follows (given two interpretations v, w over S): $v \leq_i w$ iff $v(s) \leq_i w(s)$ for every $s \in S$.[2] The set of two-valued interpretations extending an interpretation v is defined as $[v]^2 = \{\omega \in \Omega(S) \mid v \leq_i \omega\}$. Given a set of interpretations V, $\sqcap_i V(s) = v(s)$ if for every $v' \in V$, $v'(s) = v(s)$ and $\sqcap_i V(s) = u$ otherwise. $\Gamma_D(v) : S \to \{\top, \bot, u\}$ where $s \mapsto \sqcap_i \{\omega(C_s) \mid \omega \in [v]^2\}$.

Definition 1. *Let $D = (S, L, C)$ be an ADF with $v : S \to \{\top, \bot, u\}$ an interpretation:*

[2] Notice that, in general, a three-valued interpretation will be denoted with v whereas a two-valued interpretation is denoted with ω.

- v *is* complete for D *iff* $v = \Gamma_D(v)$.
- v *is* preferred for D *iff v is a $\leq_i$-maximally complete interpretation for D.*
- v *is* grounded for D *iff v is a $\leq_i$-minimally complete interpretation for D.*

We denote by $\mathsf{Cmp}(D)$, $\mathsf{Prf}(D)$ *respectively* $\mathsf{Grn}(D)$ *the sets of complete, preferred respectively grounded interpretations of D.*

Notice that any complete (and therefore preferred and grounded) interpretation of D is also a model of D. We finally define inference relations for ADFs:

Definition 2. *Given an ADF $D = (S, L, C)$ and $s \in S$ and* $\mathsf{sem} \in \{\mathsf{Prf}, \mathsf{Cmp}, \mathsf{Cmp}\}$, *we define:* $D \mid\!\sim^{\cap}_{\mathsf{sem}} s[\neg s]$ *iff* $v(s) = \top[\bot]$ *for all* $v \in \mathsf{sem}(D)$.[3]

Example 2 (Example 1 continued)**.** *The ADF of Example 1 has three complete models v_1, v_2, v_3 with: $v_1(a) = \top$, $v_1(b) = \bot$,$v_1(c) = \top$, $v_2(a) = \bot$, $v_2(b) = \top$,$v_2(c) = \top$, $v_3(a) = u$,$v_3(b) = u$,$v_3(c) = u$.*

v_3 is the grounded interpretation whereas v_1 and v_2 are both preferred.

2.3. Epistemic and Autoepistemic Logic

We recall the syntax and semantics of S5 [17]. We use $\mathbf{L}$ to denote the epistemic belief operator. By an epistemic language we mean any language $\mathcal{L}^{\mathbf{L}}$ such that $\mathbf{L}\phi \in \mathcal{L}^{\mathbf{L}}$ if $\phi \in \mathcal{L}^{\mathbf{L}}$. We denote $\mathcal{L}$ as the fragment of $\mathcal{L}^{\mathbf{L}}$ that contains all the formulas containing no occurence of the belief operator $\mathbf{L}$ and we shall from now on assume that $\mathcal{L}$ coincides with the language of propositional logic.

Definition 3. *Given Ω, a possible world structure over Ω is a set $Q \subseteq \Omega$.*

The set of all possible world structures is thus[4] $\wp(\Omega)$ and is a complete lattice under $\subseteq$. Such possible world structures can be used to model beliefs by interpretting a set of worlds Q as the states an agent considers as possible. This is the standard idea underlying the semantics of the modal logic S5 where entailment is defined as follows:

Definition 4. *Let $Q \cup \{\omega\} \subseteq \Omega$ and $\phi \in \mathcal{L}^{\mathbf{L}}$:*

- *for $\phi \in \mathsf{At}$, $Q, \omega \models \phi$ if $\omega \models \phi$*
- *$Q, \omega \models \mathbf{L}\phi$ if $Q, \omega' \models \phi$ for every $\omega' \in Q$*
- *$Q, \omega \models \phi \wedge \psi$ if $Q, \omega \models \phi$ and $Q, \omega \models \psi$*
- *$Q, \omega \models \neg\phi$ if $Q, \omega \not\models \phi$*

Finally, $Q, \omega \models \phi \rightarrow \psi$ iff $Q, \omega \models \neg\phi \vee \psi$ and $Q, \omega \models \phi \vee \psi$ iff $Q, \omega \models \neg(\neg\phi \wedge \neg\psi)$.

Example 3. *Consider the formula $\neg\mathbf{L}b \rightarrow a$ and the possible world structure $\{a\bar{b}, \bar{a}\bar{b}\}$. Observe for example that $\{a\bar{b}, \bar{a}\bar{b}\}, a\bar{b} \models \neg\mathbf{L}b \rightarrow a$ whereas $\{a\bar{b}, \bar{a}\bar{b}\}, \bar{a}\bar{b} \not\models \neg\mathbf{L}b \rightarrow a$.*

[3] Since the grounded extension is unique for any ADF [4], $\cap$ is ommited from $\mid\!\sim_{\mathsf{Grn}}$.

[4] Notice that we use $\wp$ as the power-set and not as the Weierstrass function.

[21] noticed that it is interesting to look at those possible world structures that represent "knowledge of a perfect, rational, introspective agent" [3]. In more detail, given a set of formulas $\Delta \subseteq \mathcal{L}^{\mathbf{L}}$, Moore suggests to look at those sets of possible worlds that model Δ and are closed under introspection. In terms of possible world structures, this translates to possible world structures that are fixpoints of the following operator (see [3]) (given $Q \subseteq \Omega$ and $\Delta \subseteq \mathcal{L}_{\mathbf{L}}$):

$$\Psi_\Delta(Q) = \{\omega \in \Omega \mid Q, \omega \models \bigwedge \Delta\}$$

Definition 5. *A set of worlds $Q \subseteq \Omega$ is an autoepistemic extension (in short, AEE) for $\Delta \subseteq \mathcal{L}^{\mathbf{L}}$ iff $\Psi_\Delta(Q) = Q$. An AEE Q is* consistent *iff $Q \neq \emptyset$.*

Example 4. *Let $\Delta = \{\neg\mathbf{L}b \to a; \neg\mathbf{L}a \to b\}$. We have the following autoepistemic extensions for Δ: $\{ab, \overline{a}b\}$ and $\{ab, a\overline{b}\}$. Notice that e.g. $\{\overline{a}\overline{b}\}$ is not an autoepistemic extension since $\{\overline{a}\overline{b}\}, \overline{a}\overline{b} \models \neg\mathbf{L}b \wedge \neg a$, i.e. $\{\overline{a}\overline{b}\}, \overline{a}\overline{b} \not\models \neg\mathbf{L}b \to a$. Therefore, $\overline{a}\overline{b} \notin \Psi_\Delta(\{\overline{a}\overline{b}\})$ and thus $\{\overline{a}\overline{b}\}$ does not constitute a fixed point under Ψ_Δ.*

In [21], a syntactic characterization of autoepistemic extensions was given as follows, which we recall for completeness:

Definition 6. *A* (syntactic) autoepistemic extension *of a set of autoepistemic formulas $\Delta \subseteq \mathcal{L}^{\mathbf{L}}$ is any theory $\mathcal{E} \subseteq \mathcal{L}^{\mathbf{L}}$ that satisfies (where $\phi \in \mathcal{L}^{\mathbf{L}}$):*

$$\mathcal{E} = Cn(\Delta \cup \{\mathbf{L}\phi \mid \mathcal{E} \vdash \phi\} \cup \{\neg\mathbf{L}\phi \mid \mathcal{E} \nvdash \phi\})$$

The syntactic characterization of autoepistemic extensions and the one in terms of possible worlds are equivalent (see e.g. [20]):

Theorem 1. *Given $\Delta \subseteq \mathcal{L}^{\mathbf{L}}$, $Q \subseteq \Omega$ is an autoepistemic extension of Δ iff $\{\phi \in \mathcal{L}^{\mathbf{L}} \mid \forall \omega \in Q : Q, \omega \models \phi\}$ is a syntactic autoepistemic extension of Δ.*

Furthermore, it will prove useful below to consider *maximally informative* and *minimally informative* autoepistemic extensions:[5]

Definition 7. *Given $\Delta \subseteq \mathcal{L}^{\mathbf{L}}$:*

- *$Q \subseteq \Omega$ is a* maximally informative AEE *iff it is an autoepistemic extension and there is no autoepistemic extension $Q' \subseteq \Omega$ s.t. $Q' \subset Q$.*
- *$Q \subseteq \Omega$ is a* minimally informative AEE *iff it is an autoepistemic extension and there is no autoepistemic extension $Q' \subseteq \Omega$ s.t. $Q' \supset Q$.*

We can define an inference relation based on autoepistemic logics as follows:

Definition 8. *Given an autoepistemic knowledge base Δ:*

- *$\Delta \mathrel{|\!\sim}^{\cap}_{AEL} \phi$ iff $\phi \in \mathcal{E}$ for every autoepistemic extension $\mathcal{E}$ of Δ.*
- *$\Delta \mathrel{|\!\sim}^{\cap,\mathsf{max}}_{AEL} \phi$ iff $\phi \in \mathcal{E}$ for every maximally informative AEE $\mathcal{E}$ of Δ.*
- *$\Delta \mathrel{|\!\sim}^{\cap,\mathsf{min}}_{AEL} \phi$ iff $\phi \in \mathcal{E}$ for every minimally informative AEE $\mathcal{E}$ of Δ.*

[5] Notice that a maximally informative AEE is $\subseteq$-minimal: this is so because we consider sets of worlds, and thus minimizing these sets means maximizing the informational content of these sets of worlds. Likewise, minimally informative AAEs are $\subseteq$-maximal.

3. An Epistemic Embedding of ADF-Interpretations

In ADFs, instead of restricting relations between arguments to attack or support, arguments can have *any* relation between each other. This abstraction is achieved by assigning acceptance conditions to arguments in terms of their parents. Given an ADF, semantics encode what are reasonable stances for an agent given the information encoded by an ADF in the following sense: a node can only be accepted if we have good reasons for accepting it, and having good reasons to accept a node means that we should accept the node in question. E.g. in Example 1, a can only be accepted if b is rejected, and likewise if b is rejected, a should be accepted. Formally speaking, the semantics of ADFs are based on 3-valued interpretations v over S. $v(s) = \top$ means that s is believed. Likewise, $v(s) = \bot$ encodes belief in s being false, whereas $v(s) = u$ encodes suspension of belief about s, i.e. neither believing s being true nor believing s being false. Epistemic logic allows us to give a straightforward epistemic embedding of a 3-valued interpretation. In more detail, given an ADF $D = (S, L, C)$ and 3-valued interpretation v over S, we can associate a possible world structure with v as follows:

Definition 9. *Let $D = (S, L, C)$ and $v \in \mathcal{V}$. We define $Q_v = \{\omega \in \Omega(S) \mid v \leq_i \omega\}$*

Under this interpretation, Q_v can be seen to be the set of all worlds that are possibilities (given v) for being the *actual world*. For example, if $v(s) = \top$, it will be the case that for every $\omega \in Q_v$, $\omega \models s$, i.e. in every candidate for the actual world, s is the case and consequently Q_v models belief in s. Likewise, if $v(s) = u$, there are candidates for the actual world where s is true and candidates for the actual world where s is false, and thus Q_v models neither belief in s nor belief in $\neg s$. One can observe that $Q_v = [v]^2$, i.e. the semantics of ADFs already implicitly assume possible world structures. The following result shows that $v(s) = \top[\bot]$ indeed corresponds to belief in s by Q_v:

Proposition 1. *For any interpretation $v \in \mathcal{V}$:*

- *$v(s) = \top$ iff $Q_v, \omega \models \mathbf{L}s$ (for any $\omega \in \Omega(S)$),*
- *$v(s) = \bot$ iff $Q_v, \omega \models \mathbf{L}\neg s$ (for any $\omega \in \Omega(S)$),*
- *$v(s) = u$ iff $Q_v, \omega \models \neg\mathbf{L}s \wedge \neg\mathbf{L}\neg s$ (for any $\omega \in \Omega(S)$),*

Proof. Suppose first that $v(s) = \top$. Then for every $\omega \in Q_v$, $\omega \models s$ and thus $Q_v, \omega \models \mathbf{L}s$. Suppose now that $Q_v, \omega \models \mathbf{L}s$, i.e. for every $\omega \in Q_v$, $\omega \models s$ and suppose towards a contradiction that $v(s) \neq \top$. But then there is an $\omega' \in \Omega(S)$ s.t. $v \leq_i \omega'$ and $\omega'(s) = \bot$. Since $\omega' \in Q_v$, this contradicts $Q_v, \omega \models \mathbf{L}s$. The other cases are analogous. □

The epistemic embedding of interpretations also allows for an intuitive analogue of the information ordering $\leq_i$ over $\mathcal{V}$. Recall that this ordering represents the amount of information represented by an interpretation v. Within our epistemic interpretation of $\mathcal{V}$, $v \leq_i v'$ means that the interpretation v' is commited to the same or more beliefs than v, i.e. whenever $Q_v, \omega \models \mathbf{L}\phi$ then $Q_{v'}, \omega \models \mathbf{L}\phi$ (for any $\omega \in \Omega$). This is the case when $Q_v \supseteq Q_{v'}$, i.e. the information $Q_{v'}$ gives us about the actual world is at least as specific as the information about the actual world given by Q_v. This intuition is vindicated by the following proposition (whose proof is straightforward and left out in view of spatial considerations):

Proposition 2. *$v \leq_i v'$ iff $Q_v \supseteq Q_{v'}$.*

4. Interpreting ADF-semantics in Autoepistemic Logic

In this section, we use the epistemic embedding of three-valued interpretations v over a set of nodes S to translate all of the major semantics for ADFs in autoepistemic logic. We first formulate a translation that is adequate for complete semantics. This translation allows us to show that preferred respectively grounded interpretations correspond to autoepistemic extensions that are maximally respectively minimally informative. In Section 4.2, we finally show that the translation fulfills some desirable properties.

4.1. Translating ADFs into Autoepistemic Logic

The basic idea behind our translation is the following: believing a condition C_s of a node s means that the node must be true, which formally translates as the premise $\mathbf{L}C_s \rightarrow s$. Likewise, believing the condition C_s is false means that the node must be false (i.e. $\mathbf{L}\neg C_s \rightarrow \neg s$). In other words, positive (respectively negative) beliefs in nodes imply truth (respectively falsity) of the corresponding nodes.

Definition 10. *Given an an ADF $D = (S, L, C)$, $\Delta(D) := \{\mathbf{L}C_s \rightarrow s; \mathbf{L}\neg C_s \rightarrow \neg s \mid s \in S\}$*

It will prove useful to have a method to define an interpretation v_Q on the basis of a possible world structure $Q \subseteq \Omega(S)$ as follows: $v_Q := \sqcap_i Q$.

The critical reader might perhaps wonder if the translation does not require the "reversed" conditionals $\mathbf{L}s \rightarrow C_s$ and $\mathbf{L}\neg s \rightarrow \neg C_s$, which encode a form of *explanatory closure* of ADFs which states that for every node that is believed (respectively disbelieved), an agent should be able to give a reason for this belief (respectively disbelief). This is done by adding the premises $\mathbf{L}s \rightarrow C_s$ and $\mathbf{L}\neg s \rightarrow \neg C_S$. In fact, for any $s \in S$ and any AEE of $\Delta(D)$, Q will also imply both of the above implications:[6]

Fact 1. *Given an ADF $D = (S, L, C)$ and an autoepistemic extension Q of $\Delta(D)$, $Q, \omega \models (\mathbf{L}s \rightarrow C_s) \wedge (\mathbf{L}\neg s \rightarrow \neg C_S)$ for any $\omega \in Q$ and any $s \in S$.*

Proof. [7] Consider the ADF $D = (S, L, C)$ and suppose $Q \subseteq \Omega(S)$ is an AEE of $\Delta(D)$. Suppose now that $\omega \in Q$, $s \in S$ and $Q, \omega \models \mathbf{L}s$. Then $v_Q(s) = \top$ and since v_Q is complete (with Theorem 2) and thus also a model, $v_Q(C_s) = \top$ and thus $Q, \omega' \models \mathbf{L}C_s$ for any $\omega' \in \Omega(S)$. This implies that $\omega(C_s) = \top$ for any $\omega \in Q$.

[6] We thank an anonymous reviewer of a previous version of this paper for noticing this.

[7] Notice that the proof of this fact makes use of Theorem 2, which is shown later in this paper. However, since the proof of Theorem 2 does not in any way depend on this fact, this does not cause any logic circularity.

Altogether this shows that for any $s \in S$, $Q,\omega \models \mathbf{L}s \to C_s$ for any $s \in S$. The proof for $\mathbf{L}\neg s \to \neg C_s$ is analogous.

□

It is perhaps interesting to note, however, that an alternative translation $\Delta^*(D) = \{\mathbf{L}s \to C_s, \mathbf{L}\neg s \to \neg C_S \mid s \in S\}$ is *not* adequate, i.e. there might be AEEs that are not complete:

Example 5. *Let $D = (\{a\}, L, C)$ with $C_a = \top$. The interpretation $v(a) = \top$ is grounded and preferred. Since $\Delta^*(D) = \{\mathbf{L}a \to \top, \mathbf{L}\neg a \to \bot\}$, there are two AEEs of $\Delta^*(D)$: $\{a, \overline{a}\}$ and $\{a\}$. To see that $\{a, \overline{a}\}$ is an AEE, notice that (for any $\omega \in \Omega(\{a\})$) $\{a, \overline{a}\}, \omega \models \neg\mathbf{L}a \wedge \neg\mathbf{L}\neg a$ and thus $\Delta^*(D)$ is satisfied trivially.*

We are now ready to prove the main adequacy results. We first need an intermediate result whose proof is left out in view of spatial considerations:

Lemma 1. *If Q is an AEE of $\Delta(D)$ then $[v_Q]^2 = Q$.*

Theorem 2. *Given an ADF $D = (S, L, C)$, the following statements hold:*

1. *If $Q \subseteq \Omega(S)$ is a consistent autoepistemic extension of $\Delta(D)$ then v_Q is a complete interpretation of D;*
2. *If v is a complete interpretation of D then Q_v is an autoepistemic extension of $\Delta(D)$.*

Proof. Ad 1: Suppose that Q is a consistent autoepistemic extension of $\Delta(D)$. We show that for any $s \in S$, $\Gamma_D(v_Q)(s) = v_Q(s)$. We show the case for $v_Q(s) = u$, the other cases are similar and left out in view of space restrictions.

Suppose indeed that $v_Q(s) = u$, i.e. there are some $\omega, \omega' \in Q$ s.t. $\omega(s) = \top$ and $\omega'(s) = \bot$. Since $\omega \in Q$ and Q is an AEE of $\Delta(D)$ and $\mathbf{L}\neg C_s \to \neg s \in \Delta(D)$, $Q, \omega \models s \to \neg\mathbf{L}\neg C_s$. Likewise (since $\mathbf{L}C_s \to s \in \Delta(D)$), $Q, \omega' \models \neg s \to \neg\mathbf{L}C_s$. This implies that $Q, \omega \models \neg\mathbf{L}C_s$ and $Q, \omega' \models \neg\mathbf{L}\neg C_s$. This implies that there are some $\omega'', \omega''' \in Q$ s.t. $Q, \omega'' \models C_s$ and $Q, \omega''' \models \neg C_s$. Since $Q = [v_Q]^2$ by Lemma 1, this means $\Gamma_D(v_q)(s) = u$.

Thus we have established that $v_Q(s) = x$ implies $\Gamma_D(v_Q)(s) = x$ for every $x \in \{\top, \bot, u\}$. The cases for $\Gamma_D(v_Q)(s) = x$ follow with contraposition from this and since $\{\top, \bot, u\}$ exhausts all possible values of $\Gamma_D(v_Q)$.

The proof of 2. is left out in view of spatial considerations.

□

From this the following corollary follows for the complete semantics:

Corollary 1. *Given ADF $D = (S, L, C)$ and $s \in S$: $D \mathrel{|\!\!\sim}^{\cap}_{\mathsf{Cmp}} s[\neg s]$ iff $\Delta(D) \mathrel{|\!\!\sim}^{\cap}_{AEL} s[\neg s]$.*

We now turn to grounded and preferred semantics. We first need the following Lemma:

Lemma 2. 1. *Given some $v \in \mathcal{V}$, $v = v_{Q_v}$.*

2. *Given an ADF $D = (S, L, C)$, if $Q \subseteq \Omega(S)$ is an AEE of $\Delta(D)$, then also $Q = Q_{v_Q}$.*

Proof. We sketch the proof of 2., 1 is analogous but simpler. Suppose for this $D = (S,L,C)$ and $Q \subseteq \Omega(S)$ is an AEE of $\Delta(D)$. Clearly $Q_{v_Q} \supseteq Q$. Suppose now towards a contradiction there is an $\omega \in Q_{v_Q} \setminus Q$. For any $\omega \in Q_{v_Q}$, $\omega \models s[\neg s]$ iff $v_Q(s) = \top[\bot]$, i.e. $\omega \models s[\neg s]$ iff $Q,\omega' \models \mathbf{L}s[\mathbf{L}\neg s]$ for any $\omega' \in \Omega(S)$. Thus, for every $s \in S$ s.t. $\omega(s) \neq v_Q(s)$, $v_Q(s) = u$, i.e. $Q,\omega' \models \neg\mathbf{L}s \wedge \neg\mathbf{L}\neg s$ and thus there are some $\omega',\omega'' \in Q$ s.t. $\omega'(s) = \omega(s)$ and $\omega''(s) \neq \omega(s)$. Furthermore, since Q is an AEE of $\Delta(D)$ and $\mathbf{L}C_s \to s \in \Delta(D)$ and $\mathbf{L}\neg C_s \to \neg s \in \Delta(D)$, $Q,\omega' \models \neg\mathbf{L}C_s \wedge \neg\mathbf{L}\neg C_s$ for any $\omega' \in Q$.

But then $Q,\omega'' \not\models \mathbf{L}C_s \to s$, contradiction to Q being an AEE of $\Delta(D)$ and $\omega'' \in Q$). But then $Q,\omega \models (\mathbf{L}C_s \to s) \wedge (\mathbf{L}\neg C_s \to \neg s)$. Altogether, we have established that: if $v_Q(s) = u$ then $Q,\omega \models (\mathbf{L}C_s \to s) \wedge (\mathbf{L}\neg C_s \to \neg s)$. We can easily show the same for any $s \in S$ s.t. $v_Q(s) \in \{\top,\bot\}$, which implies $Q,\omega \models \Delta(D)$ and thus $\omega \in Q$, contradiction to the supposition.

□

We notice that Lemma 2 does not in general hold for sets of possible worlds. To see this, consider the set $Q = \{a\overline{b}, \overline{a}, b\}$. Then $v_Q(a) = v_Q(b) = u$ and $Q_{v_Q} = \{ab, a\overline{b}, \overline{a}b, \overline{a}\overline{b}\}$. The proofs of the following Theorems are straightforward in view of Theorem 2, Proposition 2 and Lemma 2 and left out in view of spatial restrictions.

Theorem 3. *Given an ADF $D = (S,L,C)$, the following statements hold:*

1. *If $Q \subseteq \Omega(S)$ is a minimally informative AEE of $\Delta(D)$ then v_Q is the grounded interpretation of D;*
2. *If v is the grounded interpretation of D then Q_v is a minimaly informative AEE of $\Delta(D)$.*

Theorem 4. *Given an ADF $D = (S,L,C)$, the following statements hold:*

1. *If $Q \subseteq \Omega(S)$ is a maximally informative AEE of $\Delta(D)$ then v_Q is a preferred interpretation of D;*
2. *If v is a preferred interpretation of D then Q_v is a maximally informative AEE of $\Delta(D)$.*

From these theorems the following corollary follows for the grounded and preferred semantics:

Corollary 2. *For any ADF $D = (S,L,C)$ and $s \in S$, the following statements hold:*

- $D \mathrel{|\!\!\sim}^{\cap}_{\mathsf{Prf}} s[\neg s]$ *iff* $\Delta(D) \mathrel{|\!\!\sim}^{\cap,\mathsf{max}}_{AEL} s[\neg s]$.
- $D \mathrel{|\!\!\sim}_{\mathsf{Grn}} s[\neg s]$ *iff* $\Delta(D) \mathrel{|\!\!\sim}^{\cap,\mathsf{min}}_{AEL} s[\neg s]$.

4.2. Properties of the Translation

In [9], several desirable properties for translations between non-monotonic formalisms where suggested: *faithfulness*, *polynomiality* and *modularity*. A faithful translation is a translation that preserves adequacy between the autoepistemic extensions and the semantics of ADFs. The faithfulness of our translation is shown in Theorem 2 for complete semantics, Theorem 3 for grounded semantics and Theorem 4 for preferred semantics.

Polynomiality is motivated by the requirement that the translation should be calculable within reasonable bounds. Clearly, the translation is polynomial: in fact it is linear in the number of nodes.

Modularity was originally defined for translations between circumscription and default logic [12]. Even though the original formulation was slightly different, we follow [24] in his formulation of modularity of a translation from ADFs to a target formalism. Basically, a translation is modular if "local" changes in the translated ADF will only lead to "local" changes in the translation. More formally, for two ADFs $D_1 = (S_1, L_1, C_1)$ and $D_2 = (S_2, L_2, C_2)$, such that $S_1 \cap S_2 = \emptyset$, a translation Δ is modular iff $\Delta(D_1 \cup D_2) = \Delta(D_1) \cup \Delta(D_2)$. It is easy to observe that the translation presented in this paper is modular.[8]

5. From autoepistemic logic to ADFs

The reader might wonder if it is possible to translate autoepistemic logic into ADFs. Such a translation is indeed possible, for the following reason: in [13] a translation from autoepistemic logic to *strong autoepistemic logic* was shown. In the same paper, it was also shown that strong autoepistemic logic can be translated into Reiter's default logic [23]. In [8] Reiter's default logic was translated into abstract argumentation, which can be captured in ADFs. It thus follows that ADFs admit autoepistemic logic under a composition of translations. A direct translation, however, remains to be investigated. We leave this as an avenue for further research.

6. Related Work

The main contribution of this paper is an embedding of ADFs in epistemic logics and a translation from ADFs into autoepistemic logic based on such an embedding. To the best of our knowledge, this is the first time that such an interpretation or translation is spelled out in the literature. There are, however, some related approaches that we wish to mention.

In [10] modal logic is applied to formalize fragments of formal argumentation theory. In particular, [10] establishes a correspondence between a given argumentation framework and a modal logic frame. The idea is that the argumentation framework and the modal frame will have the same number of nodes: for every argument there will be exactly one corresponding world. The meaning of the accessibility relation is, in a sense, inversed: if a attacks b then the world corresponding to b will be an accessible from the world corresponding to a. Consequently, even though both [10] and we interpret argumentation formalisms in some modal logic, the differences should be clear: we consider a translation into epistemic logic instead of a modal logic based on a frame structurally similar to

[8] [24] remarks that it would make sense from a conceptual point of view to generalize modularity to ADFs that have nodes that are not necessarily disjoint, but remarks that technically it is difficult to formulate such a generalized criterion of modularity. We follow [24] in leaving the formulation of such a criterion for future work.

the argumentation framework and we consider the more general ADFs instead of abstract argumentation frameworks.

The connections between ADFs and other formalisms for non-monotonic reasoning have been investigated before. [24] shows that there is a translation from ADFs into normal logic programs. In that paper, it is remarked that in view of the translation from ADFs into normal logic programs, and existing translations from normal logic programs into default logic and from default logic into autoepistemic logic (both by [7]), there exists a translation from ADFs into autoepistemic logic. We now give such a translation and argue for its conceptual adequacy.

Finally, we mention [15,11] where the correspondence between logics for non-monotonic conditionals are investigated. The results of that paper are that a subset of the complete models, namely the 2-valued models (interpretations $v \in \Omega(S)$ s.t. $v(s) = v(C_s)$ for every node s) can be straightforwardly modelled in conditional logics but for complete semantics, such a translation is less straightforward. The translations in this paper together with results on the relation between conditional logics and epistemic logic (e.g. [16]) can be used to shed further light on the correspondence between conditional logics and ADFs.

7. Conclusion and Outlook

In this paper, we have given an epistemic interpretation of ADFs and have formulated an intuitive, faithful, polynomial and modular translation from ADFs into autoepistemic logic. Not only is this interesting from a conceptual point of view, but this translation also is a starting point for further investigations into the connection between ADFs and other formalisms, since there are studies on the relationship between autoepistemic logic and other formalisms, such as default logic [9,7], logic programming [19,18] and circumscription [14]. Furthermore, the epistemic interpretation undertaken in this paper allows us to apply techniques developed in epistemic logic to ADFs. For example, *dynamic epistemic logic* [25] is a well-established field that uses epistemic logic to model changes in knowledge. The epistemic interpretation of ADFs in this paper can take advantage of developments in dynamic epistemic logic (such as [2,25]) to shed further light, among others, on argumentation dynamics (a topic that has been studied mainly for abstract argumentation frameworks until now) and argumentation in multi-agent interactions. In future work, we want to translate other semantics into autoepistemic logic, such as the different formulations of the stable semantics [5,4] and look at extensions of ADFs such as prioritized ADFs [4].

Acknowledgements The research reported in this paper was supported by the German National Science Foundation, DFG-project KE-1413/11-1.

References

[1] João Alcântara and Samy Sá. On three-valued acceptance conditions of abstract dialectical frameworks. *Electronic Notes in Theoretical Computer Science*, 344:3–23, 2019.

[2] Alexandru Baltag, Lawrence S Moss, and Sławomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Readings in Formal Epistemology*, pages 773–812. Springer, 2016.

[3] Bart Bogaerts. *Groundedness in logics with a fixpoint semantics.* PhD thesis, 2015.

[4] Gerhard Brewka, Hannes Strass, Stefan Ellmauthaler, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[5] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *KR 12*, 2010.

[6] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.

[7] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In *Logic-based artificial intelligence*, pages 127–144. Springer, 2000.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–358, 1995.

[9] Georg Gottlob. The power of beliefs or translating default logic into standard autoepistemic logic. In *Foundations of Knowledge Representation and Reasoning*, pages 133–144. Springer, 1994.

[10] Davide Grossi. On the logic of argumentation theory. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, pages 409–416. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[11] Jesse Heyninck, Gabriele Kern-Isberner, and Matthias Thimm. On the correspondence between abstract dialectical frameworks and non-monotonic conditional logics. In *33rd International FLAIRS Conference*, 2020.

[12] Tomasz Imielinski. Results on translating defaults to circumscription. *Artificial Intelligence*, 32(1):131–146, 1987.

[13] Tomi Janhunen. Representing autoepistemic introspection in terms of default rules. In *Proceedings of the 12th European Conference on Artificial Intelligence, ECAI'96*, pages 70–74. John Wiley and Sons, 1996.

[14] Tomi Janhunen. On the intertranslatability of autoepistemic, default and priority logics, and parallel circumscription. In *European Workshop on Logics in Artificial Intelligence*, pages 216–232. Springer, 1998.

[15] Gabriele Kern-Isberner and Matthias Thimm. Towards conditional logic semantics for abstract dialectical frameworks. In Carlos I. Chesnevar et al., editor, *Argumentation-based Proofs of Endearment*, volume 37 of *Tributes*. College Publications, November 2018.

[16] Costas D Koutras, Christos Moyzes, and Christos Rantsoudis. A reconstruction of default conditionals within epistemic logic. *Fundamenta Informaticae*, 166(2):167–197, 2019.

[17] Clarence Irving Lewis, Cooper Harold Langford, and P Lamprecht. *Symbolic logic.* Dover Publications New York, 1959.

[18] Vladimir Lifschitz and Grigori Schwarz. Extended logic programs as autoepistemic theories. In *LPNMR*, pages 101–114, 1993.

[19] V Wiktor Marek and Miroslaw Truszczynski. Reflexive autoepistemic logic and logic programming. *2nd Int. Ws. on LP & NMR*, pages 115–131, 1993.

[20] R Moore. Possible-world semantics for autoepistemic logic. In *Readings in nonmonotonic reasoning*, pages 137–142. Morgan Kaufmann Publishers Inc., 1987.

[21] Robert C Moore. Semantical considerations on nonmonotonic logic. *Artificial intelligence*, 25(1):75–94, 1985.

[22] Sylwia Polberg, Johannes Peter Wallner, and Stefan Woltran. Admissibility in the abstract dialectical framework. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 102–118. Springer, 2013.

[23] Raymond Reiter. A logic for default reasoning. *Artificial intelligence*, 13:81–132, 1980.

[24] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[25] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

[26] Zhiwei Zeng, Chunyan Miao, Cyril Leung, and Jing Jih Chin. Building more explainable artificial intelligence with argumentation. In *AAAI*, volume 33, pages 8044–8045, 2018.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200508

Learning Constraints for the Epistemic Graphs Approach to Argumentation

Anthony HUNTER
Department of Computer Science,
University College London, London, UK
(anthony.hunter@ucl.ac.uk)

Abstract. Epistemic graphs are a proposal for modelling how agents may have beliefs in arguments and how beliefs in some arguments may influence the beliefs in others. The beliefs in arguments are represented by probability distributions and influences between arguments are represented by logical constraints on these probability distributions. This allows for various kinds of influence to be represented including supporting, attacking, and mixed, and it allows for aggregation of influence to be captured, in a context-sensitive way. In this paper, we investigate methods for learning constraints, and thereby the nature of influences, from data. We evaluate our approach by showing that we can obtain constraints with reasonable quality from two publicly available studies.

Keywords. Probabilistic argumentation; Learning for argumentation; Non-normative argumentation.

1. Introduction

Argumentation often involves uncertainty. This can be uncertainty within an argument (e.g. uncertainty about the premises, or about the claim following the premises) or uncertainty between arguments (e.g. uncertainty about the nature of the support or attack by an argument on another). Further uncertainty arises when one agent is considering what arguments another agent believes (which can be important when the agent wants to persuade the other agent).

Following the results of an empirical study with participants [18], epistemic graphs have been introduced as a generalization of the epistemic approach to probabilistic argumentation [10,11]. In this approach, the graph is augmented with a set of epistemic constraints that can restrict the belief we have in an argument, and state how beliefs in arguments influence each other, with a varying degree of specificity. This is illustrated in Example 1.

Example 1. *Consider the graph in Figure 1, and let us assume that if* D *is strongly believed, and* B *or* C *is strongly disbelieved, then* A *is strongly believed, whereas if* D *is believed, and* B *or* C *is disbelieved, then* A *is believed. Furthermore, if* B *are* C *are believed, then* A *is disbelieved. These constraints could be reflected by the following formulae:* $\varphi_1 : p(\mathtt{D}) > 0.8 \wedge (p(\mathtt{B}) < 0.2 \vee p(\mathtt{C}) < 0.2) \rightarrow p(\mathtt{A}) > 0.8$*;*

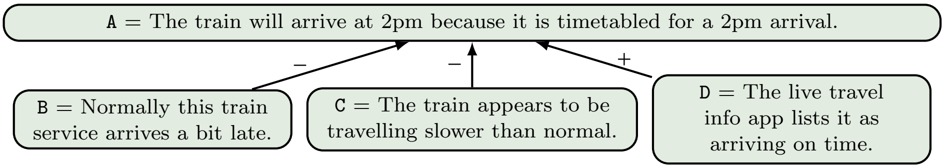

A = The train will arrive at 2pm because it is timetabled for a 2pm arrival.
−
−
+
B = Normally this train service arrives a bit late.
C = The train appears to be travelling slower than normal.
D = The live travel info app lists it as arriving on time.

Figure 1. Example of an epistemic graph. The + (resp. -) label denote support (resp. attack) relations. These are specified via the constraints given in Example 1.

$\varphi_2 : p(\mathtt{D}) > 0.5 \wedge (p(\mathtt{B}) \leq 0.5 \vee p(\mathtt{C}) \leq 0.5) \rightarrow p(\mathtt{A}) > 0.5$*; and* $\varphi_3 : (p(\mathtt{B}) > 0.5 \wedge p(\mathtt{C}) > 0.5) \rightarrow p(\mathtt{A}) < 0.5$.

Epistemic graphs can model both attack and support as well as relations that are neither positive nor negative. The flexibility of this approach allows us to both model the rationale behind existing dialectical semantics (such by Dung [4]) and to completely deviate from them when required. The fact that we can specify the conditions under which arguments should be evaluated, and that we can include constraints between unrelated arguments, permits the framework to be more context–sensitive. It also allows for better modelling of imperfect agents, which can be important in multi–agent applications. Epistemic graphs are therefore a flexible and potentially valuable tool for argumentation, and [10] has already provided methods for harnessing epistemic graphs in user modelling for persuasion dialogues where knowing about what the other agent beliefs can help in strategically choosing arguments to present (see [8] for more on computational persuasion).

To date, it has been assumed that the constraints for an epistemic graph are available somehow, though no methods for acquiring have so far been proposed. Yet a key potential advantage of taking a probabilistic approach is that we can learn epistemic graphs from data. As a first step to realizing this potential, in this paper, we investigate methods for learning constraints, and thereby the nature of influences, from data. Our approach is a form of association rule learning [1] where the rules that we learn are in the form of probabilistic constraints (i.e. constraints for epistemic graphs). To evaluate our approach, we focus on publicly available data obtained in two published surveys: on the use of Wikipedia in higher education in Spain [16]; and on political attitudes in Italy [17]. These studies collected views about specific statements which can be regarded as arguments. Using our methods, we show that we can obtain constraints with reasonable quality (in terms of support and confidence).

In the rest of the paper, we present the following: (Section 2) Review of the definitions for epistemic graphs; (Section 3) Framework for learning epistemic constraints; (Section 4) Evaluation of framework with two datasets; (Section 5) Comparison with the literature; and (Section 6) Conclusion and discussion.

2. Restricted Epistemic Graphs

This section presents a simpler version of epistemic graphs than presented in [11]. Essentially, epistemic graphs are labelled directed graphs equipped with a set of epistemic constraints for capturing the influences between arguments (as illustrated in Figure 1). Each node in the directed graph denotes an argument, and each arc denotes the influence of one argument on another. The label denotes the type of influence with options including positive (supporting), negative (attacking, and mixed. Both the labelled graph and the constraints provide information about the argumentation.

In this paper, we focus on the constraints rather than on the full power of the graphs. Let $\mathcal{G}$ denote a graph. Given the arguments in the graph, denoted $\mathsf{Nodes}(\mathcal{G})$, we consider a probability distribution $P : \wp(\mathsf{Nodes}(G)) \rightarrow [0,1]$ as being a probability assignment to each subset of the set of arguments such that this sums to 1 (i.e. $\sum_{X \subseteq \mathsf{Nodes}(G)} P(X) = 1$). The constraints restrict the set of probability distributions that satisfy the arguments (as we explain in the rest of this subsection).

Rather than consider any probability distribution in this paper, we will use finite probability distributions. For certain applications a restricted set of probability distributions can be used where the probability values come from a finite set of values [11]. This may be appropriate if we want to represent probability values as in a Likert scale [15]. It also has the benefit of always producing a finite set of distributions. However, for the approach to be coherent, this set should be closed under addition and subtraction (assuming the resulting value is in the $[0,1]$ interval) and should contain 1.

Definition 1. *A finite set of rational numbers from the unit interval* Π *is a* **restricted value set** *iff* $1 \in \Pi$ *and for any* $x, y \in \Pi$ *it holds that if* $x + y \leq 1$*, then* $x + y \in \Pi$*, and if* $x - y \geq 0$*, then* $x - y \in \Pi$.

Since we will only consider restricted value sets, we will refer to them as value sets. Examples include $\{0,1\}$, $\{0,0.5,1\}$, and $\{0,0.25,0.5,0.75,1\}$.

A probability distribution P for a value set Π is a probability distribution such that for each $\Gamma \subseteq \mathsf{Nodes}(G)$, $P(\Gamma) \in \Pi$. We will assume that all our probability distributions are with respect to a given value set. We denote the set of all belief distributions on $\mathsf{Nodes}(\mathcal{G})$ by $\mathsf{Dist}(\mathcal{G})$, and the set of restricted distributions for value set Π by $\mathsf{Dist}(\mathcal{G}, \Pi)$

Based on a given graph and restricted value set, we can now define the epistemic language. In this paper, we will only consider a sublanguage of that defined in [11].

Definition 2. *The* **restricted epistemic language** *based on graph* $\mathcal{G}$ *and a restricted value set* Π *is defined as follows: an* **epistemic atom** *is of the form* $P(\alpha)\#x$ *where* $\# \in \{<, \leq, =, \geq, >\}$*,* $x \in \Pi$ *and* $\alpha \in \mathsf{Nodes}(\mathcal{G})$*; an* **epistemic formula** *is a Boolean combination of epistemic atoms.*

Example 2. *Let* $\Pi = \{0, 0.5, 1\}$. *In the restricted epistemic language w.r.t.* Π*, we can only have atoms of the form* $p(\alpha)\#0$*,* $p(\alpha)\#0.5$*, and* $p(\alpha)\#1$*, where* $\alpha \in \mathsf{Nodes}(\mathcal{G})$ *and* $\# \in \{<, \leq, =, \geq, >\}$. *From these atoms we compose epistemic formulae, using the Boolean connectives, such as* $p(\alpha) \leq 0.5 \rightarrow \neg(p(\beta) \geq 0.5)$.

The semantics for constraints come from probability distributions $P \in \mathsf{Dist}(\mathcal{G}, \Pi)$, which assign probabilities to sets of arguments. Each $\Gamma \subseteq \mathsf{Nodes}(\mathcal{G})$ corresponds to a possible world where the arguments in Γ are true.

Definition 3. *The* **probability of an argument** *is defined as the sum of the probabilities of the worlds containing it:* $P(\alpha) = \sum_{\Gamma \subseteq \mathsf{Nodes}(\mathcal{G}) \ s.t. \ \alpha \in \Gamma} P(\Gamma)$.

We say that an agent believes an argument α to be acceptable to some degree if $P(\alpha) > 0.5$, disbelieves α to be acceptable to some degree if $P(\alpha) < 0.5$, and neither believes nor disbelieves α to be acceptable when $P(\alpha) = 0.5$. Using this, we can finally produce (restricted) satisfying distributions of an epistemic atom, and therefore of an epistemic formula:

Definition 4. *Let* Π *be a value set and let* $p(\alpha)\#v$ *be an epistemic atom where* $\# \in \{<, \leq, =, \geq, >\}$. *The* **satisfying distributions**, *or equivalently* **models**, *of* $p(\alpha)\#v$ *are defined as* $\mathsf{Sat}(p(\alpha)\#v) = \{P' \in \mathsf{Dist}(\mathcal{G}) \mid P'(\alpha)\#v\}$. *The* **restricted satisfying distribution** *of* $\psi = p(\alpha)\#v$ *w.r.t.* Π *are defined as* $\mathsf{Sat}(\psi, \Pi) = \mathsf{Sat}(\psi) \cap \mathsf{Dist}(\mathcal{G}, \Pi)$.

The set of satisfying distributions for a given epistemic formula is as follows where ϕ and ψ are epistemic formulae: $\mathsf{Sat}(\phi \wedge \psi) = \mathsf{Sat}(\phi) \cap \mathsf{Sat}(\psi)$; $\mathsf{Sat}(\phi \vee \psi) = \mathsf{Sat}(\phi) \cup \mathsf{Sat}(\psi)$; and $\mathsf{Sat}(\neg\phi) = \mathsf{Sat}(\top) \setminus \mathsf{Sat}(\phi)$. For a set of epistemic formulae $\Phi = \{\phi_1, \ldots, \phi_n\}$, the set of satisfying distributions is $\mathsf{Sat}(\Phi) = \mathsf{Sat}(\phi_1) \cap \ldots \cap \mathsf{Sat}(\phi_n)$. The same holds when restricting probabilities to a value set Π.

Example 3. *Consider the formula* $p(\mathtt{A}) > 0.5 \rightarrow \neg(p(\mathtt{B}) > 0.5)$ *with* $\Pi = \{0, 0.5, 1\}$. *Examples of probability distributions that satisfy the formula include* P_1 *s.t.* $P_1(\emptyset) = 1$, P_2 *s.t.* $P_2(\emptyset) = P_2(\{\mathtt{A}\}) = 0.5$, P_3 *s.t.* $P_3(\{\mathtt{A}\}) = 1$, *or* P_4 *s.t.* $P_4(\{\mathtt{A}\}) = P_3(\{\mathtt{A}, \mathtt{B}\}) = 0.5$ *(omitted sets are assigned* 0*). The probability distribution* P_5 *s.t.* $P_5(\{\mathtt{A}, \mathtt{B}\}) = 1$ *does not satisfy the formula.*

The restricted epistemic language does not incorporate features of the full epistemic language (as presented in [11]) such as terms that are Boolean combinations of arguments (e.g. $P(\mathtt{B} \vee \mathtt{C}) > 0.6$ which says that the probability argument B or argument C is greater than 0.6) or summation of probability values (such as $P(\mathtt{A}) + P(\mathtt{B}) \leq 1$ which says that the sum of probability A and probability B is less than or equal to 1). Nonetheless, the restricted epistemic language is a useful sublanguage as a starting point for learning constraints. We focus on this sublanguage in this paper as it simplifies the presentation and evaluation.

3. Learning Framework

We now present a general framework for generating a class of epistemic constraints from data as follows: we define the format for the data, the format for constraints that we will learn, and an algorithm for learning these constraints. To illustrate, we will use examples taken from two studies (that we will discuss further in Section 4) concerning use of Wikipedia in higher education in Spain [16], and political attitudes in Italy [17].

3.1. Input for Learning

We assume that each item of data is a function that gives a value on an 11 point scale to each attribute. We use a data item to represent the responses that a

	Pu3	Qu1	Qu3	Enj1
903	0.3	0.5	0.3	0.7
904	0.9	0.9	0.9	0.9
905	0.7	0.7	0.5	0.9
908	0.3	0.5	0.7	0.3
909	0.5	0.5	0.7	0.7

Table 1. Some rows and columns of data from the Spanish study (after mapping Likert values to our 11 point scale) where Pu3 denotes the argument that "Wikipedia is useful for teaching", Qu1 denotes the argument that "Articles in Wikipedia are reliable", Qu3 denotes the argument that "Articles in Wikipedia are comprehensive", and Enj1 denotes the argument that "Articles in Wikipedia stimulate curiosity".

participant gives to each question where the attribute denotes the statement (i.e. the argument), and the assignment is their answer (as illustrated in Table 1).

Definition 5. *A* **data item** *is a function d from a set of attributes to a set of values. A* **dataset***, $D = \{d_1, \ldots, d_n\}$, is a set of data items over attributes (i.e. arguments) $A = \{a_1, \ldots, a_m\}$. So for $d \in D$, and for $\alpha \in A$, $d(\alpha) \in \{0, 0.1, 0.2, , \ldots, 0.9, 1\}$.*

The Spanish and Italian studies used Likert scales (7, 8 and 10 point scales) for recording participants responses to arguments. For a common format, we map each value in the Likert scale to our 11 point scale (e.g. for the 7 point scale, we use the mapping $1 \mapsto 0$, $2 \mapsto 0.2$, $3 \mapsto 0.3$, $4 \mapsto 0.5$, $5 \mapsto 0.7$, $6 \mapsto 0.8$, and $7 \mapsto 1$). So we use each answer in the Likert scale as a proxy for the participant's belief in the argument. The 11-point scale allows us to represent total disbelief (i.e. 0), total belief (i.e. 1), and the values in between obtainable with 0.1 graduations.

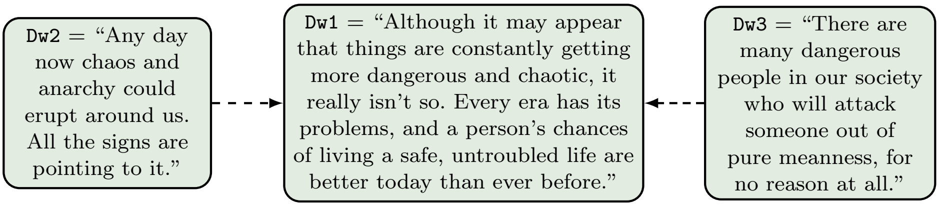
Example 4. *Consider Table 1. From row 903, we get $d_{903}(\texttt{Pu3}) = 0.3$, $d_{903}(\texttt{Qu1}) = 0.5$, $d_{903}(\texttt{Qu3}) = 0.3$, and $d_{903}(\texttt{Enj1}) = 0.7$.*

Given the set of arguments in the data, we then identify relationships between them. For a pair of arguments α and β, we say that α **influences** β if a change in the belief in α will potentially result in the change in the belief in β. For instance, an argument influences another argument if it appears to attack it (i.e. it could be regarded as a counterargument), or if it appears to support it. But relationships may be more subtle or mixed (see [11] for more details).

Definition 6. *An* **influence tuple** *is a tuple $(\{\alpha_1, \ldots, \alpha_n\}, \beta)$, where $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathsf{Nodes}(\mathcal{G}) \setminus \{\beta\}$ and $\beta \in \mathsf{Nodes}(\mathcal{G})$ and each α_i influences β. We refer to each α_i as an* **influencer** *and β as an* **influence target***.*

In this paper, we identified influence tuples by hand (i.e. by reading the statements in order to judge which arguments might be influenced by each argument). Potential alternatives to doing this by hand include automated reasoning with background knowledge about the arguments (such as causal relationships), and natural language processing to find logical relationships such as attack.

Example 5. *Consider the arguments* Dw1 *to* Dw3 *in Figure 2. By inspection we may regard* Dw2 *and* Dw3 *as attackers of* Dw1*, and so treat* Dw2 *and* Dw3 *as influencers of* dw1*. Hence, the influence tuple is* $(\{\texttt{Dw2}, \texttt{Dw3}\}, \texttt{Dw1})$.



Dw2 = "Any day now chaos and anarchy could erupt around us. All the signs are pointing to it."

Dw1 = "Although it may appear that things are constantly getting more dangerous and chaotic, it really isn't so. Every era has its problems, and a person's chances of living a safe, untroubled life are better today than ever before."

Dw3 = "There are many dangerous people in our society who will attack someone out of pure meanness, for no reason at all."

Figure 2. Arguments from the Italian study considered in Example 5. The dashed arcs denote influences.

So the input to the induction process is a data tuple and a set of influence tuples which provides extra information to guide the learning process. The learning process will ascertain (for the population of the study) whether there is indeed a relationship between some/all of the influencers and the influence target and if so, what the nature of that influence is. For instance, it could be that one argument does indeed contradict another argument, but for the population of a study, most people believe the attacker and the attackee. In this way, we want constraints that represent the beliefs of the population of the study rather than represent some normative interpretation of the arguments.

3.2. Output from Learning

The aim of learning is to take the input (a data set and a set of influence tuples) and return a set of constraints where each constraint is a rule. This set of rules will be a subset of the candidate rules defined next. Obviously each candidate rule is an epistemic formula (according to Definition 2).

Definition 7. *Let $I = (\{\beta_1, \ldots, \beta_n\}, \alpha)$ be an influence tuple, and Π be a value set. The set of* **candidate rules** *for I and Π is*

$$\begin{aligned}\mathsf{Rules}(I,\Pi) = \{&p(\gamma_1)\#_1 v_1 \wedge \ldots \wedge p(\gamma_k)\#_k v_k \rightarrow p(\alpha)\#_{k+1} v_{k+1} \mid \\ &\{\gamma_1,\ldots,\gamma_k\} \subseteq \{\beta_1,\ldots,\beta_n\} \text{ and } \#_i \in \{\leq, >\} \text{ and } v_i \in \Pi \setminus \{0,1\}\}\end{aligned}$$

Example 6. *Let $I = (\{\mathtt{Qu1}\}, \mathtt{Enj1})$ be an influence tuple and let $\Pi = \{0, 0.5, 1\}$. From this, the set of candidate rules $\mathsf{Rules}(I,\Pi)$ is*

$$\begin{array}{ll} p(\mathtt{Qu1}) > 0.5 \rightarrow p(\mathtt{Enj1}) > 0.5 & p(\mathtt{Qu1}) > 0.5 \rightarrow p(\mathtt{Enj1}) \leq 0.5 \\ P(\mathtt{Qu1}) \leq 0.5 \rightarrow p(\mathtt{Enj1}) > 0.5 & p(\mathtt{Qu1}) \leq 0.5 \rightarrow p(\mathtt{Enj1}) \leq 0.5 \end{array}$$

So for each influence tuple, the output of the induction process will be a set of rules, and these will be selected from the candidates in $\mathsf{Rules}(I,\Pi)$.

3.3. Generate Rules from Data

In the following, we introduce the 2-way generalization step that generates a rule from a data item. It has a precondition (above the line) and a postcondition (below the line). In the postcondition, the epistemic atoms in the rule are either of the form greater than 0.5 or less than or equal to 0.5 (i.e. two possible intervals). This gives us the most general kind of rule that we can obtain, and provides a baseline for comparison with frameworks for generating a wider variety of rules.

Definition 8. *Let d be a data item and $(\{\alpha_1,\ldots,\alpha_n\},\beta)$ be an influence tuple. The* **2-way generalization step** *is the following where for each i, if $v_i > 0.5$, then $\#_i$ is "$>$", else if $v_i \leq 0.5$, then $\#_i$ is "$\leq$".*

$$\frac{d(\alpha_1) = v_1,\ldots,d(\alpha_n) = v_n, d(\beta) = v_{n+1}}{p(\alpha_1)\#_1 0.5 \wedge \ldots \wedge p(\alpha_n)\#_n 0.5 \rightarrow p(\beta)\#_{n+1} 0.5}$$

For a data item d that satisfies the precondition, then $\mathsf{TwoWayGen}(d,I,\Pi)$ *returns the rule given in the postcondition, otherwise it returns nothing.*

Example 7. *The following is the result of applying the 2-way generalization rule to the data in row 908 in Table 1.*

$$p(\texttt{Qu1}) \leq 0.5 \wedge p(\texttt{Qu3}) > 0.5 \wedge p(\texttt{Enj1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$$

Definition 9. *Let D be a dataset, I be an influence tuple, and Π be a set of values. The* **generalize** *function, denoted* $\mathsf{Generalize}(D,I,\Pi)$*, returns the set* $\{\mathsf{TwoWayGen}(d,I,\Pi) \mid d \in D\}$.

Whilst we have focused on a 2-way generalization step, which results in a specific kind of rule, there are various ways we could expand the variety of rules that we could generate from the data. For instance, from the data item d where $d(\texttt{A}) = 0.2$, $d(\texttt{B}) = 0.6$, and $d(\texttt{C}) = 0.9$, we might want to obtain the generalization $p(\texttt{A}) \leq 0.2 \wedge p(\texttt{B}) \geq 0.6 \rightarrow p(\texttt{C}) \geq 0.9$ which involves representation of tighter intervals on belief (less than or equal to 0.2 instead of less than or equal to 0.5, and greater than or equal to 0.6 or 0.9 instead of greater than 0.5).

3.4. Identify the Best and Simplest rules

So from a dataset, an influence tuple, and a value set, we obtain a set of rules. At this stage, these are just candidates, and there is no guarantee that they are good with respect to the data.

Definition 10. *Let Π be a value set. For an atom of the form $p(\beta)\#v$, let* $\mathsf{Values}(p(\beta)\#v,\Pi) = \{x \in \Pi \mid x\#v\}$,

Example 8. *For the rule in Example 7,* $\mathsf{Values}(p(\texttt{Pu3}) \leq 0.5,\Pi) = \{0, 0.5\}$*, where* $\Pi = \{0, 0.5, 1\}$.

In order to harness measures from association rule learning, which we present in Table 2, we require the following subsidiary definitions below. Informally, a rule is fired by a data item when the conditions of the rule are satisfied by the data item. Furthermore, a rule agrees with a data item when the consequent is satisfied by the data item. Finally, a rule is correct with respect to a data item when the rule being fired implies the consequent is satisfied by the data item.

Definition 11. *Let $d \in D$ be a data item, and let $R = \phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi_{n+1}$ be a rule, and for $i \in \{1,\ldots,n+1\}$, let ϕ_i be of the form $P(\alpha_i)\#_i v_i$. We say R is* **fired** *by d iff for each ϕ_i s.t. $i \leq n$, $d(\alpha_i) \in \mathsf{Values}(P(\alpha_i)\#_i v_i,\Pi)$; R* **agrees** *with d iff $d(\alpha_{i+1}) \in \mathsf{Values}(\phi_{n+1},\Pi)$; and R is* **correct** *w.r.t. d iff if R is fired by d, then R agrees with d.*

Measure	Definition
$\mathsf{Support}(R, D)$	$\frac{1}{\lvert D\rvert} \times \lvert\{d \in D \mid R \text{ is fired by } d\}\rvert$
$\mathsf{Confidence}(R, D)$	$\frac{1}{\lvert D\rvert} \times \lvert\{d \in D \mid R \text{ is correct w.r.t. } d\}\rvert$
$\mathsf{Lift}(R, D)$	$\frac{\lvert\{d \in D \mid R \text{ is correct w.r.t. } d\}\rvert}{\lvert\{d \in D \mid R \text{ is fired by } d\}\rvert \times \lvert\{d \in D \mid R \text{ agrees with } d\}\rvert}$

Table 2. Measures for support, accuracy and lift where R is a rule, and D is a dataset.

```
$\mathsf{Generate}(D, I, \tau_{\mathsf{support}}, \tau_{\mathsf{accuracy}})$
    $AllRules = \mathsf{Generalize}(D, I, \Pi)$
    $BestRules = \mathsf{Best}(AllRules, D, \tau_{\mathsf{support}}, \tau_{\mathsf{accuracy}})$
    $return\ \mathsf{Simplest}(BestRules)$
```

Figure 3. The **generate algorithm** where D is a dataset, I is a set of influence tuples, Π is a value set, $\tau_{\mathsf{support}} \in [0, 1]$ (resp. $\tau_{\mathsf{confidence}} \in [0, 1]$) is a threshold for support (resp. confidence).

Example 9. *Consider the rule* $P(\mathtt{Pu3}) \leq 0.5 \Rightarrow P(\mathtt{Enj1}) > 0.5$ *with data from Table 1. The rule is fired with 903, 908, and 909, and is correct with 903 and 909.*

Given a set of rules and a dataset, the best rules are those that exceed the thresholds for support and confidence and have lift greater than 1.

Definition 12. *For a set of rules Rules, and a dataset D, with a threshold for support* $\tau_{\mathsf{support}} \in [0, 1]$*, and a threshold for confidence* $\tau_{\mathsf{confidence}} \in [0, 1]$*, the set of* **best rules***, denoted* $\mathsf{Best}(Rules, D, \tau_{\mathsf{support}}, \tau_{\mathsf{confidence}})$*, is* $\{R \in Rules \mid \mathsf{Support}(R, D) > \tau_{\mathsf{support}}$ *and* $\mathsf{Confidence}(R, D) > \tau_{\mathsf{confidence}}$ *and* $\mathsf{Lift}(R, D) > 1\}$*.*

For a set of rules with a particular head, the simplest are those with a minimal set (w.r.t. set inclusion) of conditions. The following Simplest function collects the simplest rules for each head in a set of rules.

Definition 13. *For a rule* $R = \phi_1 \wedge \ldots \wedge \phi_n \rightarrow \psi$*, let* $\mathsf{Conditions}(R) = \{\phi_1, \ldots, \phi_n\}$ *and* $\mathsf{Head}(R) = \psi$*. For a set of rules Rules, the* **simplest rules***, denoted* $\mathsf{Simplest}(Rules)$*, is the set of rules* $\{R \in Rules \mid$ *for all* $R' \in Rules$*, if* $\mathsf{Head}(R) = \mathsf{Head}(R')$*, then* $\mathsf{Conditions}(R) \subseteq \mathsf{Conditions}(R')\}$*.*

The algorithm for generating the rules is given in Figure 3, and we evaluate a Python implementation[1] of the algorithm in the next section.

4. Evaluation

In this paper, we consider data from two published studies. The data from each study contains the answers from asking individuals a number of questions including their level of agreement with certain statements (as illustrated in Table 1).

[1] Code available at http://www0.cs.ucl.ac.uk/staff/A.Hunter/papers/epilearn.zip

So each row in the data concerns an individual. Each statement can be regarded as an argument. The studies are: (1) the appropriateness of Wikipedia in a Spanish higher education institute [16] which was obtained from 901 individuals and involved 26 statements; and (2) views on political issues in Italy [17] which was obtained from 774 individuals and involved 75 statements.

Example 10. *Some of the statement from the Spanish dataset, are* Pu3 = *"Wikipedia is useful for teaching",* Qu1 = *"Articles in Wikipedia are reliable",* Qu3 = *"Articles in Wikipedia are comprehensive",* Enj1 = *"Articles in Wikipedia stimulate curiosity",* Use2 = *"I use Wikipedia as a platform to develop educational activities with students",* Use3 = *"I recommend my students to use Wikipedia",* Bi1 = *"In the future, I will recommend the use of Wikipedia to my colleagues and students", and* Bi2 = *"In the future, I will use Wikipedia in my teaching activities".*

Example 11. *Some of the statements from the Italian dataset, are* Sys2 = *"In general, the political system works as it should",* Sys3 = *"The Italian society must be radically changed",* Sys7 = *"Our society gets worse year by year",* Sys8 = *"Our society is organized so that people generally get what they deserve",* Dw6 = *"Every day as society become more lawless and bestial, a person's chances of being robbed, assaulted, and even murdered go up and up", and* Dw8 = *"It seems that every year there are fewer and fewer truly respectable people, and more and more persons with no morals at all who threaten everyone else".*

For each dataset, we constructed a set of influence tuples by hand based on the text descriptions given for each argument (i.e. the statement) considered in the study. Then, for each influence tuple, we used our algorithm to generate the constraints using the training data (which was a randomly selected subset of 80% of the dataset), and to avoid over-fitting, a maximum of 4 conditions per rule. We evaluated the rules for support, confidence, and lift (as defined in Table 2) using the remaining 20% of the data. Some rules learned from the Spanish (respectively Italian) dataset are given in Example 12 (respectively Example 13).

Example 12. *The following are some of the rules generated from the Spanish dataset, with influence tuple* $(\{\texttt{Qu1}, \texttt{Qu3}, \texttt{ENJ1}, \texttt{JR1}, \texttt{JR2}, \texttt{SA1}\}, \texttt{Pu3})$

1. $p(\texttt{Qu3}) > 0.5 \land p(\texttt{Qu1}) > 0.5 \rightarrow p(\texttt{Pu3}) > 0.5$
2. $p(\texttt{Enj1}) \leq 0.5 \land p(\texttt{Qu1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$
3. $p(\texttt{Jr2}) \leq 0.5 \land p(\texttt{Enj1}) \leq 0.5 \rightarrow p(\texttt{Pu3}) \leq 0.5$

Example 13. *The following are some of the rules generated from the Italian dataset, with the influence tuples* $(\{\texttt{Sys1}, \texttt{Sys3}, \texttt{Sys4}, \texttt{Sys5}, \texttt{Sys6}, \texttt{Sys7}, \texttt{Sys8}\}, \texttt{Sys2})$ *and* $(\{\texttt{Sys1}, \texttt{Sys2}, \texttt{Sys4}, \texttt{Sys5}, \texttt{Sys6}, \texttt{Sys7}, \texttt{Sys8}\}, \texttt{Sys3})$.

1. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys2}) \leq 0.5$
2. $p(\texttt{Sys8}) > 0.5 \rightarrow p(\texttt{Sys2}) \leq 0.5$
3. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys3}) > 0.5$

For each constraint generated by our algorithm, we tested it using the testing data (i.e. the subset of the dataset after subtracting the training data). We ran

Study	Influence target	No. of influencers	No. of of rules	Condi- tions	Support	Confi- dence	Lift	Time (sec)
Spain	Use2	19	11.3	1.0	0.68	0.95	1.04	192.34
Spain	Use3	19	14.0	1.69	0.60	0.84	1.16	178.36
Spain	Bi1	17	15.8	1.84	0.54	0.82	1.15	148.02
Spain	Bi2	17	12.7	2.1	0.51	0.80	1.20	140.98
Spain	Qu1	13	3.3	2.07	0.51	0.84	1.37	56.55
Spain	Qu3	13	4.2	1.68	0.58	0.88	1.17	48.66
Italy	Dw1	9	3.1	2.45	0.43	0.80	1.22	14.33
Italy	Dw3	9	4.0	1.0	0.75	0.84	1.15	15.39
Italy	Dw6	9	5.0	1.02	0.69	0.88	1.11	17.95
Italy	Dw8	9	4.2	1.7	0.67	0.83	1.22	16.65
Italy	Sys2	7	7.0	1.0	0.76	0.96	1.03	7.89
Italy	Sys3	7	1.6	1.48	0.52	0.82	1.22	8.21

Table 3. Results for the Spanish and Italian datasets with 10 repetitions. Column 3 is the number of influencers in the influence tuple. Column 5 is the average number of conditions per rule. For columns 4 to 9, the value is the average of the repetitions with $\tau_{\mathsf{confidence}} = 0.8$ and $\tau_{\mathsf{support}} = 0.4$.

the Python implentation in an evaluation on a Windows 10 HP Pavilion Laptop (with AMD A10 2GHz processor and 8GB RAM). In Table 3, we give results for some influence tuples with the Spanish and Italian datasets.

These results show that we are able to obtain reasonable quality constraints (in terms of support, confidence, and lift) from data by our simple version of association learning. Furthermore, the number of rules selected, and the complexity of those rules, tend to be appropriate for the application (i.e. enough rules to give insights into the data but still reasonably concise). Also, the number of rules and the average measure of support per rule are quite high (for example, for Use2 in Table 3, it is 11.3 and 0.68 respectively) which means that the data shows that there is indeed a number of ways that the target is influenced by other arguments, and each of those ways occurs frequently. Note, we have set a quite high threshold for support, and by lowering this, we can raise lift above 2.

The time performance is also reasonable. For the Spanish dataset, we consider sets of influencers of cardinality of up to 19 arguments. This means that a large number of rules can be constructed for each subset of influencers (e.g. over 70K for 19 influencers with a maximum of 4 conditions per rule). Yet the number of rules returned by the algorithm is often in the range of 10 to 20 rules, and the algorithm is running in less than 200 seconds. Furthermore, it is reasonable to expect that for many domains we should be able to restrict the number of influencers for each argument to less than 20 (and compare this with most argument graphs where far fewer attackers per argument are represented).

5. Comparison with the Literature

Two important approaches to probabilistic (abstract) argumentation are the constellations and the epistemic approaches [7]. In the constellations approach, there is uncertainty about which arguments and attacks should appear in the argument graph [5,14]. In contrast, in the epistemic approach, the topology of the argument graph is fixed, but there is uncertainty about whether an argument is believed [20,7,2,6,12]. The approach of epistemic graphs is a generalization of the epistemic approach.

For some time, there has been interest in using argumentation for improving machine learning and using machine learning for generating arguments (for a review, see [3]). In the literature, there are three recent proposals for learning for argumentation that are based on probabilistic techniques, though they are different to our proposal. The first proposal uses the usual labels for arguments *in*, *out* and *undecided*, augmented with *off* for denoting that the argument does not occur in the graph [19]. A probability distribution over labellings gives a form of probabilistic argumentation. For learning, the probability distribution is used to generate labellings that are used as data, and then the argument graph that best describes this data is identified. The second proposal takes as input a profile $\langle X_1, \ldots, X_n \rangle$ where each X_i is a set of acceptable arguments, and by using Bayes theorem, the output is a posterior probability for a set of arguments being an extension. This is calculated using a Bayesian network that incorporates assumptions about the relationships between choice of semantics and choice of attacks, and how these influence extensions [13]. The third proposal generates the probability distribution over subgraphs as used in the constellations approach [9]. It takes as input a profile $[(\phi_1, v_1), \ldots, (\phi_n, v_n)]$ where each ϕ_i is a Boolean combination of arguments that specifies an opinion on the topology of the argument graph, and v_i is the belief in that opinion, and returns the probability distribution that best represents the opinions. These three proposals concern uncertainty about the structure of the graph. Clearly none involve the epistemic approach to probabilistic argumentation, and in particular, none consider how constraints for epistemic graphs could be obtained from data.

6. Discussion

In this paper, we have proposed a framework for learning a class of constraints for epistemic graphs and evaluated it with two datasets. Generating epistemic graphs for argumentation offers a valuable way of constructing a representation of how arguments interact. A significant barrier to the deployment of argumentation formalisms has been the challenge of how to construct the required representations. Taking a probabilistic approach allows us to overcome this hurdle and thereby scale up the kind of problem we can tackle with an argumentation solution.

From the point of view of association rule learning [1], we have only presented a very simple framework to show that it is viable to generate constraints for epistemic graphs in this way. In future work, we will introduce alternatives to the 2-way generalization step so that we can learn a wider variety of rules from the restricted epistemic language presented using tight constraints and a wider variety of values as discussed at the end of Section 3.3, and more complex rules such as with heads that provide both upper and lower bounds on belief (e.g. $p(\texttt{Sys7}) > 0.5 \rightarrow p(\texttt{Sys3}) > 0.5 \wedge p(\texttt{Sys3}) \leq 0.7$). We will also consider a less restricted version of the language of epistemic graphs (i.e. use the full language as defined in [11]), and we will consider how we can learn labels for the epistemic graphs.

References

[1] R. Agrawal, T. Imieliski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of SIGMOD '93*, pages 207–216. ACM Press, 1993.

[2] P. Baroni, M. Giacomin, and P. Vicig. On rationality conditions for epistemic probabilities in abstract argumentation. In *Proceedings of COMMA'14*, 2014.

[3] O. Cocarascu and F. Toni. Argumentation for machine learning: A survey. In *Proceedings of COMMA'16*, pages 219–230. IOS Press, 2016.

[4] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[5] P. M. Dung and P. M. Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proceedings of COMMA'10*, pages 171–182, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[6] D. Gabbay and O. Rodrigues. Probabilistic argumentation: an equational approach. *Logica Universalis*, 9(3):345–382, 2015.

[7] A. Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013.

[8] A. Hunter, L. Chalaguine, T. Czernuszenko, E. Hadoux, and S. Polberg. Towards computational persuasion via natural language argumentation dialogues. In *Proceedings of KI'19*, volume 11793 of *LNCS*, pages 18–33. Springer, 2019.

[9] A. Hunter and K. Noor. Aggregation of perspectives using the constellations approach to probabilistic argumentation. In *Proceedings of AAAI'20*, pages 2846–2853. AAAI Press, 2020.

[10] A. Hunter, S. Polberg, and N. Potyka. Updating Belief in Arguments in Epistemic Graphs. In *Proceedings of KR'18*, pages 138–147. AAAI Press, 2018.

[11] A. Hunter, S. Polberg, and M. Thimm. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. *Artificial Intelligence*, 281:103236, 2020.

[12] A Hunter and M Thimm. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research*, 59:565–611, 2017.

[13] H. Kido and K. Okamoto. A Bayesian approach to argument-based reasoning for attack estimation. In *Proceedings of IJCAI'17*, pages 249–255. IJCAI, 2017.

[14] H. Li, N. Oren, and T. Norman. Probabilistic argumentation frameworks. In *Proceedings of TAFA'11*, volume 7132 of *LNCS*, pages 1–16. Springer, 2012.

[15] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1931.

[16] A. Meseguer-Artola, E. Aibar, J. Llads, J. Minguilln, and M. Lerga. Factors that influence the teaching use of Wikipedia in higher education. *Journal of the Association for Information Science and Technology*, 67(5):1224–1232, 2016.

[17] V. Pellegrini, L. Leone, and M. Giacomantonio. Dataset about populist attitudes, social world views, socio-political dispositions, conspiracy beliefs, and anti-immigration attitudes in an italian sample. *Data in Brief*, 25:104144, 2019.

[18] S. Polberg and A. Hunter. Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *International Journal of Approximate Reasoning*, 93:487 – 543, 2018.

[19] R. Riveret and G. Governatori. On learning attacks in probabilistic abstract argumentation. In *Proceedings of AAMAS'16*, pages 653–661, 2016.

[20] M. Thimm. A probabilistic semantics for abstract argumentation. In *Proceedings of ECAI'12*. IOS Press, 2012.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200509

Revisiting SAT Techniques for Abstract Argumentation

Jonas KLEIN and Matthias THIMM
University of Koblenz-Landau, Germany

Abstract. We present MINIAF, a general SAT-based abstract argumentation solver that can be used with any SAT solver. We use this general solver to evaluate 12 different SAT solvers wrt. their capability of handling abstract argumentation problems. While our results show that the runtime performance of different SAT solvers are generally comparable, we also observe some statistically significant differences.

Keywords. abstract argumentation, algorithms, satisfiability

1. Introduction

Approaches to formal argumentation [2] encompass non-monotonic reasoning techniques that focus on the interplay between arguments. One of the most influential models in this area is that of abstract argumentation [18] which represents argumentation scenarios as directed graphs, where arguments are identified with vertices and an “attack” between one argument and another is modelled via a directed edge. In order to reason with abstract argumentation frameworks one considers *extensions*, i. e., sets of arguments that are mutually acceptable, given some formal account to “acceptability” [6]. Many of the reasoning problems have been shown to be intractable in general [19] and there has been an increased effort in recent years to develop algorithms and systems to solve problems of practically relevant sizes [32,23]. One of the predominant paradigms for algorithms in this context, is to reduce the reasoning problem to one or more calls to a *satisfiability* (SAT) solver [9]. Systems following this paradigm are, e. g., ArgSemSAT [14,15], pyglaf [1], μ-toksia [29], argmat-sat [30], and many more. The actual systems differ in some subtleties how the reasoning problem is encoded in a SAT problem, strategies for iterative calls to SAT solvers, and, in particular, the employed SAT solver. For example, ArgSemSAT uses MiniSAT[1] [20] while pyglaf and μ-toksia use Glucose [3], and argmat-sat uses CryptoMiniSat5[2].

In this paper, we revisit SAT-based techniques for reasoning with abstract argumentation and, in particular, ask the question *if* and *how* the choice of a concrete SAT solver may influence the performance of the overall argumentation system. In order to address this question independently of any existing SAT-based argumentation solver (that may be tailored towards the use of a concrete SAT-solver), as a first contribution we present MINIAF, a minimal implementation of a reasoning engine making use of SAT-solving

[1] Although [15] also reports on an experimental comparison with using Glucose.
[2] `https://github.com/msoos/cryptominisat`

techniques. This solver can be parametrised by any SAT solver following the command line interface of the SAT competition[3]. As a second contribution, we perform an extensive experimental analysis of running MINIAF with 12 different SAT solvers in order to compare the SAT solvers performance on the ICCMA17 [23] benchmark set. Our findings are that most SAT solvers exhibit a similar performance, although certain deviations can be observed. In summary, the contributions of this paper are as follows.

1. We present MINIAF, a minimal and flexible SAT-based argumentation solver (Section 3).
2. We perform an extensive experimental evaluation parametrising MINIAF with 12 different SAT solvers (Section 4).

We discuss relevant preliminaries in Section 2 and conclude in Section 5.

2. Preliminaries

An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ where A is a set of arguments and R is a relation $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$. For two arguments $a, b \in \mathsf{A}$ the relation $a\mathsf{R}b$ means that argument a attacks argument b. For $a \in \mathsf{A}$ define $a^- = \{b \mid b\mathsf{R}a\}$ and $a^+ = \{b \mid a\mathsf{R}b\}$. We say that a set $S \subseteq \mathsf{A}$ *defends* an argument $b \in \mathsf{A}$ if for all a with $a\mathsf{R}b$ then there is $c \in S$ with $c\mathsf{R}a$.

Semantics are given to abstract argumentation frameworks by means of extensions [18]. An extension E is a set of arguments $E \subseteq \mathsf{A}$ that is intended to represent a coherent point of view on the argumentation modelled by AF. Arguably, the most important property of a semantics is its admissibility. An extension E is called *admissible* if and only if

1. E is *conflict-free*, i. e., there are no arguments $a, b \in E$ with $a\mathsf{R}b$ and
2. E *defends* every $a \in E$,

and it is called *complete* (CO) if, additionally, it satisfies

3. if E defends a then $a \in E$.

Different types of classical semantics can be phrased by imposing further constraints. In particular, a complete extension E

- is *grounded* (GR) if and only if E is minimal,
- is *preferred* (PR) if and only if E is maximal, and
- is *stable* (ST) if and only if $\mathsf{A} = E \cup \{b \mid \exists a \in E : a\mathsf{R}b\}$.

All statements on minimality/maximality are meant to be with respect to set inclusion. Note that the grounded extension is uniquely determined and that stable extensions may not exist [18].

Example 1. Consider the abstract argumentation framework AF_1 depicted as a directed graph in Figure 1. In AF_1 there are three complete extensions E_1, E_2, E_3 defined via

[3] `http://www.satcompetition.org`

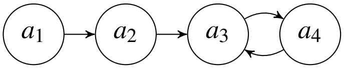


Figure 1. Abstract argumentation framework AF_1 from Example 1.

$$E_1 = \{a_1\}$$
$$E_2 = \{a_1, a_3\}$$
$$E_3 = \{a_1, a_4\}$$

E_1 is also grounded and E_2 and E_3 are both stable and preferred.

Let $\sigma \in \{\mathsf{CO}, \mathsf{GR}, \mathsf{ST}, \mathsf{PR}\}$ be some semantics and $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an abstract argumentation framework. Then, an argument $a \in \mathsf{A}$ is *skeptically accepted* in AF if a is contained in *every* σ-extension. An argument $a \in \mathsf{A}$ is *credulously accepted* in AF if a is contained in *some* σ-extension.

An equivalent way of defining different types of semantics is by means of *labellings*, rather than extensions [5,12]. Given a set of arguments S, a *labelling* is a total function $\mathsf{L} : S \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. An argument $a \in S$ is either labelled in-meaning a is accepted, labelled out-meaning a is rejected-or labelled undec-meaning the status of a is undecided. Given an $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, the set of all *labellings* is denoted as $\mathfrak{L}(\mathsf{AF})$. A *labelling* $\mathsf{L} \in \mathfrak{L}(\mathsf{AF})$ is called a *complete labelling* if and only if for any $a \in \mathsf{A}$ holds:

1. $\mathsf{L}(a) = \texttt{in} \Leftrightarrow \forall b \in a^-, \mathsf{L}(b) = \texttt{out}$;
2. $\mathsf{L}(a) = \texttt{out} \Leftrightarrow \exists b \in a^-, \mathsf{L}(b) = \texttt{in}$;

Comparable to the extension-based definition of semantics, other semantics can be phrased by imposing further constraints to a *complete labelling*. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be an argumentation framework. A *complete labelling* $\mathsf{L} \in \mathfrak{L}(\mathsf{AF})$

- is *grounded* if and only if L is maximising the set of arguments labelled undec,
- is *preferred* if and only if L is maximising the set of arguments labelled in, and
- is *stable* if and only if there is no argument labelled undec.

The following definition further emphasises the inherent connection between extensions and labellings: Let $\texttt{in}(\mathsf{L}) = \{a \in \mathsf{A} | \mathsf{L}(a) = \texttt{in}\}$ and $\texttt{out}(\mathsf{L})$ resp. $\texttt{undec}(\mathsf{L})$ be defined analogously. A labelling L is a complete (grounded, preferred, stable) labelling if and only if $\texttt{in}(\mathsf{L})$ is a complete (grounded, preferred, stable) extension.

3. A minimal SAT-based solver: miniAF

MINIAF[4] is a lightweight SAT-based solver for reasoning tasks in abstract argumentation. It is implemented in the C programming language and based on the {j}ArgSemSAT [14,16] approach. To solve any reasoning task for a given AF, MINIAF traverses the search space of a complete extension via a SAT solver. In general, this task can be broken down in three sub-tasks: (1) Encoding the constraints analogous to a *complete la-*

[4] Source code available at https://github.com/jklein94/miniAF.

belling of the AF as a propositional formula; (2) iteratively modify or generate new formulæ based on previous found models and the reasoning task; and (3) using an external SAT solver to search for models of these formulæ.

The system is capable of solving the following tasks:

- EE-σ: Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ enumerate all sets $E \subseteq \mathsf{A}$ that are σ-extensions.
- SE-σ: Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ return some set $E \subseteq \mathsf{A}$ that is a σ-extension.
- DC-σ: Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, $a \in \mathsf{A}$ decide if a is credulously accepted under σ.
- DS-σ: Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, $a \in \mathsf{A}$ decide if a is skeptically accepted under σ.

for $\sigma \in \{\mathsf{CO}, \mathsf{GR}, \mathsf{ST}, \mathsf{PR}\}$. The MINIAF solver is parameterisable with any SAT solver, specified by a absolute path, following the commandline interface of the SAT competition. As input MINIAF supports abstract argumentation frameworks in the `ASPARTIX` format [22] and the `Trivial Graph Format`.[5]

In the following section, a more detailed explanation of the used algorithms is given. Since all algorithms are based on traversing the search space of a complete extension, the encoding of a corresponding complete labeling is first defined. Based on this encoding, the procedures for solving the above-stated problems are described for each semantics σ.

3.1. Complete semantics

For a given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, MINIAF constructs a propositional formula Π_{AF}, so that each satisfying assignment of Π_{AF} corresponds to a *complete labelling* of AF. In particular, the following SAT encoding is used [13]. Given $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, with $|\mathsf{A}| = k$ and the bijection $\phi : \{1, ..., k\} \to \mathsf{A}$ an indexing of A. Let $V(\mathsf{AF}) \triangleq \cup_{1 \leq i \leq |\mathsf{A}|} \{I_i, O_i, U_i\}$ be the variables of AF. The conjunction of clauses (1)–(6), defined of the variables $V(\mathsf{AF})$, is an encoding of a *complete labelling*:

$$\bigwedge_{i \in \{1,...,k\}} ((I_i \vee O_i \vee U_i) \wedge (\neg I_i \vee \neg O_i) \wedge (\neg I_i \vee \neg U_i) \wedge (\neg O_i \vee \neg U_i) \tag{1}$$

$$\bigwedge_{\{i | \phi(i)^- = \emptyset\}} (I_i \wedge \neg O_i \wedge \neg U_i) \tag{2}$$

$$\bigwedge_{\{i | \phi(i)^- \neq \emptyset\}} \left(\bigwedge_{\{j | \phi(j) \to \phi(i)\}} \neg I_i \vee O_j \right) \tag{3}$$

$$\bigwedge_{\{i | \phi(i)^- \neq \emptyset\}} \left(I_i \vee \left(\bigvee_{\{j | \phi(j) \to \phi(i)\}} (\neg O_j) \right) \right) \tag{4}$$

[5] See `http://en.wikipedia.org/wiki/Trivial_Graph_Format`

$$\bigwedge_{\{i|\phi(i)^- \neq \emptyset\}} \left(\neg O_i \vee \left(\bigvee_{\{j|\phi(j)\rightarrow\phi(i)\}} I_j \right) \right) \tag{5}$$

$$\bigwedge_{\{i|\phi(i)^- \neq \emptyset\}} \left(\bigwedge_{\{j|\phi(j)\rightarrow\phi(i)\}} \neg I_j \vee O_i \right) \tag{6}$$

The resulting formula is in conjunctive normal form (CNF), as from SAT solvers demanded. To enumerate all extensions, each time a solution s is found, the formula Π_{AF} is updated to the conjunction $\Pi_{\mathsf{AF}} \wedge \neg s$, thus excluding the previous result. This formula is then passed back to the SAT solver to find a solution, i. e. an additional *complete labelling*. The procedure is repeated until no satisfying assignment is found, therefore enumerating all extensions.

To decide the credulous acceptance of an argument a, Π_{AF} is updated to $\Pi_{\mathsf{AF}} \wedge I_{\phi^{-1}(a)}$. If some extension with a labelled as `in` exists, i. e. there is a solution to $\Pi_{\mathsf{AF}} \wedge I_{\phi^{-1}(a)}$, a is credulously accepted. To determine whether a is contained in every complete extension and thus skeptically accepted, MINIAF uses the grounded extension.

3.2. Stable semantics

The *stable labellings* are complete labellings with no argument labelled `undec`. Consequently, they are the solutions to the formula $\Pi'_{\mathsf{AF}} := \Pi_{\mathsf{AF}} \wedge \bigwedge_{a\in A} \neg U_{\phi^{-1}(a)}$, which excludes the label `undec` for every argument. The enumeration of all stable extensions is computed the same way as for the complete semantics.

An argument a is credulously accepted, if there is a solution to the formula $\Pi'_{\mathsf{AF}} \wedge I_{\phi^{-1}(a)}$. The question of whether a is labelled as `in` in every stable labelling can be rephrased as: Is there a stable labelling where a is labelled as `out`? The equivalent formula to this question is $\Pi'_{\mathsf{AF}} \wedge O_{\phi^{-1}(a)}$ [16]. If there is a solution to the formula, a is not accepted. In the case that there is no solution and a *stable labelling* exist, a is skeptically accepted.

3.3. Preferred semantics

The *preferred labellings* are computed by using an evolution of the `PrefSAT` algorithm [16]. In general the algorithm consists of two routines: (1) Iterating over a set of complete labellings to identify the preferred ones and (2) an optimization procedure to maximise complete labellings wrt. set inclusion. The credulous acceptance of an argument a is decided by finding—analogous to the complete semantics—a solution to the formula $\Pi_{\mathsf{AF}} \wedge I_{\phi^{-1}(a)}$. To check whether a is contained in every *preferred labelling*, MINIAF subsequently enumerates all *preferred labellings* until it finds a labelling, where a is not in the set of arguments labelled `in`. If no counterexample is found, the argument is skeptically accepted.

3.4. Grounded semantics

Grounded labellings, i. e. complete labellings maximising the set of `undec` arguments, are computed with basically the same optimization procedure as the preferred labellings.[6] However, the arguments labelled `undec` are maximized, rather than the arguments labelled `in`. Since the grounded extension is unique, the problem of credulous and skeptical acceptance of an argument a are equivalent. If a is contained in the *grounded labelling*, it is credulously and skeptically accepted.

4. Experiments

In this section, we present the results of an experimental analysis, in which we investigated the impact of various state-of-the-art SAT solvers on the performance of MINIAF. This analysis aims to give an overview *if* and *how* the overall performance of a SAT-based system is affected by the choice of the exploited SAT-solver. Below, we give a brief description of the investigated SAT solvers and the experimental setup and subsequently discuss our findings.

4.1. Experimental setup

In our experiments, we compared a total of 12 SAT solvers:

CADICAL [8]: is based on conflict-driven clause learning (CDCL) [27] with inprocessing [24].

GLUCOSE (Version 4.1) [4]: is a CDCL solver heavily based on MINISAT [21], with a special focus on removing useless clauses as soon as possible, and an original restart scheme.

The familiy of MAPLELCMDISTCHRONOBT-DL (Version 3, 2.2 and 2.1) [25]: solvers are based on the SAT Competition 2018 winner MAPLELCMDISTCHRONOBT [28] augmented with duplicate learnts heuristic.

MAPLELCMDISTCBTCOREFIRST [17]: is a hack version of MAPLELCMDISTCHRONOBT. This solver adds only Core First Unit Propagation. The remainder keeps unchanged.

MERGESAT [26]: is a CDCL solver based on the competition winner of 2018, MAPLELCMDISTCHRONOBT, and adds several known techniques as well as some novel ideas.

PADC_MAPLE_LCM_DIST [31]: is based on the SAT Competition 2017 winner MAPLE_LCM_DIST and integrates the periodic aggressive learned clause database cleaning (PADC) strategy [31].

PSIDS_MAPLELCMDISTCHRONOBT [31]: is based on MAPLELCMDISTCHRONOBT and integrates the polarity state independent decaying sum (PSIDS) heuristic.

PICOSAT [7] (Version 965): is an attempt to optimise low-level performance of BooleForce,[7] which shares many of its key features with MiniSAT(version 1.14).

[6] Note that we use a reduction to SAT here as well, despite the fact that the grounded labelling can be computed in polynomial time. We do that because we wish to have a general system that makes use of SAT solvers as often as possible without relying on proprietary algorithms.

[7] `http://fmv.jku.at/booleforce`.

RELAXED_LCMDISTCHRONOBT [11]: is a CDCL-based solver. The method used for this solver aims to improve CDCL solvers by relaxing the backtracking and integrating local search techniques. As a local search solver CCANR [10] is used.

OPTSAT [17]: is a CDCL solver using the core first unit propagation technique.

For the evaluation we used the ICCMA'17 benchmark.[8] This benchmark is made up of three groups: A, B and C. Each group, in turn, consists of 350 instances classified into 5 hardness categories: (1) very easy, (2) easy, (3) medium, (4) hard and (5) too hard. Since the *grounded* labelling is uniquely defined, only the SE and DC problems were employed. According to the ICCMA'17 rules [23], each task was assigned to a group as follows:

- A: DS-PR, EE-PR, EE-CO
- B: DS-ST, DC-ST, SE-ST, EE-ST, DC-PR, SE-PR, DC-CO
- C: DS-CO, SE-CO, DC-GR, SE-GR

For all 14 tasks, MINIAF was run 12 times—every time parameterised with a different SAT solver—on the instances of the corresponding group. A cutoff value of 600 seconds (10 minutes) per instance was imposed. All SAT solvers were executed with their default (and non-parallel) configuration. For each SAT solver and task we recorded: (1) the number of solved instances, (2) the number of unsolved instances and (3) the execution time per solved instance.

We ran the experiments on a virtual machine running Ubuntu 18.04 with a 2.9 GHz CPU core and 8GB of RAM.

4.2. Results

The performance achieved by MINIAF is measured in terms of instance coverage (**Cov.**)—percentage of successfully analysed instances—and Penalised Average Runtime (**PAR10**). The PAR10 score is a hybrid measure, defined as the average of runtimes which counts (1) the runtimes of unsolved instances as ten times the cutoff value and (2) the runtimes of solved instances as the actual runtimes. Thus, it allows runtime to be considered and still setting a strong focus on instance coverage. The results of this analysis, with regards to the different semantics, are shown in Table 1 (CO track), Table 2 (ST track), Table 3 (PR track) and Table 4 (GR track). The first column (**SAT**) contains the names of the used SAT solvers. Hereinafter, we will refer to MINIAF just with the name of the used SAT solver to express MINIAF was parameterised with this solver.

Considering the performance achieved on all instances of a track (**ALL**), most SAT solvers are generally comparable. The CADICAL solver performs best (PAR10 score and coverage) for the CO, ST and GR track. As for the PR track, the MAPLELCMDISTCBTCOREFIRST system accomplished the best results. However, the fact that a concrete SAT solver excels all other systems on the whole set of instances, does not necessarily mean this solver exhibits the best performance for all computational tasks of the considered semantics. Rather, we note that for three (CO, ST, PR) of the four semantics, there is at least one task where the overall best solver for this track is outperformed

[8] A more detailed description of the ICCMA'17 benchmark and the selection process can be found here `http://argumentationcompetition.org/2017/benchmark_selection_iccma2017.pdf`.

					CO					
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	**1027.65**	**83.04**	3009.22	50.29	**35.83**	**99.43**	**41.75**	**99.33**	882.95	85.43
Glucose	1100.00	82.15	3084.61	49.14	131.47	98.29	197.61	97.33	857.38	86.00
MapleLCMDistChronoBT-DL-v2.1	1057.65	82.89	2974.07	51.14	229.29	96.57	230.37	96.67	678.69	89.14
MapleLCMDistChronoBT-DL-v2.2	1065.61	82.74	2988.36	50.86	244.73	96.29	230.38	96.67	679.64	89.14
MapleLCMDistChronoBT-DL-v3	1115.78	81.85	3022.13	50.29	245.54	96.29	249.69	96.33	822.01	86.57
MapleLCMdistCBTcoreFirst	1092.59	82.30	2968.79	51.14	269.04	96.00	330.52	95.00	693.15	88.86
MergeSAT	1054.95	82.89	2965.29	51.14	197.32	97.14	230.39	96.67	709.02	88.57
PADC_Maple_LCM_Dist	1050.67	82.96	**2947.17**	**51.43**	246.77	96.29	230.77	96.67	**660.84**	**89.43**
PSIDS_MapleLCMDistChronoBT	1055.16	82.89	2950.45	**51.43**	228.95	96.57	248.92	96.33	677.13	89.14
PicoSAT	1093.74	81.93	3212.29	46.86	35.90	**99.43**	41.90	**99.33**	934.61	84.57
Relaxed_LCMDistChronoBT	1128.34	81.78	3121.88	48.86	213.51	96.86	230.26	96.67	819.41	86.86
optsat	1156.70	81.70	3183.14	47.43	258.05	97.14	306.84	96.67	757.36	87.71

Table 1. Performance comparison for all instances (ALL) and the different tasks of the CO track: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

					ST					
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	**959.40**	**84.43**	**1717.27**	**72.00**	**589.89**	**90.57**	**834.88**	**86.57**	695.56	88.57
Glucose	1321.71	78.71	2353.80	61.71	861.00	86.57	1249.78	80.00	822.25	86.57
MapleLCMDistChronoBT-DL-v2.1	1161.59	81.50	2232.03	64.00	740.48	88.57	1044.60	83.43	629.24	**90.00**
MapleLCMDistChronoBT-DL-v2.2	1160.10	81.57	2196.68	64.57	730.31	88.86	1035.16	83.71	678.27	89.14
MapleLCMDistChronoBT-DL-v3	1219.18	80.50	2180.47	64.86	837.35	86.86	1117.99	82.29	740.91	88.00
MapleLCMdistCBTcoreFirst	1156.61	81.57	2210.21	64.29	707.02	89.14	1064.30	83.14	644.90	89.71
MergeSAT	1146.66	81.71	2206.40	64.29	722.25	88.86	1029.71	83.71	628.27	**90.00**
PADC_Maple_LCM_Dist	1136.12	81.86	2171.51	64.86	702.17	89.14	1043.43	83.43	**627.36**	**90.00**
PSIDS_MapleLCMDistChronoBT	1151.37	81.64	2159.23	65.14	737.93	88.57	1080.59	82.86	627.75	**90.00**
PicoSAT	1416.85	76.64	2251.46	62.86	1074.08	82.29	1474.78	75.71	867.06	85.71
Relaxed_LCMDistChronoBT	1326.29	78.93	2528.81	59.14	875.66	86.57	1145.96	82.00	754.72	88.00
optsat	1130.05	82.00	2060.24	66.86	736.38	88.57	1016.62	84.00	706.97	88.57

Table 2. Performance comparison for all instances (ALL) and the different tasks of the ST track: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

					PR					
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	1383.89	77.33	**2537.02**	**58.29**	1039.66	83.14	1024.59	83.33	882.95	85.43
Glucose	1414.81	76.96	2710.77	55.43	1027.31	83.71	1005.27	83.67	857.38	86.00
MapleLCMDistChronoBT-DL-v2.1	1294.54	79.04	2610.56	57.14	896.74	86.00	941.76	84.67	678.69	89.14
MapleLCMDistChronoBT-DL-v2.2	1286.44	79.19	2594.44	57.43	862.88	86.57	962.53	84.33	679.64	89.14
MapleLCMDistChronoBT-DL-v3	1360.39	77.93	2665.65	56.29	934.23	85.43	962.87	84.33	822.01	86.57
MapleLCMdistCBTcoreFirst	**1262.10**	**79.56**	2608.12	57.14	**776.62**	**88.00**	**921.93**	**85.00**	693.15	88.86
MergeSAT	1301.08	78.89	2576.44	57.71	908.22	85.71	962.22	84.33	709.02	88.57
PADC_Maple_LCM_Dist	1279.18	79.26	2605.32	57.14	843.38	86.86	961.86	84.33	**660.84**	**89.43**
PSIDS_MapleLCMDistChronoBT	1289.68	79.11	2591.29	57.43	879.38	86.29	964.45	84.33	677.13	89.14
PicoSAT	1575.31	74.07	2743.64	54.86	1303.45	78.57	1276.93	79.00	934.61	84.57
Relaxed_LCMDistChronoBT	1467.90	76.37	2868.76	53.14	1093.36	82.86	1027.09	83.67	819.41	86.86
optsat	1319.90	78.59	2629.04	56.86	907.54	85.71	929.98	**85.00**	757.36	87.71

Table 3. Performance comparison for all instances (ALL) and the different tasks of the PR track: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

by another system. The only exception is the GR semantics. For the GR track CaDiCal performs best on all instances and each task (SE-GR, DC-GR).

Furthermore, we observe noticeably differences for individual tasks and semantics[9]:

[9] We also carried out a significant analysis of the execution times that show significant differences. As a statistical test, we used the Kruskal–Wallis test and the Dunn-Bonferroni test for post-hoc analysis.

	GR					
SAT	**ALL**		**SE**		**DC**	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	**38.64**	**99.38**	**35.95**	**99.43**	**41.78**	**99.33**
Glucose	235.79	96.92	219.52	97.14	254.78	96.67
MapleLCMDistChronoBT-DL-v2.1	231.39	96.62	231.12	96.57	231.70	96.67
MapleLCMDistChronoBT-DL-v2.2	239.68	96.46	246.43	96.29	231.81	96.67
MapleLCMDistChronoBT-DL-v3	239.85	96.46	246.74	96.29	231.82	96.67
MapleLCMdistCBTcoreFirst	381.63	94.15	356.64	94.57	410.79	93.67
MergeSAT	213.99	96.92	198.40	97.14	232.16	96.67
PADC_Maple_LCM_Dist	239.39	96.46	245.94	96.29	231.76	96.67
PSIDS_MapleLCMDistChronoBT	231.23	96.62	230.71	96.57	231.83	96.67
PicoSAT	38.78	**99.38**	36.06	**99.43**	41.96	**99.33**
Relaxed_LCMDistChronoBT	222.20	96.77	198.69	97.14	249.62	96.33
optsat	308.00	96.46	295.75	96.57	322.30	96.33

Table 4. Performance comparison for all instances (ALL) and the different tasks of the GR track: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface

Two of the examined SAT systems, namely CADICAL and PICOSAT, perform distinctly better on the instances of the GR track. A similar scenario shows the result for the CO semantics in Table 1. Here too, CADICAL and PICOSAT stand out from the other solvers for the CO-SE and CO-DS tasks.[10] It is interesting, however, that the PICOSAT solver achieves the worst results in terms of coverage and PAR10 score for the all other tasks on this track. In addition, we find that some solvers tend to do better for a certain reasoning problem, regardless of the semantics under consideration. For example, the SAT solver PADC_MAPLE_LCM_DIST achieves the best results for the DC problem of semantics CO, ST and PR. The SE problem for semantics CO, ST and GR is best solved by CADICAL. Surprisingly, none of the MAPLELCMDISTCHRONOBT-DL (Version 3, 2.2, 2.1) solvers achieves the best performance for any task, even though they ranked second place for the SAT track (Version 3, 2.2 and 2.1) and first place for the UNSAT (Version 3) and SAT+UNSAT (Version 3, 2.2, 2.1) track in the last years SAT competition.[11]

Apart from the inherent complexity of a particular reasoning problem, the performance of an argumentation system is also affected by the hardness of the instance to be solved. In order to identify deviations in the performance concerning the level of difficulty of an instance, we compared the SAT solvers based on the hardness categories of the benchmark set.

CADICAL is able to solve all instances of the hardness categories Very Easy,[12] Easy and Medium best. For the Hard and Too Hard instances, PADC_MAPLE_LCM_DIST attains the best results. Moreover we observe—covering the previously presented results—that most solvers are comparable, although there are some differences between the categories. For example, CADICAL and PICOSAT perform clearly better for the Easy instances.

Another interesting scenario is shown in Table 7. Albeit, the PICOSAT system is nearly indistinguishable from the best solver for all instances of the Very Easy, Easy and the DC instances of the Medium set, it performs significantly worse for all other Medium

[10] This observation is actually not so surprising, as CO-SE can be answered just as GR-SE and CO-DS and DC-GR are identical.

[11] `http://sat-race-2019.ciirc.cvut.cz/index.php`.

[12] Since the results for the Very Easy instances are practically identical for most solvers, we refrain from presenting them in a table.

	Easy									
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	**207.82**	**96.71**	**789.64**	**87.33**	**67.9**	**99.0**	**87.13**	**98.67**	1.89	**100.0**
Glucose	415.02	93.43	1475.29	76.0	145.83	98.0	258.13	96.0	6.66	**100.0**
MapleLCMDistChronoBT-DL-v2.1	501.59	92.0	1455.93	76.67	348.34	94.5	293.54	95.33	95.12	98.5
MapleLCMDistChronoBT-DL-v2.2	502.05	92.0	1457.71	76.67	348.58	94.5	293.68	95.33	95.05	98.5
MapleLCMDistChronoBT-DL-v3	509.46	91.86	1491.17	76.0	347.43	94.5	296.01	95.33	95.3	98.5
MapleLCMdistCBTcoreFirst	486.63	92.29	1491.94	76.0	265.85	96.0	294.77	95.33	97.34	98.5
MergeSAT	508.19	91.86	1490.15	76.0	344.68	94.5	295.05	95.33	95.07	98.5
PADC_Maple_LCM_Dist	507.85	91.86	1488.85	76.0	345.6	94.5	293.44	95.33	95.16	98.5
PSIDS_MapleLCMDistChronoBT	492.17	92.14	1415.43	77.33	344.74	94.5	294.72	95.33	95.25	98.5
PicoSAT	251.28	96.0	912.25	85.33	69.34	**99.0**	165.61	97.33	**1.75**	**100.0**
Relaxed_LCMDistChronoBT	629.48	90.0	1811.17	71.0	338.86	94.67	375.38	94.0	89.93	98.67
optsat	466.13	92.71	1406.24	77.33	247.87	96.5	302.67	95.33	101.89	98.5

Table 5. Performance comparison for all instances (ALL) and the different tasks of the Easy instances: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

	Medium									
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	**414.46**	**93.43**	**1507.2**	**75.67**	**200.05**	**97.0**	**157.16**	**97.67**	**2.29**	**100.0**
Glucose	577.71	91.57	1857.93	70.33	280.67	**97.0**	349.88	95.33	85.45	99.25
MapleLCMDistChronoBT-DL-v2.1	580.02	91.43	1800.67	71.33	328.18	96.0	304.37	96.0	123.11	98.5
MapleLCMDistChronoBT-DL-v2.2	582.79	91.36	1796.89	71.33	339.31	95.75	306.31	96.0	123.07	98.5
MapleLCMDistChronoBT-DL-v3	594.49	91.14	1816.03	71.0	355.57	95.5	322.83	95.67	121.01	98.5
MapleLCMdistCBTcoreFirst	631.71	90.5	1796.96	71.33	399.79	94.75	360.5	95.0	193.08	97.25
MergeSAT	563.79	91.64	1812.36	71.0	265.99	**97.0**	302.58	96.0	121.06	98.5
PADC_Maple_LCM_Dist	580.02	91.36	1791.13	71.33	338.01	95.75	302.25	96.0	122.04	98.5
PSIDS_MapleLCMDistChronoBT	577.46	91.43	1794.49	71.33	311.87	96.25	322.7	95.67	121.35	98.5
PicoSAT	738.02	87.93	2161.91	64.67	577.44	90.5	509.28	91.67	2.24	**100.0**
Relaxed_LCMDistChronoBT	829.19	87.38	2684.03	56.5	466.09	93.67	394.88	94.5	204.1	97.33
optsat	601.44	91.36	1801.61	71.33	371.92	95.75	306.55	96.33	152.0	98.25

Table 6. Performance comparison for all instances (ALL) and the different tasks of the Medium instances: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

	Hard									
SAT	ALL		EE		SE		DS		DC	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	952.46	84.5	3390.94	44.33	**413.22**	**93.4**	**553.84**	**91.14**	235.99	96.22
Glucose	1125.6	81.81	3728.3	38.67	646.12	89.8	744.06	88.0	219.98	96.89
MapleLCMDistChronoBT-DL-v2.1	966.19	84.44	3407.01	44.33	473.8	92.6	591.66	90.57	177.36	97.33
MapleLCMDistChronoBT-DL-v2.2	964.68	84.5	3366.79	45.0	476.57	92.6	612.96	90.29	179.19	97.33
MapleLCMDistChronoBT-DL-v3	1003.63	83.88	3433.98	44.0	557.75	91.2	616.48	90.29	179.94	97.33
MapleLCMdistCBTcoreFirst	990.34	84.06	3358.3	45.0	514.61	92.0	614.44	90.29	232.66	96.44
MergeSAT	946.33	84.75	3298.24	**46.0**	462.17	92.8	609.01	90.29	178.72	97.33
PADC_Maple_LCM_Dist	**940.79**	**84.81**	**3292.2**	**46.0**	448.95	93.0	610.44	90.29	**176.59**	**97.33**
PSIDS_MapleLCMDistChronoBT	946.44	84.75	3298.69	46.0	461.65	92.8	611.18	90.29	177.69	97.33
PicoSAT	1155.87	81.06	3775.77	37.67	610.91	90.0	807.0	86.86	286.11	95.56
Relaxed_LCMDistChronoBT	1042.21	83.33	3578.9	41.5	490.12	92.25	623.38	90.0	219.18	96.86
optsat	1021.56	84.06	3496.95	42.67	539.23	92.4	635.53	90.57	207.46	97.33

Table 7. Performance comparison for all instances (ALL) and the different tasks of the Hard instances: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

instances. The performance of PicoSAT is also in the lower range for the Hard and Too Hard set. The opposite is the case for the Relaxed_LCMDistChronoBT solver. It tends to achieve similar—for two problems even higher—coverage for the Hard and Too Hard instances, but lower coverage for the Very Easy, Easy and Medium set. We can derive that a good result on a particular hardness category (or problem), does not

	Too Hard									
SAT	**ALL**		**EE**		**SE**		**DS**		**DC**	
	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.	PAR10	Cov.
CaDiCal	3313.65	45.0	6000.0	0.0	2949.47	51.0	2387.0	60.67	2555.19	57.67
Glucose	3464.51	42.43	6000.0	0.0	3010.92	50.0	2627.89	56.67	2766.28	54.0
MapleLCMDistChronoBT-DL-v2.1	3107.57	48.71	6000.0	0.0	2837.28	53.0	2498.8	58.67	2055.83	66.67
MapleLCMDistChronoBT-DL-v2.2	3109.57	48.71	6000.0	0.0	2731.88	55.0	2464.64	59.33	2112.71	65.67
MapleLCMDistChronoBT-DL-v3	3347.25	44.43	6000.0	0.0	2891.96	52.0	2653.66	56.0	2519.44	58.33
MapleLCMdistCBTcoreFirst	3115.65	48.57	6000.0	0.0	2673.72	56.0	2538.8	58.0	2109.2	65.67
MergeSAT	3140.19	48.14	6000.0	0.0	2901.41	52.0	2466.64	59.33	2126.66	65.33
PADC_Maple_LCM_Dist	**3072.81**	**49.29**	6000.0	0.0	2719.46	55.0	2497.63	58.67	**2014.59**	**67.33**
PSIDS_MapleLCMDistChronoBT	3132.51	48.29	6000.0	0.0	2897.68	52.0	2581.88	57.33	2052.36	66.67
PicoSAT	3569.55	40.57	6000.0	0.0	3066.47	49.0	3011.66	50.0	2800.98	53.33
Relaxed_LCMDistChronoBT	3206.9	46.92	6000.0	0.0	**2538.58**	**58.0**	**2136.9**	**65.0**	2447.71	59.6
optsat	3231.18	46.57	6000.0	0.0	2950.19	51.0	2447.11	60.0	2332.47	61.67

Table 8. Performance comparison for all instances (ALL) and the different tasks of the Too Hard instances: Used SAT solver, instance coverage and PAR10 score. Best result highlighted in boldface.

necessarily transfer to other categories.

5. Summary and Conclusion

In this paper, we compared the performance of MINIAF parameterised with 12 different state-of-the-art SAT solvers on the ICCMA17 benchmark. The results of our analysis shows that: (1) the performance of most SAT solvers is generally comparable for all considered problems, but (2) some systems tend to be more suitable for individual reasoning tasks than others. These insights indicate that the use of SAT-based portfolio systems—i. e., systems that select different SAT solvers depending on instance and task information—may be beneficial for addressing a wide variety of abstract argumentation problems. Moreover, since all SAT solvers have been evaluated with their standard configurations, future work could investigate the influence of various parameter configurations on performance.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (project number 375588274).

References

[1] M. Alviano. The pyglaf argumentation reasoner. In *The Third International Competition on Computational Models of Argumentation (ICCMA'19)*, 2019.

[2] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, and S. Villata. Toward artificial argumentation. *AI Magazine*, 38(3):25–36, October 2017.

[3] G. Audemard and L. Simon. Lazy clause exchange policy for parallel SAT solvers. In *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, pages 197–205, 2014.

[4] G. Audemard and L. Simon. Glucose and Syrup in the SAT'17. *Proceedings of SAT Competition*, pages 16–17, 2017.

[5] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The knowledge engineering review*, 26(4):365–410, 2011.

[6] P. Baroni, M. Caminada, and M. Giacomin. Abstract argumentation frameworks and their semantics. In P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors, *Handbook of Formal Argumentation*, pages 159–236. College Publications, 2018.

[7] A. Biere. Picosat essentials. *Journal on Satisfiability, Boolean Modeling and Computation*, 4(2-4):75–97, 2008.

[8] A. Biere. Cadical at the sat race 2019. *SAT RACE 2019*, page 8, 2019.

[9] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185. IOS Press, 2009.

[10] S. Cai, C. Luo, and K. Su. Ccanr: A configuration checking based local search solver for non-random satisfiability. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 1–8, 2015.

[11] S. Cai and X. Zhang. Four relaxed CDCL Solvers. *SAT RACE 2019*, page 35, 2019.

[12] Martin W.A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2–3):109–145, 2009.

[13] F. Cerutti, P. E. Dunne, M. Giacomin, and M. Vallati. Computing preferred extensions in abstract argumentation: A sat-based approach. In *International Workshop on Theorie and Applications of Formal Argumentation*, pages 176–193. Springer, 2013.

[14] F. Cerutti, M. Giacomin, and M. Vallati. Argsemsat: Solving argumentation problems using sat. *COMMA*, 14:455–456, 2014.

[15] F. Cerutti, M. Giacomin, and M. Vallati. How we designed winning algorithms for abstract argumentation and which insight we attained. *Artificial Intelligence*, 276:1–40, 2019.

[16] F. Cerutti, M. Vallati, and M. Giacomin. jargsemsat: an efficient off-the-shelf solver for abstract argumentation frameworks. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 541–544, 2016.

[17] J. Chen. Smallsat, Optsat and MapleLCMdistCBTcoreFirst: Containing Core First Unit Propagation. *SAT RACE 2019*, page 31, 2019.

[18] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[19] W. Dvořák and P. E. Dunne. Computational problems in formal argumentation and their complexity. In P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors, *Handbook of Formal Argumentation*, chapter 14. 2018.

[20] N. Eén and N. Sörensson. An extensible sat-solver. In *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, pages 502–518, 2003.

[21] N. Eén and N. Sörensson. An extensible sat-solver. In *International conference on theory and applications of satisfiability testing*, pages 502–518, 2003.

[22] U. Egly, S. A. Gaggl, and S. Woltran. Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, 1(2):147–177, 2010.

[23] S. A. Gaggl, T. Linsbichler, M. Maratea, and S. Woltran. Design and results of the second international competition on computational models of argumentation. *Artificial Intelligence*, 279:103193, 2020.

[24] M. Järvisalo, M. J. H. Heule, and A. Biere. Inprocessing rules. In *International Joint Conference on Automated Reasoning*, pages 355–370. Springer, 2012.

[25] S. Kochemazov, O. Zaikin, V. Kondratiev, and A. Semenov. MapleLCMDistChronoBT-DL, duplicate learnts heuristic-aided solvers at the SAT Race 2019. *SAT RACE 2019*, page 24, 2019.

[26] N. Manthey. Mergesat. *Proceedings of SAT Competition*, 2019:29, 2019.

[27] J. Marques-Silva, I. Lynce, and S. Malik. Conflict-driven clause learning sat solvers. In *Handbook of satisfiability*, pages 131–153. IOS Press, 2009.

[28] A. Nadel and V. Ryvchin. Chronological backtracking. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 111–121, 2018.

[29] A. Niskanen and M. Järvisalo. μ-toksia Participating in ICCMA 2019. In *The Third International Competition on Computational Models of Argumentation (ICCMA'19)*, 2019.

[30] F. Pu, H. Ya, and G. Luo. argmat-sat: Applying sat solvers for argumentation problems based on boolean matrix algebra. In *The Second International Competition on Computational Models of Argumentation (ICCMA'17)*, 2017.

[31] R. K. Tchinda and C. T. Djamegni. PADC MapleLCMDistChronoBT, PADC Maple LCM Dist and PSIDS MapleLCMDistChronoBT in the SR19. *SAT RACE 2019*, page 33, 2019.

[32] M. Thimm and S. Villata. The first international competition on computational models of argumentation: Results and analysis. *Artificial Intelligence*, 252:267–294, August 2017.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200510

Towards an Argument Mining Pipeline Transforming Texts to Argument Graphs

Mirko LENZ [1], Premtim SAHITAJ, Sean KALLENBERG, Christopher COORS, Lorik DUMANI, Ralf SCHENKEL, and Ralph BERGMANN
Trier University, Trier, Germany

Abstract. This paper tackles the automated extraction of components of argumentative information and their relations from natural language text. Moreover, we address a current lack of systems to provide a complete argumentative structure from arbitrary natural language text for general usage. We present an argument mining pipeline as a universally applicable approach for transforming German and English language texts to graph-based argument representations. We also introduce new methods for evaluating the performance based on existing benchmark argument structures. Our results show that the generated argument graphs can be beneficial to detect new connections between different statements of an argumentative text.

Keywords. computational argumentation, argument mining, argument graph construction, argument graph metrics

1. Introduction

Argumentation plays an integral role in many aspects of daily human interaction. People use arguments to form opinions, discuss ideas or change the views of others. Many resources dealing with argumentation are available, but the content is mostly unstructured. Due to the current capabilities of modern hardware, Computational Argumentation (CA) is a field of increasing interest. While previous work [1,2] has focused rather on individual tasks such as claim detection [3], this paper targets the automated extraction of argumentative components and their relations from natural language text. We address a gap in the argument mining field where end-to-end pipelines that generate complex argument structures for CA are not prevalent. We present such a pipeline that provides a universally applicable approach for transforming German and English language texts to graph-based representations [4] in the popular AIF format [5]. We also introduce new methods for evaluating results based on benchmark data and present a new argument graph corpus.

2. Foundations and Related Work

Argumentation, in a formal way, is described as a set of arguments in texts. An argument is constructed by at least two *Argumentative Discourse Units* (ADUs) which represent

[1] Corresponding author. E-mail: `info@mirko-lenz.de`

different components of argumentation, e.g. claims and premises. Additionally, we can represent the stance between two ADUs as a supporting or attacking directed relation. A *major claim* is defined as the claim that describes the key concept in an argumentative text [2]. An *argument graph* describes a structured representation of argumentative text [6]. We use a variant of the well-known Argument Interchange Format (AIF) [5], extended to support the explicit annotation of a major claim M [7]. Claims, premises, and the major claim are represented as *information nodes* (I-nodes) I while relations between them are represented by *scheme nodes* (S-nodes) S. We define an argument graph G as triple $G = (V, E, M)$ with a set of nodes $V = I \cup S$ and a set of edges $E \subseteq V \times V$.

We aim at addressing the research gap of a general-use end-to-end pipeline for the German and English languages by following and extending the approaches of related work in the field. Cabrio and Villata [8] define the central stages of an argument mining framework to be argument extraction and relation prediction. Stab and Gurevych [2] present an approach for extracting arguments by identifying ADUs with further classification into major claim, claims, and premises by considering structural, lexical, syntactical, and contextual features [3]. The segmentation of natural language text into ADUs is simplified by considering textual boundaries on the sentence level [9]. Many researchers formulate relation prediction as a binary classification problem to distinguish between support and attack [10]. The argumentative information is then used to construct an argument graph from the extracted ADUs [8]. To the best of our knowledge, only Stab and Gurevych [2] addressed a method to link ADUs within the same paragraph in an argumentative text. Nguyen and Litman [11] developed a specialized end-to-end argument mining system that includes the identification of relevant ADUs, the classification of components as well as the prediction of their relations.

To assist argument mining techniques, a diverse selection of corpora exists. Stab et al. [12] and Eger et al. [13] provide a corpus with 402 annotated persuasive essays—in the following called PE. It consists of $11,078$ nodes and $10,676$ edges. Another corpus has been developed by the ReCAP project [14], composed of 100 argument graphs dealing with educational issues in Germany. It consists of $4,814$ nodes and $4,838$ edges.

3. Argument Mining Pipeline

The pipeline introduced by Nguyen and Litman [11] is used as the basis of our proposed architecture and extended by a novel graph construction process. Our pipeline is designed in a modular way where each step describes an individual and interchangeable module.

Argument Extraction As a first step, the input text is segmented into sentences [2,14]. Then, multiple types of features are extracted, derived from Stab and Gurevych [12] as well as Lippi et al. [15], depicted in our GitHub project.[2] The basis of the entire approach is the correct identification of ADUs. Based on these features, the sentences are classified into argumentative and non-argumentative units. The ADUs are then further categorized into claims and premises using a separate classifier.

[2] https://github.com/ReCAP-UTR/Argument-Graph-Mining, licensed under Apache 2.0.

Relationship Type Classification To construct an argument graph from a natural language text, it is necessary to consider the task of textual entailment. Here, we assign the relation type between the identified ADUs [16]. We consider only the inference from premises to claims. Due to the complexity of considering a multi-class stance problem and the lack of training data of more sophisticated argument schemes (e.g., Walton et al. [17]), we train a model to only classify attacking and supporting relations. GloVe embeddings are used as the only feature for this task to focus on semantic information. Based on the model's metadata, we detect indifferent results (i.e., having a classification probability below a configurable threshold). In this case, the type support is used.

Major Claim Detection A very crucial step in the graph generation is the location of the major claim. Neither pretrained models nor sufficient training data are available, as each text usually has only one major claim, regardless of its length, making machine learning-based approaches infeasible. The classifier by Stab et al. [2] cannot be applied as it condenses all classification steps into a single model, which does not fit our proposed pipeline. Thus, we examine the following heuristics:

FIRST: The first claim based on the text position is chosen as the major claim. This is done because the main argument is often referred to in the introduction or headline (e.g., Dumani et al. [14]). CENTROID: When treating the major claim as the core proposition of the text, we can assume that it should be very similar to all ADUs. Thus, we can compute the centroid of all embeddings to estimate the core message. The major claim is then the ADU with the highest cosine similarity to the centroid. PAIRWISE: Pairwise cosine similarity of all embeddings of the ADUs is computed. The major claim is defined as having the highest average similarity to all other ADUs. The rational for this technique is similar to CENTROID. PROBABILITY: Again, a cross product of all ADUs is computed. Based on the relationship classification (see above), the major claim is defined as having the highest average classification probability except for neutral results, (i.e., we select the ADU where the model shows the highest certainty in all of its predicted relations).

Graph Construction Utilizing the acquired information, we can now construct the graph. To the best of our knowledge, there is no automatic procedure that links ADUs to complex graphs. We propose three algorithms to address this task. In all cases, ADUs are used as I-nodes and the S-nodes between them are derived from the relationship type classification. As a simplification, the major claim is set as the root.

FLAT TREE: Our baseline approach connects all ADUs as I-nodes to the major claim using the predicted S-nodes, resulting in a two-layer graph. While not suitable for complex texts, it may still provide sufficient results for smaller ones. ADU POSITION: This technique makes use of typical argument compositions. We assume that premises belonging to a claim are contained in the same paragraph and thus positioned in close proximity of the claim in the original text [2]. In the first step, all claim I-nodes are connected to the major claim using the respective S-nodes. Then, each premise I-node is connected to the nearest claim via an S-node. If no claim is detected, all premise I-nodes are connected directly to the major claim via S-nodes. The resulting graph consists of at least two and at most three layers. PAIRWISE COMPARISON: This method leverages the class probabilities of the relationship type classification. Its idea is to draw an edge between ADUs whose relation probability is above a certain threshold. First of all, tuples of ADUs (a, b) are computed such that b has the highest relation probability among all possible connections of a. If multiple ADUs reach the same maximal value, the first one is chosen. Then, a configurable lower bound (in our case 0.98) below this maximal probability is defined. Each ADU related to the major claim with a score above the lower bound is connected as an I-node via a corresponding S-node. If the major claim has no connections after this step, the ADU that first occurs in the text is used as an I-node and connected to the major claim. Then, the remaining ADUs are connected iteratively (via S-nodes) to the I-node where their score is above the lower bound. If there remain ADUs not used after a certain amount of repetitions, they are connected to the major claim using a support S-node.

4. Experimental Evaluation

In this section we evaluate our end-to-end approach by assessing the resulting argument graph structures. Moreover, we compare the correspondence of our automatically generated graph to a given benchmark graph.

Hypotheses The following hypotheses, covering all aspects of the pipeline, will be tested in our evaluation: **(H1)** Using sentences as an argumentative unit yields a robust approximation of the manual segmentation. **(H2)** Selecting the major claim using FIRST will give the best results as it reflects common argumentation patterns. **(H3)** Using a threshold for the relationship type classification (i.e., a value above 0.5) will perform best as supporting arguments occur more often than attacking ones. **(H4)** Using ADU POSITION to construct graphs will result in the best approximation of the benchmark data due to the claim-premise information. **(H5)** Providing the pipeline with predefined ADUs will result in graphs that better reflect the human annotation than end-to-end graphs.

Experimental Setup and Datasets The implementation has been done in Python and is available on GitHub. Three datasets are used for the evaluation: ReCAP, PE (see Section 2) and a new one created for our tasks. The ReCAP corpus contains fragments such as headlines and metadata that were removed manually from the input files. We are using two versions of the PE dataset. PE_{17} is based on Stab et al. [12]. The length of the ADUs differs greatly and is not in line with our sentence-based segmentation. PE_{18} is based on Eger et al. [13] and was transformed by us from word- to sentence-based labels to conform to our segmentation approach. A major difference is that PE_{17} has information about relations between ADUs (i.e., available as argument graphs), while PE_{18} only provides the ADUs. We also explored the open discourse platform `kialo.com` due to the availability of much larger argument graphs. We extracted the 589 debates in the popular collection (as of Jan. 2020), consisting of $190,269$ I-nodes, $189,680$ S-nodes and $379,360$ edges. The data is available in English and German (translated via `deepl.com`) on request from the authors.

Classification Models For ADU and claim - premise classification we chose an ensemble stacking method build from a layer of a logistic regression, random forest and adaptive boosted decision tree [18] as they were shown to perform well for those specific tasks [19]. The classifiers' first layer adds their predictions as feature to the input features and passes them on to the final estimator which provides the output prediction. For the output layer we chose extreme gradient boosted random forest [20]. The ADU model was trained using the PE_{18} and ReCAP datasets in their respective native languages (i.e., German for ReCAP and English for PE_{18}) to mitigate any translation errors. The claim-premise classifier was trained using PE_{18} for both languages as it is the only one that

Table 1. Results of the ADU and claim-premise classification. $A :=$ Accuracy, $P :=$ Precision, $R :=$ Recall

(a) ADU model.

Language	A	P	R	F_1
English (PE_{18})	0.80	0.80	1.0	0.89
German (ReCAP)	0.54	0.52	0.66	0.58

(b) Claim-premise model.

Language	A	P	R	F_1
English (PE_{18})	0.52	0.52	0.68	0.59
German (PE_{18})	0.76	0.73	0.13	0.22

differentiates between claims and premises while also using sentences as units. To eliminate biases, a 90/10 train/test split has been performed before training. The models were trained through a 5-fold stratified cross-validation on the training set and tuned through a random search. The reported values are results from a single evaluation on the test set.

We observed that the ADU classification reached highly varying results between the two datasets. Probably the limited quantity of training data in the ReCAP dataset is the main reason for the variation. On the more than four times larger essay data we obtained an accuracy score of 0.80 which yields a strong indication of the model's generalization ability. The claim-premise classification unfortunately did not meet expectations on neither the persuasive essays nor on the ReCAP data. We explain the difference in predictive power on both datasets due to the fact that the structure of the ReCAP dataset is too dissimilar to the PE_{18} dataset on which the models are trained on. In Table 1 we report accuracy A, precision P, recall R and F_1 values for the used classification models.

The training of the relationship type model was done with the Kialo dataset due to the large number of available relations. The triples were split into 70% training and 30% testing data. Among state-of-the-art classifiers, extreme gradient boosting achieved the highest accuracy for both languages with 0.678 and 0.668 for the English and German language, respectively. Logistic regression performed very similar (0.672 and 0.664) while being computationally simpler, leading us to choose the latter.

Argument Graph Metrics To assess the quality of the entire pipeline as well as its individual steps, multiple metrics are needed. We are not aware of existing measures that enable the verification of our hypotheses and thus introduce a novel approach. For each element in the benchmark graph (i.e., I-nodes, S-nodes, major claim and edges), the corresponding item in the generated graph is determined to compute an agreement.

To compare the ADU segmentation, we need a mapping between the I-nodes of the benchmark graph G_b and the generated graph G_g. It is based on the Levenshtein distance [21] $\text{dist}(u_b, v_g)$ between the benchmark I-node u_b and the generated I-node v_g and the derived similarity $\text{sim}(u_b, v_g) = 1 - (\text{dist}(u_b, v_v) / \max\{|u_b|, |v_g|\})$. The mapping $m\colon u_b \mapsto v_g$ assigns each I-node of the benchmark graph an I-node of the generated graph s.t. their similarity is higher than any other combination of I-nodes. In case that two generated nodes have the same similarity to the benchmark node, we pick the first one. If the ADU segmentation between the benchmark and generated graph differs, the benchmark node is mapped to the generated node having the highest similarity while ignoring the other nodes. The *I-nodes agreement* $\mathscr{I}$ is defined by the weighted arithmetic mean of the similarity between the benchmark I-nodes and their respective mappings. The *major claim agreement* $\mathscr{M}$ is specified as a binary metric that is 1 iff the major claims are mapped or there is none defined in the benchmark and 0 otherwise.

For the evaluation of S-nodes, we need to consider the surrounding I-nodes, because S-nodes do not contain textual content that could be used for similarity assessments. We compute all combinations of connections of the benchmark S-node $\text{in}(u_b) \times \text{out}(u_b)$ and determine individual tuples based on their respective mappings as $(m(\text{in}), m(\text{out}))$. Using this information, it is possible to compare the benchmark S-node with the information provided by the relationship type classification. The *S-node agreement* $\mathscr{S}$ is then defined as the number of correctly classified relationships divided by the total number of tuples.

Lastly, edges need to be considered as well. As they do not contain textual information, we use the triple (x, y, z) where x and z represent I-nodes and y an S-node. Thus, we consider two edges at a time. The two edges in the benchmark graph are mapped to their counterparts in the generated graph if they connect the same I-nodes (as determined by the mapping m). The direction of the edges is not relevant. The S-node y is ignored deliberately to mitigate potential errors during earlier tasks. The *edges agreement* $\mathscr{E}$ is determined by dividing the number of mapped edges by the total number of edges.

5. Results and Discussion

We will now evaluate the pipeline using the test splits of the German ReCAP corpus and the English PE corpora. Exemplary cases can be found in the extended version [22].

German ReCAP Corpus The test set for the ReCAP corpus contains ten texts with benchmark graphs. We get an *I-node* agreement $\mathscr{I} = 0.461$ for all possible combinations of parameters. In most cases, there were fewer, but larger ADUs in the generated graph compared to the benchmark. This stands in contrast to the fact that the average ADU length in the ReCAP corpus is 1.1, indicating mismatches in the definition of a sentence, for example due to punctuation. It also contradicts **H1**. Table 2a shows the results of the three *major claim* detection approaches. They are very similar, differing only in one case (as we have exactly one major claim per text). The two best methods CENTROID and PAIRWISE predicted exactly the same major claims. As FIRST performed worst here, **H2** might be rejected. All thresholds for the *relationship type* classification are depicted in Table 2b. The best result can be obtained using 1.0 (i.e., the classifier always predicts support), which means that almost all of the relations in the benchmarks are of the type support. With such a skewed distribution, this corpus may not be suitable to assess **H3**, thus we will postpone it to PE. When comparing end-to-end with preset ADUs, we observe that the latter one delivers slightly worse performance with all thresholds above 0.6. This could be caused by the smaller preset ADUs which provide less contextual information for the classifier. This stands in slight contrast to **H5**. Lastly, Table 2c shows the three *graph construction* methods. As the scores depend on the major claim method, we used the best approach (i.e., CENTROID/PAIRWISE) for the end-to-end graph. The algorithm FLAT TREE delivered the best results across the board, contradicting **H4**. As expected, the scores themselves are very low, especially for the end-to-end graph, making manual examination of individual edges necessary. When comparing the end-to-end graph with the one using preset ADUs, we notice a major increase in the agreement score. Using the best method, almost half of the edges were connected correctly, providing support for **H5**. This is in large part caused by using the correct major claim as the root node.

English PE Corpus For the following evaluation, the test split (see Section 4) of the PE corpus is used, consisting of 40 cases. The results of PE_{17} are very similar to the findings of the ReCAP corpus. The I-node agreement $\mathscr{I} = 0.622$ is higher than for the

Table 2. Aggregated results of the evaluation using the ReCAP corpus.

(a) Major claim methods.

Method	$\mathscr{M}$
CENTROID	**.200**
FIRST	.100
PAIRWISE	**.200**
PROBABILITY	.100

(b) Relationship type thresholds.

Threshold	$\mathscr{S}_{\text{e2e}}$	$\mathscr{S}_{\text{preset}}$
0.5	.460	.514
⋮	⋮	⋮
0.9	.927	.898
1.0	**.937**	**.902**

(c) Graph construction methods (CENTROID major claim for e2e).

Method	$\mathscr{E}_{\text{e2e}}$	$\mathscr{E}_{\text{preset}}$
ADU POSITION	.064	.166
FLAT TREE	**.095**	**.449**
PAIRWISE COMP.	.054	.296

ReCAP graphs, providing support for **H1**. CENTROID and PAIRWISE performed best for identifying the major claim ($\mathscr{M} = 0.1$), contradicting **H2**. A threshold of 0.9 for the relationship type classification yields the highest agreements ($\mathscr{S}_{\text{e2e}} = 0.936$ and $\mathscr{S}_{\text{preset}} = 0.912$). Again, the S-node distribution is skewed, but as two different corpora show the same results, we can accept **H3** for certain corpora. The best edge agreement scores can be obtained using ADU POSITION for the end-to-end graph ($\mathscr{E}_{\text{e2e}} = 0.130$) and FLAT TREE for the graph with preset ADUs ($\mathscr{E}_{\text{preset}} = 0.274$). All graph construction methods show a low agreement, thus **H4** needs to be rejected. The use of preset ADUs provides a benefit in the edge agreement with only a small decrease in the S-node agreement, leading to the final acceptance of **H5**. Overall, the findings show the robustness of the proposed approach for varying input data. The PE_{18} dataset provides another perspective on the pipeline by using sentence-based segmentation. The I-node agreement $\mathscr{I} = 0.799$ shows a decent approximation of the segmentation, leading to the partial acceptance of **H1** for certain corpora (e.g., essays). The *major claim* agreement $\mathscr{M}$ is 0.125 for CENTROID and PAIRWISE, 0.175 for PROBABILITY and 0.250 for FIRST. As FIRST was only best in this specific corpus and the values are low overall, we have to reject **H2**.

6. Conclusion and Future Work

In this work, we investigated new methods towards the automated mining of argument graphs from natural language texts for both English and German. The pipeline successfully extends previous approaches [11] by generating even complex graphs as end product. Our results show that there are great differences in the resulting graphs based on the type of input data. For very homogeneous corpora such as PE, the agreement is very high, but in heterogeneous datasets such as ReCAP, the methods performed rather poor. When looking beyond the goal to approximate a human annotation as much as possible, the generated graphs might be very beneficial to detect new connections between single statements of an argumentative text. Using multiple methods to construct different representations from a single text might also help in educating professional annotators by discussing the strengths and weaknesses of individual cases.

In future work we plan to provide a more flexible approach for segmenting a text into potential ADUs. A limitation of the current evaluation procedure lies in the edge agreement, which could be tackled by providing multiple benchmark graphs to account for uncertainty. As the ReCAP corpus makes use of detailed argumentation schemes [17], the pipeline should be extended make use of them. Finally, we will investigate the potential use of argument graphs for the task of measuring argument quality [23] in unstructured texts through the use of argument mining.

Acknowledgments This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project *ReCAP*, Grant Number 375342983 (2018–2020), as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999). We would also like to thank *DeepL* for providing free access to their translation API.

References

[1] Levy R, Bogin B, Gretz S, Aharonov R, Slonim N. Towards an argumentative content search engine using weak supervision. In: COLING; 2018. p. 2066–2081.

[2] Stab C, Gurevych I. Identifying Argumentative Discourse Structures in Persuasive Essays. In: EMNLP; 2014. p. 46–56.

[3] Lippi M, Torroni P. Argument Mining from Speech: Detecting Claims in Political Debates. In: AAAI; 2016. p. 2979–2985.

[4] Craven R, Toni F. Argument Graphs and Assumption-Based Argumentation. Artificial Intelligence. 2016;233:1–59.

[5] Chesñevar C, McGinnis J, Modgil S, Rahwan I, Reed C, Simari G, et al. Towards an Argument Interchange Format. Knowl Eng Rev. 2006;21(4):293–316.

[6] Stede M, Afantenos SD, Peldszus A, Asher N, Perret J. Parallel Discourse Annotations on a Corpus of Short Texts. In: LREC; 2016. .

[7] Lenz M, Ollinger S, Sahitaj P, Bergmann R. Semantic Textual Similarity Measures for Case-Based Retrieval of Argument Graphs. In: ICCBR. vol. 11680 of Lecture Notes in Computer Science; 2019. p. 219–234.

[8] Cabrio E, Villata S. Five Years of Argument Mining: a Data-driven Analysis. In: IJCAI; 2018. p. 5427–5433.

[9] Stab C, Miller T, Schiller B, Rai P, Gurevych I. Cross-topic Argument Mining from Heterogeneous Sources. In: EMNLP; 2018. p. 3664–3674.

[10] Stab C, Gurevych I. Parsing Argumentation Structures in Persuasive Essays. Computational Linguistics. 2017;43(3):619–659.

[11] Nguyen HV, Litman DJ. Argument Mining for Improving the Automated Scoring of Persuasive Essays. In: AAAI; 2018. p. 5892–5899.

[12] Stab C, Gurevych I. Parsing Argumentation Structures in Persuasive Essays. Computational Linguistics. 2017 Sep;43(3):619–659.

[13] Eger S, Daxenberger J, Stab C, Gurevych I. Cross-lingual Argumentation Mining: Machine Translation (and a bit of Projection) is All You Need! In: COLING; 2018. p. 831–844.

[14] Dumani L, Biertz M, Witry A, Ludwig AK, Lenz M, Ollinger S, et al.. The ReCAP Corpus: A Corpus of Complex Argument Graphs on German Education Politics; 2020.

[15] Lippi M, Torroni P. Context-Independent Claim Detection for Argument Mining. In: IC-AI. IJCAI'15; 2015. p. 185–191.

[16] Cabrio E, Villata S. Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions. In: ACL; 2012. p. 208–212.

[17] Walton D, Reed C, Macagno F. Argumentation Schemes; 2008.

[18] Schapire RE. A Brief Introduction to Boosting. In: IJCAJ. IJCAI'99; 1999. p. 1401–1406.

[19] Aker A, Sliwa A, Ma Y, Lui R, Borad N, Ziyaei S, et al. What works and what does not: Classifier and feature analysis for argument mining. In: ArgMining@EMNLP; 2017. p. 91–96.

[20] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. CoRR. 2016;abs/1603.02754.

[21] Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady. 1966 Feb;10:707.

[22] Lenz M, Sahitaj P, Kallenberg S, Coors C, Dumani L, Schenkel R, et al. Towards an Argument Mining Pipeline Transforming Texts to Argument Graphs. arXiv: 2006 04562. 2020;.

[23] Wachsmuth H, Naderi N, Hou Y, Bilu Y, Prabhakaran V, Thijm TA, et al. Computational Argumentation Quality Assessment in Natural Language. In: EACL; 2017. p. 176–187.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200511

Explanation Semantics for Abstract Argumentation

Beishui LIAO [a], Leendert VAN DER TORRE [b,a]
[a] *Zhejiang University*
[b] *University of Luxembourg*

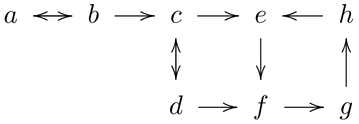
Abstract. This paper studies explanation semantics of argumentation by using a principle-based approach. In particular, we introduce and study explanation semantics associating with each accepted argument a set of such explanation arguments. We introduce various principles for explanation semantics for abstract argumentation, and list various relations among them. Then, we introduce explanation semantics based on defence graphs, and show which principles they satisfy.

Keywords. Argumentation semantics, explanation, defense graph

1. Introduction

In this paper we consider the use of formal argumentation for explainable AI [15]. According to the empirical results reported by Ye and Johnson [19], justification is the most effective type of explanation to bring about changes in user attitudes toward the system. Formal argumentation, as a formalism for representing and reasoning with inconsistent and incomplete information [1,8], provides various ways for explaining why a claim or a decision is made, in terms of justification, dialogue, and dispute trees [11]. Besides some application specific methods such as argumentation-based explanation in case-based reasoning [5] and in scientific debates [18], etc., there are some approaches for defining general theories of explanation about acceptance of arguments in terms of the notion of defense [9,20]. Along this line of work, in this paper, we study a related notion of explanation for abstract argumentation as a kind of semantics: an argument is accepted because some other arguments are accepted, and propose a new semantics, called explanation semantics.

Some basic notions of explanation semantics are illustrated by the following example. The graph below represents an argumentation framework, of which the nodes are called arguments, and the arrows represent attacks between arguments. The graph contains three strongly connected components (SCCs), $\{a, b\}$, $\{c, d\}$ and $\{e, f, g, h\}$, which represents the graph-theoretic property that there is a path from each element to each other element of the SCC. The three preferred extensions are $\{\{a, c, f, h\}, \{a, d, e, g\}, \{b, d, e, g\}\}$.



Dauphin *et al.* [6] observe that in such examples, every strongly connected component can be seen as a choice to accept some attack-free set of arguments of the SCC. For example, if argument a is chosen in the first SCC, then either c or d can be chosen in the second SCC. However, if b is chosen in the first SCC, then only d can be chosen in the second SCC. Thus, the choice in the first SCC determines the set of alternatives in the second SCC. Likewise, whatever is chosen in the first or second SCC, in the third SCC there is only one alternative.

The explanation extensions may be $\{\{a^a, c^c, f^c, h^c\}, \{a^a, d^d, e^d, g^d\}, \{b^b, d^b, e^b, g^b\}\}$. For the first choice between accepting argument a or accepting argument b, each argument is labeled by itself, which expresses that the choice does not depend on other choices. For the choice between accepting argument c and accepting argument d, it partially depends on the first choice. If accepting argument a is chosen, then the choice between accepting c or d is not restricted. Alternatively, if accepting argument b is chosen, then the only choice is to accept d. Finally, either accepting c or accepting d is chosen, the remaining choice is unique.

Thus we say that the reason that g is accepted, is because d is accepted in case a is accepted, or because b is accepted. Distinguishing this kind of explanations provides more information than only the acceptance or rejection of g. Also, we can distinguish direct from indirect reasons, and more.

The layout of this paper is as follows. In Section 2 we introduce the standard terminology of Dung's abstract argumentation and our variant of explanation semantics. In Section 3 we introduce various principles/properties of explanation semantics, and in Section 4 and 5 we introduce some concrete examples of explanation semantics.

2. Abstract Argumentation and Explanation

In this section, we recall some basic notions of abstract argumentation that are used in this paper, and then we introduce explanation semantics.

2.1. Traditional semantics

All notions in this paper are defined on abstract argumentation frameworks, which is a directed graph in which nodes are called arguments and arrows represent attacks between arguments. As usual, we write a^- for the set of attackers of a, and a^+ for the set of arguments a attacks.

Definition 1 (Argumentation framework) *An argumentation framework is a pair $F = (A, \rightarrow)$ where A is a set of arguments and $\rightarrow \subseteq A \times A$ is a binary relation over A, called attacks. An argument a attacking an argument b is written as $a \rightarrow b$. A set of arguments B attacks a, written as $B \rightarrow a$, if there exists $b \in B$ such that $b \rightarrow a$. Given $a \in A$, we define $a_F^- = \{b \in A \mid b \rightarrow a\}$ and $a_F^+ = \{b \in A \mid a \rightarrow b\}$. When $a_F^- = \emptyset$, we say that a is unattacked, or a is an initial argument. When the context is clear, we also write a^+ and a^- for a_F^+ and a_F^- respectively.*

Definition 2 (Traditional argumentation semantics) *Let $\mathcal{F}$ be the set of all argumentation frameworks $F = (A, \rightarrow)$. Let an extension of F be a subset of A. Traditional argumentation semantics is a function σ from $\mathcal{F}$ to sets of their extensions, associating with each argumentation framework F a subset of 2^A, denoted as $\sigma(F)$.*

Given an argumentation framework $F = (A, \rightarrow)$, various types of argument extensions of F can be defined as follows.

Definition 3 (Dung's argumentation semantics) *Let $F = (A, \rightarrow)$ be an argumentation framework, $\mathcal{E} \subseteq A$ be a set of arguments, and $a \in A$ be an argument. $\mathcal{E}$ is conflict-free if and only if there exist no $a, b \in A$ such that $a \rightarrow b$. $\mathcal{E}$ defends a if and only if for each $b \in a^-_F$, $\mathcal{E} \rightarrow b$. $\mathcal{E}$ is admissible if and only if $\mathcal{E}$ is conflict-free, and each argument in $\mathcal{E}$ is defended by $\mathcal{E}$.*

- *$\mathcal{E}$ is a complete extension if and only if $\mathcal{E}$ is admissible, and each argument in A that is defended by $\mathcal{E}$ is in $\mathcal{E}$.*
- *$\mathcal{E}$ is the grounded extension if and only if $\mathcal{E}$ is the minimal (with respect to set-inclusion) complete extension.*
- *$\mathcal{E}$ is a preferred extension if and only if $\mathcal{E}$ is a maximal (with respect to set-inclusion) complete extension.*
- *$\mathcal{E}$ is a stable extension if and only if $\mathcal{E}$ is conflict-free and $\mathcal{E}$ attacks each argument that is not in $\mathcal{E}$.*

2.2. Explanation semantics

In the following definition, an explanation semantics is a function from graphs to sets of explanation extensions, where each explanation extension is a set of explanation arguments, where each explanation is a set of arguments. We use the letter E for extension, and we use the letter R for explanation, which expresses that the explanation is the reason the argument is accepted.

Definition 4 (Explanation semantics) *Let an explanation of each argument in F be a subset of A, and let an explanation extension of F be a subset of A, of which each argument is labeled with an explanation. Explanation semantics is a function Σ from $\mathcal{F}$ to sets of their explanation extensions, denoted as $\Sigma(F)$. We write a^R for the argument a with explanation R. When R contains a single argument (say, b), a^R is also written as a^b for conciseness.*

Each explanation semantics induces a traditional semantics, simply by stripping the labels. In such a case, we say that the explanation semantics explains the traditional semantics.

Definition 5 (Explaining argumentation semantics) *Explanation semantics Σ explains traditional semantics σ iff for all F, we have $\sigma(F) = \{\{x \mid x^R \in E\} \mid E \in \Sigma(F)\}$.*

Definition 6 (Explainable semantics) *A traditional semantics is explainable with respect to properties X iff there is an explanation semantics satisfying properties X, explaining the traditional semantics.*

3. Principles for explanation semantics

We start with three elementary properties. The first property is called U for Uniqueness and says that each accepted argument has a unique explanation.

Property 1 (Uniqueness) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R, a^S \in E$, we have $R = S$.*

The second property is called A for *Acceptance* and says that an explanation consists of a set of accepted arguments.

Property 2 (Acceptance) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R \in E$, the explanation R consists of arguments that are part of the extension $\{x \mid x^S \in E\}$, i.e., $R \subseteq \{x \mid x^S \in E\}$.*

The third property says that the explanation defends the accepted argument, possibly recursively. It uses the following characteristic function returning all arguments in F recursively defended by the arguments in S, which we write as c.

Definition 7 (Characteristic function) $c_0(S, F) = \{a \in F \mid S \text{ defends } a\}$. $c_{i+1}(S, F) = c_0(S \cup c_i(S, F), F)$. $c_\infty(S, F) = \cup_{i=0}^{\infty} c_i(S, F)$.

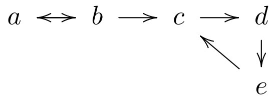
The third property is called I for *Indirect Defense* and says that for all $a^R \in E$, if we iteratively apply the characteristic function to explanation R, then we get a set of arguments containing a.[1]

Property 3 (Indirect Defense) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R \in E$, we have $a \in c_\infty(R, F)$.*

The fourth property strengthens indirect defense to direct defense. Obviously property D implies property I, in the sense that if an explanation semantics satisfies property D, it also satisfies property I.

Property 4 (Direct defense) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R \in E$, we have $a \in c_0(R, F)$.*

Example 1 (Two-three cycle) *Consider the following widely discussed two-three cycle framework:*



There are two preferred extensions $\{a\}$ and $\{b, d\}$ under Dung's argumentation semantics. The unique explanation extensions satisfying Properties UAID and explaining these preferred extensions are $\{a^a\}$ and $\{b^b, d^b\}$.

Proposition 1 (Explainable semantics, Prop. UAID) *All traditional Dung semantics are explainable with respect to Properties UAID.*

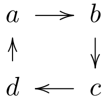
[1] Alternatively, we could require that for all $a^R \in E$, if we iteratively apply the characteristic function to label R, then we get a set of conflict-free arguments containing a. However, since all semantics we consder are conflict free, in the sense that the accepted arguments do not attack each other, we do not consider this variant of Property 3.

Proof For all $\mathcal{E} \in \sigma(\mathcal{F})$, $\forall a \in \mathcal{E}$, since $\mathcal{E}$ defends a, there exist a set of sets $R_1, \ldots, R_n \subseteq \mathcal{E}$ where $n \geq 1$, such that $a \in c_0(R_i, F)$ where $i = 1, \ldots, n$. Let $R_a \in \{R_1, \ldots, R_n\}$ be a minimal set with respect to set inclusion. Let $E = \{a^{R_a} \mid a \in \mathcal{E}\}$. It holds that E satisfies Properties UAID. □

The fifth property says that explanations are minimal in the sense that they do not contain superfluous arguments.

Property 5 (Minimality) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R \in E$, for all $S \subset R$ we have $a \notin c_\infty(S, F)$.*

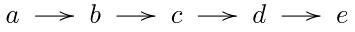
Example 2 (Four-cycle) *Consider the following four-cycle framework:*



There are two preferred extensions $\{a, c\}$ and $\{b, d\}$. For $\{a, c\}$, there are four different choices for the explanation extensions satisfying Properties UAIM, $\{a^a, c^a\}$, $\{a^c, c^c\}$, $\{a^a, c^c\}$, $\{a^c, c^a\}$, but only the latter also satisfies Property D.

Note that concerning Properties UAIDM, in contrast to Proposition 1, not all traditional Dung semantics are explainable. Consider the following counterexample.

Example 3 (Direct defense vs Minimality) *For the argumentation framework below, there is only one complete extension $\{a, c, e\}$, and only $E = \{a^{\{\}}, c^{\{a\}}, e^{\{c\}}\}$ satisfies Properties UAID, but E does not satisfy Property M, since $E' = \{a^{\{\}}, c^{\{\}}, e^{\{\}}\}$ is also a complete explanation extension.*



We therefore consider only UAIM in the following two propositions.

Proposition 2 *For all explanation semantics satisfying Properties UAIM, the label of each element of the grounded explanation extension is an empty set.*

Proof Assume that there is an element a^R such that R is not an empty set. Since $a^\emptyset$ satisfies Properties AIM, according to Properties U and M, it turns out that a^R is not an element of the explanation extension. Contradiction. □

Proposition 3 (Explainable semantics, Prop. UAIM) *All traditional Dung semantics are explainable with respect to Properties UAIM.*

Proof For all $\mathcal{E} \in \sigma(\mathcal{F})$, $\forall a \in \mathcal{E}$, since $\mathcal{E}$ defends a, there exist a set of sets $R_1, \ldots, R_n \subseteq \mathcal{E}$ where $n \geq 1$, such that $a \in c_\infty(R_i, F)$ where $i = 1, \ldots, n$. Let $R_a \in \{R_1, \ldots, R_n\}$ be a minimal set with respect to set inclusion. Let $E = \{a^{R_a} \mid a \in \mathcal{E}\}$. It holds that E satisfies Properties UAIM. □

The sixth property relates explanations by a kind of transitivity.

Property 6 (Transitivity) *For all argumentation frameworks F, all explanation extensions E of F, and all explained arguments $a^R, b^S \in E$, if $b \in R$, then $S \subseteq R$.*

Example 4 (Continue Example 2) *Among $\{a^a, c^c\}$, $\{a^a,\ c^a\}$, $\{a^c,\ c^c\}$ and $\{a^c, c^a\}$, only $\{a^c, c^a\}$ does not satisfy Property T, while others do.*

Transitivity together with the properties UAIM has as a surprising consequence that explanation arguments are themselves self-explanatory.

Proposition 4 (Self-explanation) *For all explanation semantics satisfying Properties UAIMT, if $a^R \in E$ and $b \in R$, then there exists $b^S \in E$; when S is a singleton, $b^b \in E$, i.e. b is self-explanatory.*

Proof According to Property A, b is in the corresponding extension $\{x \mid x^T \in E\}$ under Dung's argumentation semantics. So, there exists $S \subseteq \{x \mid x^T \in E\}$ such that $b^S \in E$. Then, according to Property T, $S \subseteq R$. Assume that b is not in S. Then, we may remove b from R to obtain $R' = R \setminus \{b\}$ and $a^{R'} \in E$. So, R is not minimal, contradicting Property M. Therefore, $b \in S$. When S is a singleton, $S = \{b\}$. So, $b^b \in E$. □

Note that in Proposition 4, when S is a not singleton, it might not hold that $b^b \in E$.

Example 5 *Consider the argumentation framework below. We have an explanation extension $E = \{b^{\{b,d\}}, d^d\}$. It holds that $b^{\{b,d\}} \in E$ and $b \in \{b, d\}$, but $b^b \notin E$.*

$$a \leftrightarrow b \leftarrow c \leftrightarrow d$$

Proposition 5 (Explainable semantics, Prop. UAIMT) *All traditional Dung semantics are explainable with respect to Properties UAIMT.*

Proof We need only to verify that Property T holds for each explanation extension, on the condition that Properties UAIM hold. According to the proof of Proposition 3, for all minimal sets $a^{R_a}, b^{R_b} \in E$, when $b \in R_a$, let $R'_a = (R_a \setminus \{b\}) \cup R_b$. Since we have $b \in c_\infty(R_b, F)$ and R_b is minimal, after replacing b with R_b, R'_a is minimal. So, there exists $E' = (E \setminus \{a^{R_a}\}) \cup \{a^{R'_a}\}$ such that $R_b \subseteq R'_a$. □

The following example further illustrates the idea in the above proof.

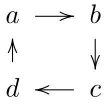
Example 6 *Continue Example 2, for $E = \{a^c, c^a\}$, let $R_a = \{c\}$ and $R_c = \{a\}$. Since $c \in R_a$, let $R'_a = (R_a \setminus \{c\}) \cup R_c = \{a\}$. Let $E' = (E \setminus \{a^{R_a}\}) \cup \{a^{R'_a}\} = \{a^a, c^a\}$. So, E' is an explanation extension satisfying Properties UAIMT.*

Let the defense set of $a^R \in E$ be the set $\{x^S \in E \mid \exists y : x \to y \to a\}$.

Property 7 (Explanation Inheritance) *For all $a^R \in E$ and $b \in R$, there is a c^S in the defense set of a^R such that $b \in S$.*

The following example illustrates property E.

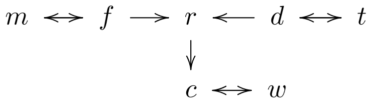
Example 7 *Consider again the four cycle framework:*



For $\{a, c\}$, the explanation extensions satisfying property UAIMTE are $\{a^a, c^a\}$, and $\{a^c, c^c\}$.

The following example is from Rienstra *et al.* [17].

Example 8 (Eating out) *The argumentation framework shown below represents the decision making of an agent planning to eat out.*



He will eat meat or fish (m or f) and take a taxi or drive himself (t or d). He drinks red wine (r) but not with fish or when driving (f and d attack r). Finally, he drinks either cola or water (c or w), but no cola if he drinks red wine (r attacks c).

The direction of the attacks implies that the agent first chooses independently between m and f and between t and d. Then he determines the status of r, which depends on f and d. Finally he chooses between c and w, which depends on r. Note that we can, of course, imagine different scenarios, but this would involve different directions of attack. E.g., if the decision about r came before *the decision between t and d, then the attack of d on r would be reversed.*

Now consider the preferred extenson $\{m, t, r, w\}$. The possible explanation extensions satisfying UAIMT are $\{m^m, t^t, r^m, w^m\}$, $\{m^m, t^t, r^t, w^t\}$, $\{m^m, t^t, r^m, w^t\}$, $\{m^m, t^t, r^t, w^m\}$, $\{m^m, t^t, r^m, w^r\}$, $\{m^m, t^t, r^t, w^r\}$, $\{m^m, t^t, r^m, w^w\}$, $\{m^m, t^t, r^t, w^w\}$. In other words, the explanation of r is either m or t, and the explanation of w is either m, t, r or w. Only the latter four satisfy property D.

Explanation $\{m^m, t^t, r^m, w^t\}$ and $\{m^m, t^t, r^t, w^m\}$ do not satisfy property E.

Proposition 6 *All traditional Dung semantics are explainable with respect to Properties UAIMTE.*

Proof We need only to verify that Property E holds on the conditions that Properties UAIMT hold. For an explanation extension E satisfying Properties UAIMT, and for all $a^R \in E$, since $a \in \{x \mid x^T \in E\}$, there exists $c, y \in A$ such that $c \to y \to a$ and $c \in \{x \mid x^T \in E\}$. So, there exists S' such that $c^{S'} \in E$. According to Proposition 5, given that $b \in R$, if $c = b \in R$, then $S' \subseteq R$. Then, according to the proof of Proposition 4, $b \in S'$. In this case, let $S = S'$, we have $b \in S$. Otherwise, $b \neq c$. Let $S = (S' \setminus c_\infty(\{b\}, F)) \cup \{b\}$. It holds that c^S satisfies Properties UAIMT. In this case, it holds that $b \in S$. □

Example 9 *Consider again the four-cycle framework: For $\{a, c\}$, $\{a^a, c^c\}$ satisfies Properties UAIM but not Property E, since $a \notin \{c\}$. Given that $c_\infty(\{a\}, F) = \{a, c\}$, let $S = (\{c\} \setminus \{a, c\}) \cup \{a\} = \{a\}$. As a result, we have $\{a^a, c^a\}$ as an explanation extension satisfying Property E.*

4. Examples of explanation semantics

Based on the principles introduced in the previous section, we may define various explanation semantics.

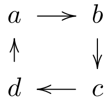
Definition 8 *Let $F = (A, \rightarrow)$ be an argumentation framework, and $X_F = \{a^R \mid a \in A, R \subseteq A\}$. For all $E \subseteq X_F$,*

- *E is conflict-free if and only if $\{a \mid a^R \in E\}$ is conflict-free.*
- *E is direct if and only if it is conflict-free and satisfies Properties UAID.*
- *E is a minimal explanation extension if and only if it is conflict-free and satisfies Properties UAIM.*
- *E is transitive if and only if it is a minimal explanation extension and satisfies Property* T.
- *E is explanation inherited if and only if it is transitive and satisfies Property* E.

Meanwhile, orthogonally, we say that E is complete (respectively, preferred, stable, and grounded), if and only if $\{x \mid x^R \in E\}$ is complete (respectively, preferred, stable, and grounded) under Dung's argumentation semantics.

The set of explanation extensions is represented as $\Sigma_\sigma(F)$, where $\Sigma \in \{\mathbf{D}, \mathbf{M}, \mathbf{T}, \mathbf{E}\}$, indicating direct, minimal, transitive and explanation inherited semantics, respectively, and σ is a Dung's semantics.

Example 10 *Consider again the four-cycle framework:*



- $\mathbf{M}_{pr}(F) = \{E_1, \ldots, E_8\}$, *where* $E_1 = \{a^a, c^a\}$, $E_2 = \{a^c, c^c\}$, $E_3 = \{a^a, c^c\}$, $E_4 = \{a^c, c^a\}$, $E_5 = \{b^b, d^b\}$, $E_6 = \{b^d, d^d\}$, $E_7 = \{b^b, d^d\}$, *and* $E_8 = \{b^d, d^b\}$.
- $\mathbf{D}_{pr}(F) = \{E_4, E_8\}$.
- $\mathbf{T}_{pr}(F) = \{E_1, E_2, E_3, E_5, E_6, E_7\}$.
- $\mathbf{E}_{pr}(F) = \{E_1, E_2, E_5, E_6\}$.

According to Definition 8, it is obvious that for all F, $\mathbf{E}_\sigma(F) \subseteq \mathbf{T}_\sigma(F) \subseteq \mathbf{M}_\sigma(F)$.

Meanwhile, according to Example 10, it seems that for all F, $\mathbf{D}_\sigma(F) \subseteq \mathbf{M}_\sigma(F)$. Unfortunately, this is not the case in general: remember that in Example 3, $\mathbf{D}_\sigma(F) = \{E\}$, $\mathbf{M}_\sigma(F) = \{E'\}$, $E \neq E'$, and therefore $\mathbf{D}_\sigma(F) \not\subseteq \mathbf{M}_\sigma(F)$.

5. Explanation based on weak defense graphs

In this section, we formulate two examples of explanation semantics based on a kind of meta-argumentation framework, of which the nodes are no longer arguments, but pairs of arguments, reflecting a weak notion of defense.

Before we formally introduce this meta-argumentation theory, we introduce this weak notion of defense, which we call defense graphs. We start by making two observations concerning the role of defense in Dung's theory. The first observation is that the notion of defense by itself is too weak to capture all relevant properties of an argumentation framework. For example, an argumentation framework with three arguments, each attacking the next one in the sequence $a \longrightarrow b \longrightarrow c$ has a defense graph where a defends c, but nothing is said about b. If we represent the defense relation by a double arrow, then the defense graph may be visualized by $a \Longrightarrow c \quad b$

We cannot take this defense graph as the basis for formal argumentation, because it is no longer clear whether argument b can be accepted or not. Thus, a defense graph represents some information about argumentation frameworks, but not everything.

The second observation concerning the notion of defense in formal argumentation is that it is not a binary relation over arguments, like the attack relation is a binary relation over arguments, but it is a relation between a set of arguments and an argument. In this sense, the defense relation is different from the so-called support relation, which is often studied in abstract argumentation.

The following definition of defense graph deals with these two issues in the following way. First, a defense graph is defined relatively to an argumentation framework. Thus, it is not meant to replace the attack relation, but it is used in addition to it. Also, we consider arguments defended by the empty set, i.e. arguments which are not attacked (called initial arguments in graph theory). Second, whereas a set of arguments S defends argument b when it attacks *all* attackers of b, we say that a defends b when a attacks *some* attacker of b. In defense graphs, we are thus slightly abusing the word "defense" for a similar but distinct notion. We could distinguish the two notions by writing $\forall$defends and $\exists$defends, but since the difference is always clear from context, we prefer to overload the concept of defense.

Definition 9 (Weak defense) *Let $F = (A, \rightarrow)$ be an argumentation framework. For $a, b \in A$,*

- *$\langle a, b\rangle$ is a weak defense if and only if $\exists c \in A$ such that $a \rightarrow c$ and $c \rightarrow b$.*
- *$\langle \emptyset, b\rangle$ is a weak defense iff b is initial.*

The set of weak defenses of F is denoted as DEF_F. Given a weak defense $\langle a, b\rangle$ or $\langle \emptyset, b\rangle \in \mathrm{DEF}_F$, we call a the *defender*, and b the *defendee*, of the defense. Given a set $D \subseteq \mathrm{DEF}_F$, we write $\mathrm{defendee}(D) = \{b \mid \langle a, b\rangle, \langle \emptyset, b\rangle \in D\}$ to denote the set of defendees in D, $\mathrm{defender}(D) = \{a \mid \langle a, b\rangle \in D\}$ to denote the set of defenders in D, and $\arg(D) = \mathrm{defendee}(D) \cup \mathrm{defender}(D)$ be the set of arguments who are defendees and defenders in D.

We now define the attacks of the meta-argumentation framework, which are attacks between weak defenses.

Definition 10 (Attacks between weak defenses) *For all $\langle x, a\rangle$, $\langle y, b\rangle \in \mathrm{DEF}_F$ where $x, y \in A \cup \{\emptyset\}$ and $a, b \in A$, we say that $\langle x, a\rangle$ attacks $\langle y, b\rangle$, denoted as $\langle x, a\rangle \rightarrow \langle y, b\rangle$ iff $x \rightarrow y$, or $x \rightarrow b$, or $a \rightarrow y$, or $a \rightarrow b$.*

The set of attacks between weak defenses and their defeaters is denoted as $\rightarrow_F$. We call $\mathrm{DG}_F = (\mathrm{DEF}_F, \rightarrow_F)$ a defense graph. Given an extension $\mathcal{E}$ of F under Dung's argumentation semantics, let $\mathrm{defense}(\mathcal{E}) = \{\langle x, y\rangle \in \mathrm{DEF}_F \mid x \in \mathcal{E} \cup \{\emptyset\}, y \in \mathcal{E}\}$.

We have the following proposition, corresponding to Theorems 1, 2 and Corollaries 1, 2 in [13] with slightly modified notations.

Proposition 7 *Given $F = (A, \rightarrow)$ and its defense graph $\mathrm{DG}_F = (\mathrm{DEF}_F, \rightarrow_F)$, it holds that $\forall D \in \sigma(\mathrm{DG}_F)$, $\arg(D) \in \sigma(F)$; and $\forall \mathcal{E} \in \sigma(F)$, $\mathrm{defense}(\mathcal{E}) \in \sigma(\mathrm{DG}_F)$.*

Definition 11 *Given $F = (A, \rightarrow)$ and its defense graph $\mathrm{DG}_F = (\mathrm{DEF}_F, \rightarrow_F)$, $\forall D \in \sigma(\mathrm{DG}_F)$, let $E = \{a^{R_a} \mid \langle x, a\rangle \in D\}$ where $R_a = \{b \mid \langle b, a\rangle \in D\} \setminus \{\emptyset\}$. We call E a Direct explanation extension. The set of Direct explanation extensions is denoted $\mathrm{Direct}(F)$.*

Proposition 8 *Direct explanation semantics satisfies Properties UAID.*

Proof According to Definition 11, Properties UAID hold by definition. □

In this paper, we view a defense as a transitive relation, i.e., if $\langle a, b\rangle$ and $\langle b, c\rangle$ then $\langle a, c\rangle$. Based on this notion, we have the following definition.

Definition 12 *Given $F = (A, \rightarrow)$ and its defense graph $\mathrm{DG}_F = (\mathrm{DEF}_F, \rightarrow_F)$, $\forall D \in \sigma(\mathrm{DG}_F)$, let D^* be the transitive closure of D. let $E = \{a^{R_a} \mid \langle x, a\rangle \in D\}$ where $R_a = \{a \mid \langle a, a\rangle \in D^*\} \cup \{b \mid \langle b, a\rangle \in D^*, \langle b, b\rangle \in D^*\}$. We call E a Root explanation extension. The set of Root explanation extensions is denoted $\mathrm{Root}(F)$.*

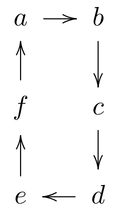
Proposition 9 *Root explanation semantics satisfies Properties UAITE.*

Proof First, since for each $a^{R_a} \in E$, R_a is unique, Property Uniqueness is satisfied. Second, according to Proposition 7, it holds that if $\langle a, b\rangle \in D$ then there exists $\langle c, a\rangle \in D$. So, in terms of Definition 12, $R_a \subseteq \arg(D)$ and $\{b \in E \mid b^{R_b}\} = \arg(D)$. So, $R_a \subseteq \{b \in E \mid b^{R_b}\}$, and Property Acceptance is satisfied. Third, according to Definition 12, $a \in c_\infty(R_a, F)$, and therefore Properties Indirect Defense hold. Fourth, for all $a^{R_a}, b^{R_b} \in E$, if $b \in R_a$, assume that $R_b \not\subseteq R_a$. Then, exists $c \in R_b$ such that $c \notin R_a$. So, $\langle c, a\rangle \notin D^*$. Since when $b \neq a \neq c$, $\langle b, a\rangle \in D^*$ and $\langle c, b\rangle \in D^*$. As a result, $\langle c, a\rangle \in D^*$. Contradiction. So, Property Transitivity holds. Fifth, if $a = b$, then let $c^S = a^{R_a}$. In this case, Property Explanation Inheritance holds. Otherwise, $a \neq b$. In this case, $\langle b, a\rangle \in D^*$ and $\langle b, b\rangle \in D^*$. Let $c^S = b^{R_b}$ where $b \in R_b$. Property Explanation Inheritance also holds. □

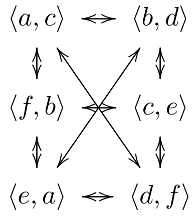
Note that Root explanation semantics does not satisfy Properties Direct defense and Minimality, as illustrated by the following examples.

Example 11 *Given F_1 and DG_{F_1} below, under preferred semantics, there are two extensions of DG_{F_1}: $D_1 = \{\langle a, c\rangle, \langle c, e\rangle, \langle e, a\rangle\}$, $D_2 = \{\langle b, d\rangle, \langle d, f\rangle, \langle f, b\rangle\}$. So, $D_1^* = D_1 \cup \{\langle a, e\rangle, \langle a, a\rangle, \langle c, a\rangle, \langle c, c\rangle, \langle e, c\rangle, \langle e, e\rangle\}$ and $D_2^* = D_2 \cup \{ \langle b, f\rangle, \langle b, b\rangle, \langle d, b\rangle, \langle d, d\rangle, \langle f, d\rangle, \langle f, f\rangle\}$. So, we have two Root explanation extensions: $E_1 = \{a^{\{a,c,e\}}, c^{\{a,c,e\}}, c^{\{a,c,e\}}\}$, and $E_2 = \{b^{\{b,f,d\}}, f^{\{b,f,d\}}, d^{\{b,f,d\}}\}$, which do not satisfy Property Minimality.*

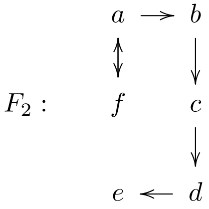
F_1 :

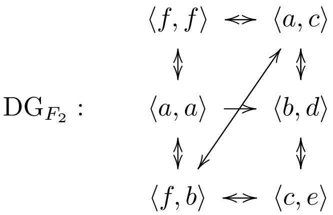


DG_{F_1} :



Example 12 *Given F_2 and DG_{F_2} below, under preferred semantics, there are two extensions of DG_{F_1}: $D_1 = \{\langle f, f\rangle, \langle f, b\rangle, \langle b, d\rangle\}$, $D_2 = \{\langle a, a\rangle, \langle a, c\rangle, \langle c, e\rangle\}$. So, $D_1^* = D_1 \cup \{\langle f, d\rangle\}$ and $D_2^* = D_2 \cup \{\ \langle a, e\rangle\}$. So, we have two Root explanation extensions: $E_1 = \{f^{\{f\}}, b^{\{f\}}, d^{\{f\}}\}$, and $E_2 = \{a^{\{a\}}, c^{\{a\}}, e^{\{a\}}\}$, which do not satisfy Property Direct defense.*





6. Conclusions and future work

We study explanation semantics of argumentation by using a principle-based approach. More specifically, in this paper we introduce the explanation principles Uniqueness, Acceptance, Indirect defense, Direct defense, Minimality, Transitivity, Explanation Inheritance. Furthermore, we define various examples of explanations of traditional abstract argumentation semantics. In further work, the formal approach in this paper needs to be extended to informal argumentation as well [3,7].

The work in this paper can be further developed in many ways for both the principles and the explanation semantics. For example, instead of only explaining why an argument is accepted, we can also explain why it is rejected. Explanations can be restricted to core arguments or to representations [14]. Explanation semantics can be combined with, for example, labeling semantics and ranking semantics can be used to rank explanations as well. Moreover, support relations or numerical arguments or attacks can be used to define more sophisticated notions of explanation. The abstract theory of explanation can be further developed for structured argumentation. For example, explanation arguments can refer to evidence or to ethical or legal principles. We believe that such a study of explanation in structured argumentation can also inspire new theories of explanation in abstract argumentation.

More concepts from the general theory of explanation [15] can be studied in formal argumentation, and a general theory of explanation for abstract argumentation can be developed, combining explanation semantics with other notions of explanation in formal argumentation, for example in dialogue [4]. A striking similarity between both is that the notion of defense plays a central role, and such a unified theory of argumentation explanation may lead to a more formal argumentation in which attack and defense are at par. This may also bring the theory of formal argumentation closer to theories of attack and defense in other disciplines such as security [12,10] and in biology [16,2].

Acknowledgement

This material is based in part upon work supported by the “2030 Megaproject” - New Generation Artificial Intelligence of China under Grant No. 2018AAA0100904, the Natural Science Foundation of Zhejiang Province under Grant No. LY20F030014, and the National Social Science Foundation of China No.18ZDA290 and No.17ZDA026.

References

[1] Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors. *Handbook of formal argumentation*, volume 1. College Publications, 2018.

[2] Howard Barringer, Dov M. Gabbay, and John Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, pages 59–98, 2005.

[3] Marcos Cramer and Mathieu Guillaume. Empirical study on human evaluation of complex argumentation frameworks. In *JELIA 2019*, pages 102–115, 2019.

[4] Kristijonas Cyras, David Birch, Yike Guo, Francesca Toni, Rajvinder Dulay, Sally Turvey, Daniel Greenberg, and Tharindi Hapuarachchi. Explanations by arbitrated argumentative dispute. *Expert Syst. Appl.*, 127:141–156, 2019.

[5] Kristijonas Cyras, Ken Satoh, and Francesca Toni. Explanation for case-based reasoning via abstract argumentation. In *COMMA 2016*, pages 243–254, 2016.

[6] Jérémie Dauphin, Marcos Cramer, and Leendert W. N. van der Torre. Abstract and concrete decision graphs for choosing extensions of argumentation frameworks. In *COMMA 2018*, pages 437–444, 2018.

[7] Jérôme Delobelle and Serena Villata. Interpretability of gradual semantics in abstract argumentation. In *ECSQARU 2019*, pages 27–38, 2019.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[9] Xiuyi Fan and Francesca Toni. On explanations for non-acceptable arguments. In *TAFA 2015*, pages 112–127, 2015.

[10] Dov Gabbay, Ross Horne, Sjouke Mauw, and Leendert van der Torre. Argumentation-based semantics for attack-defense networks. In *Proceedings of The Seventh International Workshop on Graphical Models for Security (GRAMSEC2020)*, 2020.

[11] Alejandro Javier García, Nicolás D. Rotstein, and Guillermo Ricardo Simari. Dialectical explanations in defeasible argumentation. In *ECSQARU 2007*, pages 295–307, 2007.

[12] Barbara Kordy, Sjouke Mauw, Sasa Radomirovic, and Patrick Schweitzer. Foundations of attack-defense trees. In Pierpaolo Degano, Sandro Etalle, and Joshua D. Guttman, editors, *FAST 2010*, volume 6561 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 2010.

[13] Beishui Liao and Leendert W. N. van der Torre. Defense semantics of argumentation: encoding reasons for accepting arguments. *CoRR*, abs/1705.00303, 2017.

[14] Beishui Liao and Leendert W. N. van der Torre. Representation equivalences among argumentation frameworks. In *COMMA 2018*, pages 21–28, 2018.

[15] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[16] Sergio M. Pellis and Vivien C. Pellis. Differential rates of attack, defense, and counterattack during the developmental decrease in play fighting by male and female rats. *Developmental Psychobiology*, 23, 1990.

[17] Tjitze Rienstra, Matthias Thimm, Beishui Liao, and Leendert W. N. van der Torre. Probabilistic abstract argumentation based on SCC decomposability. In *KR 2018*, pages 168–177, 2018.

[18] Dunja Seselja and Christian Straßer. Abstract argumentation and explanation applied to scientific debates. *Synthese*, 190(12):2195–2217, 2013.

[19] L. Richard Ye and Paul E. Johnson. The impact of explanation facilities in user acceptance of expert system advice. *MIS Quarterly*, 19(2):157–172, 1995.

[20] Zhiwei Zeng, Chunyan Miao, Cyril Leung, Zhiqi Shen, and Jing Jih Chin. Computing argumentative explanations in bipolar argumentation frameworks. In *AAAI 2019*, pages 10079–10080, 2019.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200512

Possible Controllability of Control Argumentation Frameworks

Jean-Guy MAILLY
LIPADE, Université de Paris, jean-guy.mailly@u-paris.fr

Abstract. The recent Control Argumentation Framework (CAF) is a generalization of Dung's Argumentation Framework which handles argumentation dynamics under uncertainty; especially it can be used to model the behavior of an agent which can anticipate future changes in the environment. Here we provide new insights on this model by defining the notion of possible controllability of a CAF. We study the complexity of this new form of reasoning for the four classical semantics, and we provide a logical encoding for reasoning with this framework.

Keywords. Abstract argumentation, uncertainty, computational complexity

1. Introduction

Abstract argumentation [1] has become an important subfield of Knowledge Representation and Reasoning research in the last decades. Intuitively, an abstract argumentation framework (AF) is a directed graph where nodes are arguments and edges are relations (usually attacks) between these arguments. The outcome of such an AF is an evaluation of the arguments' acceptance (through extensions [1,2], labellings [3] or rankings [4]). The question of argumentation dynamics has arisen more recently, and many different approaches have been proposed (see *e.g.* [5,6,7,8,9,10,11,12,13,14]). Roughly speaking, the question of these works is "how to modify an AF to be consistent with a given piece of information?". Such a piece of information can be "argument a should be accepted in the outcome of the AF". A particular version of this problem is called *extension enforcement* [7,15,10,12,13]: it consists in modifying an AF s.t. a given set of arguments becomes (included in) an extension of the AF. The recently proposed *Control Argumentation Framework* (CAF) [14] is a generalization of Dung's AF which incorporates different notions of uncertainty in the structure of the framework. The *controllability* of a CAF w.r.t. a set of arguments is the fact that, whatever happens in the uncertain part of the CAF (*i.e.* whatever is the real situation of the world), the target set of arguments is accepted. This is somehow a generalization of extension enforcement, where uncertainty is taken into account.

In this paper, we study what we call *possible controllability* (and then, controllability defined in [14] can be renamed as *necessary controllability*). The idea of possible controllability w.r.t. a target set of arguments is that this target should be accepted in *at least* one of the possible completions of the uncertain part. Necessary controllability trivially implies possible controllability, while the converse is not true. This form of reasoning can be applied in different situations. Possible controllability makes sense in situations where an agent is unable to guarantee some result (the fact that some argument a is accepted), but she wants to be sure that the opposite result (a is rejected) is not necessary true. For instance, possible controllability is similar to the reasoning of the defendant's lawyer during a trial. Thanks to the principle of *presumption of innocence*, the lawyer does not have to prove that the defendant *is* innocent, but he has to prove that the defendant *may be* innocent. This means that if there is some uncertainty in the case, the lawyer wants to exhibit the fact that one possible world encompassed by this uncertainty implies that his client is innocent.[1] This means that the lawyer's knowledge about the case can be represented by a CAF, and the lawyer wants to guarantee that the argument "the defendant is innocent" is accepted in at least one completion of the CAF, *i.e.* one possible world. In this kind of scenario, possible controllability is particularly useful since it is (presumably) easier to search for one completion that accepts the target instead of checking that the target is accepted in each of the (exponentially many) completions.

The paper is organized as follows. We first recall the background notions of logic and introduce the CAF setting in Section 2. In Section 3 we define formally this new form of controllability, and we determine the complexity of this reasoning problem for the four classical semantics introduced by Dung. We also propose a QBF-based encoding which allows to determine whether a CAF is possible controllable w.r.t. a target and the stable semantics (and moreover, which allows to determine *how* to control it). We describe the related work in Section 4, and finally Section 5 concludes the paper and draws interesting future research tracks.

2. Background

2.1. Propositional Logic and Quantified Boolean Formulas

We consider a set V of Boolean variables, *i.e.* variables which can be assigned a value in $\mathbb{B} = \{0,1\}$, where 0 and 1 are associated respectively with *false* and *true*. Such variables can be combined with connectives $\{\vee, \wedge, \neg\}$ to build formulas. $x \vee y$ is true if at least one of the variables x, y is true; $x \wedge y$ is true if both x, y are true; $\neg x$ is true is x is false. Additional connectives can be defined, *e.g.* $x \Rightarrow y$ is equivalent to $\neg x \vee y$; $x \Leftrightarrow y$ is equivalent to $(x \Rightarrow y) \wedge (y \Rightarrow x)$. The definition of the connectives is straightforwardly extended from variables to formulas (*e.g.* if ϕ and ψ are formulas, then $\phi \wedge \psi$ is true when both formulas are true). A truth assignment on the set of variables $V = \{x_1, \dots, x_n\}$ is a mapping $\omega : V \to \mathbb{B}$.

Quantified Boolean Formulas (QBFs) are an extension of propositional formulas with the universal and existential quantifiers. For instance, the formula $\exists x \forall y (x \vee \neg y) \wedge (\neg x \vee y)$ is satisfied if there is a value for x such that for all values of y the proposition $(x \vee \neg y) \wedge (\neg x \vee y)$ is true. More formally, a canonical QBF is a formula $\mathcal{Q}_1 X_1 \mathcal{Q}_2 X_2 \dots \mathcal{Q}_n X_n \Phi$ where Φ is a propositional formula, $\mathcal{Q}_i \in \{\exists, \forall\}$, $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$, and $X_1, X_2, \dots, X_n$ disjoint sets of propositional variables such that $X_1 \cup X_2 \cup \dots \cup X_n = V$.[2] It is well-known that QBFs span the polynomial hierarchy. For instance, deciding whether the formula $\exists X_1 \forall X_2 \dots \mathcal{Q}_i X_i \Phi$ is true is Σ_i^p-complete. The decision problem associated

[1] On the opposite, necessary controllability [14] is close to the reasoning of the prosecutor.

[2] If some variable $x \in V$ does not explicitly belong to any X_i, *i.e.* $X_1 \cup \dots \cup X_n \subset V$, then it implicitly means that x can be existentially quantified at the rightmost level.

to QBFs of the form $\exists V, \Phi$ is equivalent to the satisfiability problem for propositional formulas (SAT), which is well-known to be NP-complete. For more details about propositional logic, QBFs and complexity theory, we refer the reader to [16,17,18].

2.2. Abstract Argumentation and Control Argumentation Frameworks

An *argumentation framework* (AF), introduced in [1], is a directed graph $\mathcal{AF} = \langle A, R \rangle$, where A is a set of *arguments*, and $R \subseteq A \times A$ is an *attack relation*. The relation *a attacks b* is denoted by $(a,b) \in R$. In this setting, we are not interested in the origin of arguments and attacks, nor in their internal structure. Only their relations are important to define the acceptance of arguments.

In [1], different acceptability semantics were introduced. They are based on two basic concepts: *conflict-freeness* and *defence*. A set $S \subseteq A$ is:

- conflict-free iff $\forall a, b \in S$, $(a,b) \notin R$;
- admissible iff it is conflict-free, and defends each $a \in S$ against its attackers.

The semantics defined by Dung are as follows. An admissible set $S \subseteq A$ is:

- a complete extension iff it contains every argument that it defends;
- a preferred extension iff it is a $\subseteq$-maximal complete extension;
- the unique grounded extension iff it is the $\subseteq$-minimal complete extension;
- a stable extension iff it attacks every argument in $A \setminus S$.

The sets of extensions of an $\mathcal{AF}$, for these four semantics, are denoted (respectively) $\mathrm{co}(\mathcal{AF})$, $\mathrm{pr}(\mathcal{AF})$, $\mathrm{gr}(\mathcal{AF})$ and $\mathrm{st}(\mathcal{AF})$.

Our approach could be adapted for any other extension semantics. Based on these semantics, we can define the status of any (set of) argument(s), namely *skeptically accepted* (belonging to each σ-extension), *credulously accepted* (belonging to some σ-extension) and *rejected* (belonging to no σ-extension). For more details about argumentation semantics, we refer the reader to [1,2].

We introduce now the notions of CAF and (necessary) controllability [14].

Definition 1. *A Control Argumentation Framework (*CAF*) is a triple $\mathcal{CAF} = \langle \mathcal{F}, \mathcal{C}, \mathcal{U} \rangle$ where $\mathcal{F}$ is the* fixed part, *$\mathcal{U}$ is the* uncertain part *and $\mathcal{C}$ is the* control part *of $\mathcal{CAF}$ with:*

- *$\mathcal{F} = \langle A_F, \rightarrow \rangle$ where A_F is a set of arguments and $\rightarrow \subseteq (A_F \cup A_U) \times (A_F \cup A_U)$ is an attack relation.*
- *$\mathcal{U} = \langle A_U, (\rightleftarrows \cup \dashrightarrow) \rangle$ where A_U is a set of arguments, $\rightleftarrows \subseteq (((A_U \cup A_F) \times (A_U \cup A_F)) \setminus \rightarrow)$ is a conflict relation and $\dashrightarrow \subseteq (((A_U \cup A_F) \times (A_U \cup A_F)) \setminus \rightarrow)$ is an attack relation, with $\rightleftarrows \cap \dashrightarrow = \emptyset$.*
- *$\mathcal{C} = \langle A_C, \Rrightarrow \rangle$ where A_C is a set of arguments, and $\Rrightarrow \subseteq \{(a_i, a_j) \mid a_i \in A_C,\ a_j \in A_F \cup A_C \cup A_U\}$ is an attack relation.*

A_F, A_U and A_C are disjoint subsets of arguments.

The different sets of arguments and attacks have different meanings. The fixed part $\mathcal{F}$ represents the part of the system which cannot be influenced either by the agent or by the environment. This means that if $a \in A_F$, then it is sure that a is an "active" argument (for instance, all of its premises are true, and cannot be falsified). Similarly, if $(a,b) \in \rightarrow$, the attack from a to b is actually part of the system and cannot be removed.

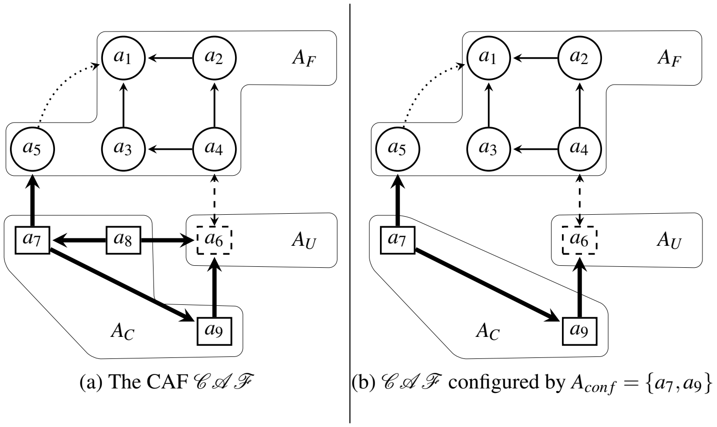
$\mathscr{U}$ is the uncertain part of the system. This means that it cannot be influenced by the agent, but it can be modified by the environment (in a wide way, this can also represent the possible actions of other agents). The uncertainty can appear in different ways. First, if $a \in A_U$, this means that there is some uncertainty about the presence of an argument (for instance, the agent is not sure whether her opponent in the debate will state argument a, or she is not sure whether the premises of a will be true at some moment). If $(a,b) \in \rightleftarrows$, then the agent is sure that there is a conflict between a and b, but she is not sure of the direction of the attack (this could be an attack (a,b), an attack (b,a), or even both at the same time). This is possible, for instance, if the agent is not sure about some preference between a and b [19]. Finally, $(a,b) \in \dashrightarrow$ means that the agent is not sure whether there is actually an attack from a to b.

The last part $\mathscr{C}$ is the *control* part. This is the part of the system which can be influenced by the agent. This means that the agent has to choose which arguments she will actually use (uttering them in the debate, or making an action to switch their premises to true). When the agent uses a subset $A_{conf} \subseteq A_C$, called a *configuration*, this defines a configured CAF where the arguments from $A_C \setminus A_{conf}$ (and the attacks concerning them) are removed. We illustrate these concepts on an example adapted from [14].

Example 1. *We define $\mathscr{CAF} = \langle F,C,U \rangle$ as follows:*

- $\mathscr{F} = \langle \{a_1,a_2,a_3,a_4,a_5\}, \{(a_2,a_1),(a_3,a_1),(a_4,a_2),(a_4,a_3)\} \rangle$*;*
- $\mathscr{U} = \langle \{a_6\}, \rightleftarrows \cup \dashrightarrow \rangle$*, with* $\rightleftarrows = \{(a_6,a_4)\}$*, and* $\dashrightarrow = \{(a_5,a_1)\}$*;*
- $\mathscr{C} = \langle \{a_7,a_8,a_9\}, \{(a_7,a_5),(a_7,a_9),(a_8,a_6),(a_8,a_7),(a_9,a_6)\} \rangle$.

$\mathscr{CAF}$ is given at Figure 1a. The configuration of $\mathscr{CAF}$ by $A_{conf} = \{a_7,a_9\}$ yields the configured CAF $\mathscr{CAF}'$ described at Figure 1b. On the figures, arguments from A_F, A_U and A_C are respectively represented as circle nodes, dashed square nodes and plain square nodes. Similarly, the attacks from $\rightarrow$, $\rightleftarrows$, $\dashrightarrow$ and $\Rrightarrow$ are represented (respectively) as plain, double-headed dashed, dotted and bold arrows.



(a) The CAF $\mathscr{CAF}$

(b) $\mathscr{CAF}$ configured by $A_{conf} = \{a_7,a_9\}$

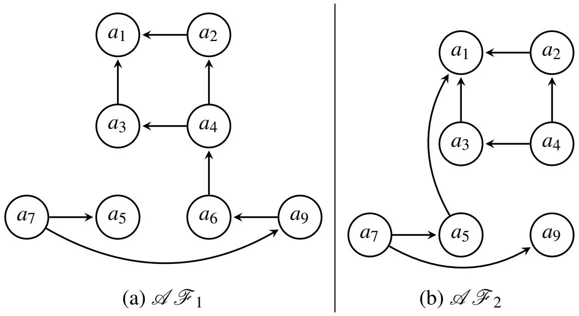
Figure 1. A CAF and a configured CAF

Now we recall the notion of completion, borrowed from [20], and adapted to CAFs in [14]. Intuitively, a completion is a classical AF which describes a situation of the world coherent with the uncertain information encoded in the CAF.

Definition 2. *Given $\mathcal{CAF} = \langle F, C, U \rangle$, a completion of $\mathcal{CAF}$ is $\mathcal{AF} = \langle A, R \rangle$, s.t.*

- $A = A_F \cup A_C \cup A_{comp}$ *where* $A_{comp} \subseteq A_U$;
- *if* $(a,b) \in R$, *then* $(a,b) \in \rightarrow \cup \rightleftarrows \cup \dashrightarrow \cup \Rightarrow$;
- *if* $(a,b) \in \rightarrow$, *then* $(a,b) \in R$;
- *if* $(a,b) \in \rightleftarrows$ *and* $a,b \in A$, *then* $(a,b) \in R$ *or* $(b,a) \in R$;
- *if* $(a,b) \in \Rightarrow$ *and* $a,b \in A$, *then* $(a,b) \in R$.

Example 2 (Continuation of Example 1). *We describe two possible completions of $\mathcal{CAF}'$. First, we consider a completion $\mathcal{AF}_1$ where the attack (a_5, a_1) is not included, while the argument a_6 (with the attack (a_6, a_4)) is included. Another possible completion is $\mathcal{AF}_2$, where a_6 is not included (so, neither the attacks related to it) while the attack (a_5, a_1) is included.*



(a) $\mathcal{AF}_1$

(b) $\mathcal{AF}_2$

Figure 2. Two possible completions of $\mathcal{CAF}'$

Now, a CAF is necessary controllable w.r.t. a target $T \subseteq A_F$ if the agent can configure it in a way which guarantees that T is accepted in every completion of the configured CAF. This necessary controllability has two versions, depending on the kind of acceptance under consideration (skeptical or credulous).

Definition 3. *Given a set of arguments $T \subseteq A_F$ and a semantics σ, $\mathcal{CAF}$ is* necessary skeptically (resp. credulously) controllable *w.r.t. T and σ iff $\exists A_{conf} \subseteq A_C$ s.t. T is included in each (resp. some) σ-extension of each completion of $\mathcal{CAF}' = \langle F, C', U \rangle$, with $C' = \langle A_{conf}, \{(a_i, a_j) \in \Rightarrow \mid a_i, a_j \in (A_F \cup A_U \cup A_{conf})\} \rangle$.*

[14] proposes a QBF-based method to determine whether a CAF is necessary controllable, and to obtain the corresponding configuration if it exists.

3. Possible Controllability

3.1. Formal Definition of Possible Controllability

The intuition of necessary controllability is that the agent is satisfied when its target is reached in every possible world encoded by the uncertain information in the CAF. While this is an interesting property (especially for applications like negotiation [21]), this may seem unrealistic for some applications, where the graph is built in such a way that some completions cannot accept the target. Here, we adapt the definition of controllability to consider that the agent is satisfied whether there exists at least one possible world (*i.e.* one completion) which accepts the target.

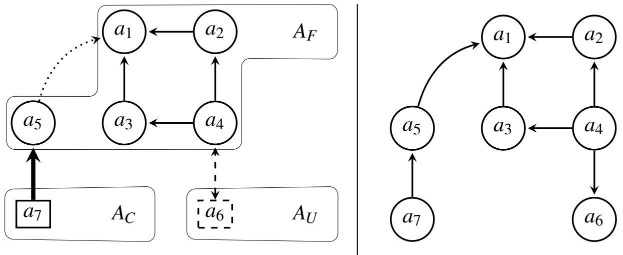
Definition 4. *Given a set of arguments* $T \subseteq A_F$ *and a semantics* σ*,* $\mathcal{CAF}$ *is* possibly skeptically (resp. credulously) controllable *w.r.t.* T *and* σ *iff* $\exists A_{conf} \subseteq A_C$ *s.t.* T *is included in each (resp. some)* σ*-extension of some completion of* $\mathcal{CAF}' = \langle F, C', U \rangle$*, with* $C' = \langle A_{conf}, \{(a_i, a_j) \in \Rightarrow \mid a_i, a_j \in (A_F \cup A_U \cup A_{conf})\} \rangle$*.*

Observation 1. Given a set of arguments $T \subseteq A_F$ and a semantics σ, if $\mathcal{CAF}$ is necessary skeptically (resp. credulously) controllable w.r.t. T and σ, then $\mathcal{CAF}$ is possibly skeptically (resp. credulously) controllable w.r.t. T and σ. The converse is false.

Example 3 (Continuation of Example 1)**.** *We observe that* $\mathcal{CAF}$ *from the previous example is not necessary skeptically controllable w.r.t. the target* $\{a_1\}$*. Indeed,*

- *if* $A_{conf} = \{a_7, a_8, a_9\}$*, then because of the attack* (a_8, a_7)*, the target is not defended against the potential threat* $(a_5, a_1) \in \dashrightarrow$*. The same thing happens if* $A_{conf} = \{a_7, a_8\}$ *or* $A_{conf} = \{a_8, a_9\}$*.*
- *if* $A_{conf} = \{a_7, a_9\}$*, this time the target is not defended against the potential threat coming from* a_6 *(in the completions where* a_6 *belongs to the system, along with the attack* (a_6, a_4)*,* a_1 *is not accepted).*
- *if* A_{conf} *is one of the three possible singletons, then again* a_1 *is not accepted in every completion (since either* a_5 *or* a_6 *is unattacked).*

On the opposite, it is possible to configure $\mathcal{CAF}$ *is such a way that* a_1 *is skeptically accepted in at least one completion. For instance, Figure 3a describes such a configured CAF, with a successful completion given at Figure 3b.*



(a) $\mathcal{CAF}$ configured by $A_{conf} = \{a_7\}$

(b) A successful completion of the CAF

Figure 3. A configured CAF and a successful completion

3.2. Computational Complexity of Possible Controllability

Now we focus on the computational complexity of deciding whether a CAF is possibly controllable. Formally, for $x \in \{sk, cr\}$ standing respectively for "skeptically" and "credulously", and $\sigma \in \{\text{co}, \text{pr}, \text{gr}, \text{st}\}$, we study the decision problem:

$\mathsf{Control}_{\sigma,p,x}^{\mathscr{CAF},T}$ Is the CAF $\mathscr{CAF}$ possibly x-controllable w.r.t. σ and T?

Proposition 1. *The complexity of* $\mathsf{Control}_{\sigma,w,x}^{\mathscr{CAF},T}$*, for* $x \in \{sk, cr\}$ *and* $\sigma \in \{\mathrm{co}, \mathrm{pr}, \mathrm{gr}, \mathrm{st}\}$*, is given at Table 1.*

σ	sk	cr
st	Σ_2^P-complete	NP-complete
co	NP-complete	NP-complete
gr	NP-complete	NP-complete
pr	Σ_3^P-complete	NP-complete

Table 1. The complexity of $\mathsf{Control}_{\sigma,p,x}^{\mathscr{CAF},T}$, for $x \in \{sk, cr\}$

Detailled proofs are omitted for space reasons. However, we can explain lower bounds from existing results. In [22], the decision problems σ-PSA (possible skeptical acceptance) and σ-PCA (possible credulous acceptance) for Incomplete Argumentation Frameworks (IAFs) have been studied. A IAF corresponds to a CAF where $\rightleftarrows = \emptyset$ and $A_C = \emptyset$ (and obviously, $\Rrightarrow$ is empty too). Thus, an argument a is skeptically (resp. credulously) accepted in some completion of the IAF iff the corresponding CAF is skeptically (resp. credulously) controllable w.r.t. the target $\{a\}$. This means that if σ-PSA (resp. σ-PCA) is C-hard (for some class C of the polynomial hierarchy), then $\mathsf{Control}_{st,p,sk}^{\mathscr{CAF},T}$ (resp. $\mathsf{Control}_{st,p,cr}^{\mathscr{CAF},T}$) is C-hard as well.

For upper bounds, we obtain some of them from the known complexity of skeptical or credulous acceptance in Dung's AFs [23]. Indeed, a completion that skeptically (resp. credulously) accepts the target is a witness that the CAF is possibly skeptically (resp. credulously) controllable w.r.t. the target. This leads to the upper bounds for possible skeptical controllability, as well as the possible credulous controllability under grounded semantics. The possible credulous controllability for the other semantics can be reduced to SAT, so they belong to NP (the method is given in details for stable semantics in the next part of the paper).

Let us also briefly discuss the complexity of possible controllability for simplified CAFs, defined by [14] as CAFs with no uncertainty (*i.e.* $A_U = \rightleftarrows = \dashrightarrow = \emptyset$). Such a CAF has only one completion for each control configuration, thus possible and necessary controllability are equivalent in this case, and complexity remains the same as in the general case, described at Table 1.

3.3. Possible Controllability Through QBFs

Inspired by [14], we propose a QBF-based method to compute possible controllability for the stable semantics. Let us first give the meaning of the propositional variables used in the encoding.

Given $\mathscr{AF} = \langle A, R \rangle$,

- $\forall x_i \in A$, acc_{x_i} represents the acceptance status of the argument x_i;
- $\forall x_i, x_j \in A$, att_{x_i,x_j} represents the attack from x_i to x_j.

Φ_{st} is the formula $\Phi_{st} = \bigwedge_{x_i \in A}[acc_{x_i} \Leftrightarrow \bigwedge_{x_j \in A}(att_{x_j,x_i} \Rightarrow \neg acc_{x_j})]$. This modified version of the encoding from [24] describes in a generic way the relation between the structure of an AF (*i.e.* the set of attacks) and the arguments' acceptance (*i.e.* the extensions) w.r.t. stable semantics.

When the *att*-variables are assigned the truth value corresponding to the attack relation of $\mathcal{AF}$ (*i.e.* att_{x_i,x_j} is assigned 1 iff $(x_i,x_j) \in R$), the models of Φ_{st} (projected on the *acc*-variables) correspond in a bijective way to $\mathrm{st}(\mathcal{AF})$.

Given $\mathcal{AF} = \langle A, R \rangle$, we define the formula $\Phi_{st}^{R} = \Phi_{st} \wedge (\bigwedge_{(x_i,x_j)\in R} att_{x_i,x_j}) \wedge (\bigwedge_{(x_i,x_j)\notin R} \neg att_{x_i,x_j})$ which represents this assignment of *att*-variables corresponding to a specific AF. For any model ω of Φ_{st}^{R}, the set $\{x_i \mid \omega(acc_{x_i}) = 1\}$ is a stable extension of $\mathcal{AF}$. In the other direction, for any stable extension $\varepsilon \in \mathrm{st}(\mathcal{AF})$, ω s.t. $\omega(acc_{x_i}) = 1$ iff $x_i \in \varepsilon$ is a model of Φ_{st}^{R}.

These variables and formula are enough to encode the stable semantics of AFs. But to determine the controllability of a CAF, we need also to consider propositional variables to indicate which arguments are actually in the system:

- $\forall x_i \in A_C \cup A_U$, on_{x_i} is true iff x_i actually appears in the framework.

Now, we can recall the encoding which relates the attack relation and the arguments statuses in $\mathcal{CAF} = \langle F, C, U \rangle$ [14]:
Notation: $\mathbf{A} = A_F \cup A_C \cup A_U$, $\mathbf{R} = \rightarrow \cup \rightleftarrows \cup \dashrightarrow \cup \Rrightarrow$

$$\begin{gathered}\Phi_{st}(\mathcal{CAF}) = \textstyle\bigwedge_{x_i\in A_F}[acc_{x_i} \Leftrightarrow \bigwedge_{x_j\in\mathbf{A}}(att_{x_j,x_i} \Rightarrow \neg acc_{x_j})]\wedge \\ \textstyle\bigwedge_{x_i\in A_C\cup A_U}[acc_{x_i} \Leftrightarrow (on_{x_i} \wedge \bigwedge_{x_j\in\mathbf{A}}(att_{x_j,x_i} \Rightarrow \neg acc_{x_j}))]\wedge \\ \textstyle\bigwedge_{(x_i,x_j)\in\rightarrow\cup\Rrightarrow} att_{x_i,x_j}) \wedge (\bigwedge_{(x_i,x_j)\in\rightleftarrows} att_{x_i,x_j} \vee att_{x_j,x_i}) \wedge (\bigwedge_{(x_i,x_j)\notin\mathbf{R}} \neg att_{x_i,x_j})\end{gathered}$$

The first line states that an argument from A_F is accepted when all its attackers are rejected (similarly to the case of classical AFs). Then, the next line concerns arguments from A_C and A_U; since these arguments may not appear in some completions of the CAF, we add the condition that on_{x_i} is true to allow x_i to be accepted. The last line specify the case in which there is an attack in the completion: attacks from $\rightarrow$ and $\Rrightarrow$ are mandatory, and their direction is known; attacks from $\rightleftarrows$ are mandatory, but the actual direction is not known. We do not give any constraint about $\dashrightarrow$, which is equivalent to the tautological constraint $att_{x_i,x_j} \vee \neg att_{x_i,x_j}$: the attack may appear or not. Finally, we know that attacks which are not in $\mathbf{R}$ do not exist.

Given a set of arguments T, the fact that T must be included in all the stable extensions is represented by:

$$\Phi_{st}^{sk}(\mathcal{CAF}, T) = \Phi_{st}(\mathcal{CAF}) \Rightarrow \bigwedge_{x_i\in T} acc_{x_i}$$

Given a set of arguments T, the fact that T must be included in at least one stable extension is represented by:

$$\Phi_{st}^{cr}(\mathcal{CAF}, T) = \Phi_{st}(\mathcal{CAF}) \wedge \bigwedge_{x_i\in T} acc_{x_i}$$

Now we give the logical encodings for possible controllability for $\sigma = \mathrm{st}$.

Proposition 2. *Given $\mathcal{CAF}$ and $T \subseteq A_F$, $\mathcal{CAF}$ is possibly skeptically controllable w.r.t. T and the stable semantics iff*

$$\begin{array}{l}\exists\{on_{x_i} \mid x_i \in A_C\}\exists\{on_{x_i} \mid x_i \in A_U\} \\ \exists\{att_{x_i,x_j} \mid (x_i,x_j) \in \dashrightarrow \cup \rightleftarrows\}\forall\{acc_{x_i} \mid x_i \in \mathbf{A}\} \\ [\Phi^{sk}_{st}(\mathscr{CAF},T) \vee (\bigvee_{(x_i,x_j)\in\rightleftarrows}(\neg att_{a_i,a_j} \wedge \neg att_{a_j,a_i}))]\end{array} \quad (1)$$

is valid. In this case, each valid truth assignment of the variables $\{on_{x_i} \mid x_i \in A_C\}$ corresponds to a configuration which reaches the target.

This encoding is not a direct adaptation of the encoding proposed in [14]. We have to explicitly exclude the joint assignment of the variables att_{x_i,x_j} and att_{x_j,x_i} to false, when $(x_i,x_j) \in\rightleftarrows$, which would be in contradiction with the definition of this conflict relation. Another method is used in [14] to rule out these assignments, but it does not yield a QBF in prenex form. But this is the method that was proposed in [21], when necessary controllability has been applied to automated negotiation.

The following result holds for possible credulous controllability:

Proposition 3. *Given $\mathscr{CAF}$ and $T \subseteq A_F$, $\mathscr{CAF}$ is possible credulously controllable w.r.t. T and the stable semantics iff*

$$\begin{array}{l}\exists\{on_{x_i} \mid x_i \in A_C\}\exists\{on_{x_i} \mid x_i \in A_U\} \\ \exists\{att_{x_i,x_j} \mid (x_i,x_j) \in \dashrightarrow \cup \rightleftarrows\}\exists\{acc_{x_i} \mid x_i \in \mathbf{A}\} \\ [\Phi^{cr}_{st}(\mathscr{CAF},T) \vee (\bigvee_{(x_i,x_j)\in\rightleftarrows}(\neg att_{a_i,a_j} \wedge \neg att_{a_j,a_i}))]\end{array} \quad (2)$$

is valid. In this case, each valid truth assignment of the variables $\{on_{x_i} \mid x_i \in A_C\}$ corresponds to a configuration which reaches the target.

We notice that in the case of possible credulous controllability, the problem reduces to SAT since all the quantifiers are existential. This corresponds to the NP upper bound for possible credulous controllability under stable semantics (Proposition 1). We keep the QBF-style notation for homogeneity with Equation 1.

Example 4 (Continuation of Example 1). *Let us describe the logical encoding for possible controllability with $\mathscr{CAF}$ as described previously and $T = \{a_1\}$. We give here the example for possible skeptical controllability:*

$$\begin{array}{l}\exists on_{a_7}, on_{a_8}, on_{a_9}, \exists on_{a_6}, \exists att_{a_5,a_1}, att_{a_6,a_5}, att_{a4,a_6}, \\ \forall acc_{a_1}, acc_{a_2}, \dots, acc_{a_9}, \\ [\Phi^{sk}_{st}(\mathscr{CAF},T) \vee (\bigvee_{(x_i,x_j)\in\rightleftarrows}(\neg att_{a_i,a_j} \wedge \neg att_{a_j,a_i}))]\end{array}$$

*Below, we give the formula $\Phi^{sk}_{st}(\mathscr{CAF},T)$. For a matter of readability, several simplifications are made. For instance, an implication like $att_{x_j,x_i} \Rightarrow \neg acc_{x_j}$ can be removed when att_{x_j,x_i} is known to be false (because x_j does not attack x_i), and can be replaced by $\neg acc_{x_j}$ when att_{x_j,x_i} is known to be true. Only the uncertain attacks need to be kept explicit in the encoding. The first three lines give the condition for the acceptance of the fixed arguments. Then, two lines give the condition for the acceptance of the control and uncertain arguments. The other lines describe the structure of the graph (*i.e. *the attack relations), and the implication gives the target for skeptical acceptance.*

$$\begin{gathered}
[[acc_{a_1} \Leftrightarrow \neg a_2 \wedge \neg a_3 \wedge (att_{a_5,a_1} \Rightarrow \neg acc_{a_5})] \\
\wedge \\
[acc_{a_2} \Leftrightarrow \neg acc_{a_4})] \wedge [acc_{a_3} \Leftrightarrow \neg acc_{a_4})] \\
\wedge \\
[acc_{a_4} \Leftrightarrow (att_{a_6,a_4} \Rightarrow \neg acc_{a_6})] \wedge [acc_{a_5} \Leftrightarrow \neg acc_{a_7})] \\
\wedge \\
[acc_{a_6} \Leftrightarrow (on_{a_6} \wedge \neg acc_{a_8} \wedge \neg acc_{a_9} \wedge (att_{a_4,a_6} \Rightarrow \neg acc_{a_4}))] \\
\wedge \\
[acc_{a_7} \Leftrightarrow (on_{a_7} \wedge \neg acc_{a_8})] \wedge [acc_{a_8} \Leftrightarrow on_{a_8}] \wedge [acc_{a_9} \Leftrightarrow (on_{a_9} \wedge \neg acc_{a_7}))] \\
\wedge \\
att_{a_2,a_1} \wedge att_{a_3,a_1} \wedge att_{a_4,a_2} \wedge att_{a_4,a_3} \wedge att_{a_7,a_5} \wedge att_{a_7,a_9} \wedge att_{a_8,a_6} \wedge att_{a_8,a_7} \wedge att_{a_9,a_6} \\
att_{a_4,a_6} \vee att_{a_6,a_4} \\
\bigwedge_{(x_i,x_j) \notin \mathbf{R}} \neg att_{x_i,x_j} \\
] \Rightarrow acc_{a_1}
\end{gathered}$$

4. Related Work

Qualitative uncertainty has been considered in other frameworks. Partial AFs [20] are special instances of CAFs where only $\dashrightarrow$ is considered. They are used as a tool in a process of aggregating several AFs. Then [25] studies the complexity of verifying in a PAF whether a set of arguments is an extension of some (or every) completion. [26] conducts a similar study for argument-incomplete AFs, *i.e.* there is some uncertainty about the presence of arguments (the part called A_U)in our framework). Finally, [27] combines both. Let us notice than in [25,26,27], both versions of the verification problem (existential and universal w.r.t. the set of completions) are studied. As mentioned previously, [22] gives the complexity of skeptical and credulous acceptance for IAFs. While being a quite general model of uncertainty, this Incomplete AF is strictly included in the CAF setting: [26] does not allow to express the uncertainty about the direction of a conflict (*i.e.* our $\rightleftarrows$ relation cannot be encoded in this framework). Moreover, none of these works [20,25,26,27] is concerned with argumentation dynamics.

Quantitative models of uncertainty have also been used; while being an interesting approach, they require more input information than qualitative models like ours. This approach is out of the scope of this paper and is kept for future work. In particular, probabilistic CAFs based on the constellations approach [28] are a promising research tracks.

Argumentation dynamics has received a lot of attention in the last ten years. Except the initial paper about CAFs [14], most of the existing work consider complete information about the input (*i.e.* no uncertainty of the initial AF is considered). As far as we know, the only proposal which can encompass uncertainty is the update of AFs through the YALLA language [11]. However, YALLA pays the price of its expressiveness, and we are not aware of any efficient computational approach for reasoning with it, contrary to our QBF-based approach for CAFs.

5. Conclusion

In this paper, we push forward the study of the Control Argumentation Frameworks. We define a ”weaker” version of controllability, where a target set of arguments needs to be accepted in at least one completion (instead of every completion). This kind of reasoning is related to a lawyer’s plea: at the end of a trial, the lawyer needs to pick arguments (in our setting, the configuration A_{conf}) such that the target (“the defendant is innocent”) is accepted in at least one completion. Somehow, possible controllability is to necessary controllability what credulous acceptance is to skeptical acceptance.

Many research tracks are still open. We plan to propose logical encodings and to study the complexity of controllability for other extension-based semantics. Also, other methods can be used for computing control configuration, especially SAT-based counter-example guided abstract refinement (CEGAR), that was successfully used for reasoning problems at the second level of the polynomial hierarchy [12]. An interesting other form of controllability to be studied is ”optimal” controllability, *i.e.* finding a configuration that allows to reach the target in as many completions as possible. This is useful in situations where a CAF is not necessary controllable, and possible controllability seems too weak. Techniques like CEGAR or QBF with soft variables [29] may be helpful for solving this problem. Also, as mentioned previously, we will study quantitative models of uncertainty in the context of CAFs. In particular, it would be interesting for real world applications to define a form of controllability w.r.t. the most probable completion, or w.r.t. the set of completions with a probability higher than a given threshold. Finally, we think that an important work to be done, in order to apply CAFs to real applications scenarios, is to determine how CAFs and controllability can be defined when the internal structure of arguments (*e.g.* based on logical formulas or rules) is known.

References

[1] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Art Intel. 1995;77:321–357.

[2] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 159–236.

[3] Caminada M. On the Issue of Reinstatement in Argumentation. In: Proc. of JELIA’06; 2006. p. 111–123.

[4] Amgoud L, Ben-Naim J. Ranking-Based Semantics for Argumentation Frameworks. In: Proc. of SUM’13; 2013. p. 134–147.

[5] Boella G, Kaci S, van der Torre LWN. Dynamics in Argumentation with Single Extensions: Abstraction Principles and the Grounded Extension. In: Proc. of ECSQARU’09; 2009. p. 107–118.

[6] Cayrol C, de Saint-Cyr FD, Lagasquie-Schiex M. Change in Abstract Argumentation Frameworks: Adding an Argument. J Artif Intell Res (JAIR). 2010;38:49–84.

[7] Baumann R, Brewka G. Expanding Argumentation Frameworks: Enforcing and Monotonicity Results. In: Proc. of COMMA’10; 2010. p. 75–86.

[8] Coste-Marquis S, Konieczny S, Mailly J, Marquis P. On the Revision of Argumentation Systems: Minimal Change of Arguments Statuses. In: Proc. of KR’14; 2014. .

[9] Doutre S, Herzig A, Perrussel L. A Dynamic Logic Framework for Abstract Argumentation. In: Proc. of KR’14; 2014. .

[10] Coste-Marquis S, Konieczny S, Mailly J, Marquis P. Extension Enforcement in Abstract Argumentation as an Optimization Problem. In: Proc. of IJCAI’15; 2015. p. 2876–2882.

[11] de Saint-Cyr FD, Bisquert P, Cayrol C, Lagasquie-Schiex M. Argumentation update in YALLA (Yet Another Logic Language for Argumentation). Int J Approx Reasoning. 2016;75:57–92.

[12] Wallner JP, Niskanen A, Järvisalo M. Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation. J Artif Intell Res. 2017;60:1–40.

[13] Doutre S, Mailly J. Semantic Change and Extension Enforcement in Abstract Argumentation. In: Proc. of SUM'17; 2017. p. 194–207.

[14] Dimopoulos Y, Mailly JG, Moraitis P. Control Argumentation Frameworks. In: Proc. of AAAI'18; 2018. p. 4678–4685.

[15] Baumann R. What Does it Take to Enforce an Argument? Minimal Change in abstract Argumentation. In: Proc. of ECAI'12; 2012. p. 127–132.

[16] Biere A, Heule M, van Maaren H, Walsh T, editors. Handbook of Satisfiability. vol. 185 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2009.

[17] Kleine Büning H, Bubeck U. Theory of Quantified Boolean Formulas. In: Handbook of Satisfiability; 2009. p. 735–760.

[18] Arora S, Barak B. Computational Complexity - A Modern Approach. Cambridge University Press; 2009.

[19] Amgoud L, Vesic S. Rich preference-based argumentation frameworks. Int J Approx Reasoning. 2014;55(2):585–606.

[20] Coste-Marquis S, Devred C, Konieczny S, Lagasquie-Schiex M, Marquis P. On the merging of Dung's argumentation systems. Artif Intell. 2007;171(10-15):730–753.

[21] Dimopoulos Y, Mailly J, Moraitis P. Argumentation-based Negotiation with Incomplete Opponent Profiles. In: Proc. of AAMAS'19; 2019. p. 1252–1260.

[22] Baumeister D, Neugebauer D, Rothe J. Credulous and Skeptical Acceptance in Incomplete Argumentation Frameworks. In: Proc. of COMMA'18; 2018. p. 181–192.

[23] Dvorák W, Dunne PE. Computational Problems in Formal Argumentation and their Complexity. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 631–688.

[24] Besnard P, Doutre S. Checking the acceptability of a set of arguments. In: Proc. of NMR'04; 2004. p. 59–64.

[25] Baumeister D, Neugebauer D, Rothe J. Verification in Attack-Incomplete Argumentation Frameworks. In: Proc. of ADT'15; 2015. p. 341–358.

[26] Baumeister D, Rothe J, Schadrack H. Verification in Argument-Incomplete Argumentation Frameworks. In: Proc. of ADT'15; 2015. p. 359–376.

[27] Baumeister D, Neugebauer D, Rothe J, Schadrack H. Verification in incomplete argumentation frameworks. Artif Intell. 2018;264:1–26.

[28] Hunter A. A probabilistic approach to modelling uncertain logical arguments. Int J Approx Reasoning. 2013;54(1):47–81.

[29] Reimer S, Sauer M, Marin P, Becker B. QBF with Soft Variables. ECEASST. 2014;70.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200513

Analysing Product Reviews Using Probabilistic Argumentation

Kawsar NOOR [a], Anthony HUNTER [a]
[a] *Department of Computer Science, University College London*

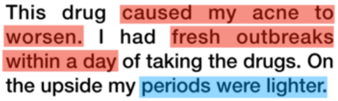
Abstract. Product reviews which are increasingly commonplace on the web typically contain a textual component and a numerical rating. The textual component can be viewed as a collection of arguments for and against the product. Whilst the reviewer may not have provided the attacks between these arguments they typically provide an indication of which set of arguments they view as being more acceptable/winning via the numerical rating (i.e. a positive rating indicates that the positive arguments are accepted and vice versa). Our framework builds upon this intuition and we propose a two step process for identifying a probability distribution over the set of possible argument graphs that the reviewer may have had in mind. The first is the *identification step* in which for a given review, we identify a distribution by analysing the relationship between the rating and polarity of arguments in the review via the constellations approach to probabilistic argumentation. The second step is the *refinement step* in which we harness ratings from multiple reviews and use this to refine our probability distribution thus enabling us to learn from the data. We illustrate the applicability of our approach by testing it with real data.

Keywords. Probabilistic argumentation; online reviews; abstract argumentation

1. Introduction

An abstract argument framework, as proposed by Dung, [7] is a graph structure in which the vertices denote arguments and the edges denote attacks between the arguments. Probabilistic argumentation extends abstract argumentation by allowing one to associate probabilities with the argument frameworks. In the epistemic approach probabilities are associated with arguments and represent uncertainty in the arguments themselves. In contrast in the constellations approach probabilities are associated with the topology of a graph and this enables one to model uncertainty in the structure of the graph.

In this paper we explore the use of the constellations approach to model agent reasoning within product reviews. These reviews contain arguments for and against the product; many reviews also have numerical ratings that capture the reviewer's overall sentiment towards the product. In essence the rating represents the reviewer's final verdict on the product and is provided in light of the arguments they have provided in favour of and against the product. With this in mind we therefore assume that for each review there is an underlying abstract argument graph which captures the reviewer's reasoning. In order to predict this graph we use the constellations approach to identify a probability distribution over potential argument graphs for each review. The distribution is useful as it can be used to predict a graph for the review that can be used to then understand the



This drug caused my acne to worsen. I had fresh outbreaks within a day of taking the drugs. On the upside my periods were lighter.

Figure 1. Example drug review in which the reviewer gave a rating of 3/10. Text spans labelled red indicate a negative argument against the drug and blue labels indicate positive arguments

reviewer's reasoning. Likewise when considering multiple reviews for a product one can develop an understanding of how all of the reviewers view a product by aggregating and reasoning with the distributions identified.

To illustrate the problem consider a reader browsing through many product reviews that contain arguments in favour of (**positive arguments**) or against (**negative arguments**) a particular product where each review comes with a numerical rating. We interpret this rating as a proxy for the polarity of the winning arguments. Hence we view a review with a high rating as an indication that the winning arguments are positive and vice versa. This also affords us an understanding of the potential graphs assignable to the review.

As an example consider the review shown in Figure 1. We can reason that the low rating is being driven by one or both of the negative arguments and consequently that the positive argument does not play much of a role in the overall assessment. We can express our reasoning using Dung's grounded semantics and say that the argument graph that the reviewer had in mind will likely have one or both of the negative arguments in its grounded extension and not the positive argument. This can be formalised further using probabilistic argumentation.

In this paper we propose a method for identifying a probability distribution over the set of graphs that the reviewer may have had in mind. This is achieved in two steps. The first is the *identification step* in which we identify a distribution for a review by building upon the assumption that there is a relationship between the rating and the winning / acceptable arguments in that review. This distribution can then be sampled from to assign a graph to the review. When considering multiple reviews we propose an additional *refinement step* that makes use of data derived from ratings taken from a dataset of reviews in order to refine the probability distribution so as to better reflect the data.

However not all reviews contain ratings and hence in our experiment section we demonstrate that we can train a machine learning model to predict ratings for such reviews. Also we see that our proposal is not limited to product reviews and can indeed be used in any situation in which agents posit arguments and proxy measures that indicate which arguments win. For example in a public debate where the viewers, over the course of the debate, accumulate arguments from both parties and instead of providing the attacks between these arguments or directly identifying the winning arguments they may instead rate each party thus indicating their overall verdict. Our approach could thus be used to identify a probability distribution over the set of possible argument graphs for each viewer.

To summarise we make two main contributions with this paper. Our first contribution is providing a methodology for identifying a probability distribution over a set of argument graphs given a review. Our second contribution is refining this distribution by incorporating data derived by analysing the ratings from multiple reviews and thus having a distribution that better reflects the reviews we are modelling.

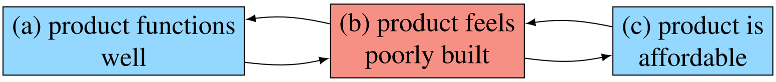

(a) product functions well
(b) product feels poorly built
(c) product is affordable

Figure 2. An example of an argument graph containing two positive arguments in favour of a product (a,c) and one negative argument against it (b)

2. Identifying a Probability Distribution for a Review

Given an argument graph (A,R) a set $B \subseteq A$ is **conflict-free** iff no two arguments $a,b \in B$ exists s.t $(a.b) \in R$. An argument b is **defended** by a set $B \in A$ iff any argument $a \in A$ attacks b then $\exists c \in B$ s.t. $(c,a) \in R$. A conflict-free set $B \subseteq A$ is an **admissible** extension iff each argument in B is defended by B. An admissible extension B is an **complete** extension iff each argument defended by B is in B. A complete extension B is a **grounded** extension if it is minimal (w.r.t set inclusion). We use the notation $\mathsf{gr}((A,R))$ to indicate the grounded extension for a graph.

We start by considering a setting in which users state positive and negative arguments which are arguments in favour of or against a particular conclusion (e.g. in the case of product reviews these are in favour or against the product). In other words we consider, as a simplifying assumption, only bipartite graphs. We thus see reviews as user provided arguments for or against the product and the rating they provide as indicators of the underlying argument graph and therefore the winning arguments. Although on the web ratings tend to be integers they can in fact be any real number.

Definition 2.1. Let A^+ be a set of positive arguments and A^- be a set of negative arguments s.t. $A^+ \cap A^- = \emptyset$. Let the minimum rating be b_{min}^{Neg} and the maximum be b_{max}^{Pos}. A **view** is a tuple $v = (A,b)$ where $A \subseteq A^+ \cup A^-$ and $b \in [b_{min}^{\mathsf{Neg}}, b_{max}^{\mathsf{Pos}}]$ is a **rating** s.t $b_{min}^{\mathsf{Neg}}, b_{max}^{\mathsf{Pos}} \in \mathbb{R}$ and $b_{min}^{\mathsf{Neg}} < b_{max}^{\mathsf{Pos}}$.

Example 2.1. Consider the arguments depicted in Figure 2 and rating. Some examples of views using the arguments $\{a,b,c\}$ and ratings in the range $[1,10]$ would be $(\{a,b,c\},9)$ and $(\{a,b\},10)$.

When considering the set of possible argument graphs that an agent may have had in mind when providing a view we are dealing with all argument graphs which contain the arguments in that view. This translates as the set of all spanning sub graphs using those arguments. We refer to this set as the **graph space**. Formally we say that given disjoint sets A^+ and A^- that the graph space is the set returned by the function $\mathsf{Space}(A^+,A^-) = \{(A^+ \cup A^-,R) | R \in \mathscr{P}((A^+ \times A^-) \cup (A^- \times A^+)\}$. An example of a a graph space given two positive and one negative argument is provided in Table 1.

Proposition 2.1. *Given a set of positive and negative arguments A^+ and A^- let $m = |A^+|$ and $n = |A^-|$. The size of the graph space is then 2^{2mn}.*

In identifying the probability distribution for a view we make the assumption that the rating in a view is proportional to the ratio of positive/negative arguments in the grounded extension for the graph the agent intended for that view; hence if the rating is high we expect this ratio to be high and vice versa. In terms of the graph space we expect that when a rating is high, those graphs that have a high proportion of positive arguments in their grounded extension will have more mass assigned to them and vice versa.

To this end we rank graphs in the graph space based on two criteria which we define in the rest of this section: the degree of polarity of the graph's grounded extension (proportion of positive/negative arguments) and an assessment of the topological structure of each graph. An analysis of the graph's attack structure provides a finer-grained understanding of how the grounded extension is achieved, therefore enabling us to better differentiate between graphs that share the same grounded extension.

Definition 2.2. Let A^+, A^- be positive and negative arguments and $S = \mathsf{Space}(A^+,A^-)$ be a graph space. For each graph $G \in S$ we define the sets $\mathsf{gr}^+(G) = \{a \in \mathsf{gr}(G) | a \in A^+\}$ and $\mathsf{gr}^-(G) = \{a \in \mathsf{gr}(G) | a \in A^-\}$. We then say that the **polarity** of a graph G is $\mathsf{Pol}(G) = |\mathsf{gr}^+| - |\mathsf{gr}^-|$ and that the graph space can be partitioned into the following sets: $\mathsf{Pos}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) > 0\}$, $\mathsf{Ntl}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) = 0\}$ and $\mathsf{Neg}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) < 0\}$.

Proposition 2.2. $\mathsf{Pol}(G) \in \mathbb{Z}$ *(i.e. the set of integers) and* $|A^-| \leq \mathsf{Pol}(G) \leq |A^+|$.

Proposition 2.3. *For any graph space in which there are m positive arguments and n negative arguments then* $|\mathsf{Pos}(S)| > |\mathsf{Neg}(S)|$ *when* $m > n$ *and* $|\mathsf{Neg}(S)| > |\mathsf{Pos}(S)|$ *when* $n > m$.

To analyse the polarity of an argument graph based on its attacks we define a function that scores each argument based on the number of attacks it inflicts and sustains.

Definition 2.3. Let $G = (A,R)$ be an argument graph. For an argument $a \in A$ the number of attacks it receives is $\mathsf{def}(a) = |\{(x,y) \in R | y = a\}|$ and the number of attacks it inflicts is $\mathsf{att}(a) = |\{(x,y) \in R | x = a\}|$. The **grade** of argument a in graph G is then $\mathsf{Grade}(a,G) = \mathsf{att}(a) - \mathsf{def}(a)$.

Example 2.2. Consider graph G_1 in Table 1. We can see that $\mathsf{Grade}(a,G_1) = 1$, $\mathsf{Grade}(c,G_1) = 1$ and $\mathsf{Grade}(b,G_1) = -2$.

The grade of an argument is maximal when it attacks all of its opponents without being attacked at all and vice versa.

Proposition 2.4. *Let* $\mathsf{Space}(A^+,A^-)$ *be a graph space. Given* $B \in \{A^+,A^-\}$ *and* $a \in B$ *then* $\mathsf{max}_{G \in \mathsf{Space}(A^+,A^-)} \mathsf{Grade}(a,G) = |A^+ \cup A^- \setminus B|$ *and* $\mathsf{min}_{G \in \mathsf{Space}(A^+,A^-)} \mathsf{Grade}(a,G) = -|A^+ \cup A^- \setminus B|$.

Proposition 2.5. *Given an argument graph G it holds that* $\sum_{a \in A^+} \mathsf{Grade}(a,G) + \sum_{b \in A^-} \mathsf{Grade}(b,G) = 0$.

The grade of an argument is a score that is a combined indicator of an argument's ability to defend its coalition whilst not being attacked by the opposition in a particular graph [10]. Given our aim of ranking graphs in a graph space we are however interested in comparing the grade of an argument in a particular graph to its grades in the other graphs in the graph space so as to assess how well it performed in that particular graph. In order to gain this relative perspective we use a process of normalisation as follows:

Definition 2.4. Given a graph space $S = \mathsf{Space}(A^+,A^-)$ and an argument $a \in A^+ \cup A^-$ the **normalised grade** for a is given below where $\mathsf{min}(a,S) = \min_{G \in S} \mathsf{Grade}(a,G)$ and $\mathsf{max}(a,S) = \max_{G \in S} \mathsf{Grade}(a,G)$.

$$\mathsf{NormGrade}(G,S,a) = \frac{\mathsf{Grade}(G,a) - \mathsf{min}(a,S)}{\mathsf{max}(a,S) - \mathsf{min}(a,S)}$$

Example 2.3. Consider Table 1 which uses the arguments $a,c \in A^+$ and $b \in A^-$. The normalised grade for arguments a,c are highest in G_1 as they are attacking all of their opponents and not being attacked. The opposite is true in G_{16}.

Proposition 2.6. *Given a graph space $S = \mathsf{Space}(A^+,A^-)$ where $p = |A^+|$ and $n = |A^-|$ then for a positive argument $a \in A^+$, $\mathrm{max}_{G\in S}\,\mathsf{Grade}(G,a) = n$ and $\mathrm{min}_{G\in S}\,\mathsf{Grade}(G,a) = -n$. Likewise for a negative argument that $b \in A^-$, $\mathrm{max}_{G\in S}\,\mathsf{Grade}(G,b) = p$ and $\mathrm{min}_{G\in S}\,\mathsf{Grade}(G,b) = -p$.*

By summing the normalised grades for the arguments in an argument graph we are able to produce a value that summarises the polarity of attacks in that graph.

Definition 2.5. Given a graph space $S = \mathsf{Space}(A^+,A^-)$, for each $G \in S$ the aggregate score for positive arguments is $AttackScore^+ = \sum_{a\in A^+} \mathsf{NormGrade}(G,S,a)$ and the aggregate score for negative arguments is $AttackScore^- = \sum_{a\in A^-} \mathsf{NormGrade}(G,S,a)$. The **aggregate polarity score** for the graph is then given by $\mathsf{AttackScore}(S,G) = AttackScore^+ - AttackScore^-$.

When considering the ordered set of attack scores for graphs in S the difference in attack score between any two consecutive graphs is a constant ΔAtt as given in the following result.

Proposition 2.7. *Let $p = |A^+|$, $n = |A^-|$ and $(AttackScore_0,..,AttackScore_m)$ be a sequence of all the attack scores ordered from largest to smallest in the set $\{\mathsf{AttackScore}(G)|G \in S\}$ s.t for each i, $AttackScore_{i+1} \geq AttackScore_i$. For any two values in the sequence it holds that the pairwise difference between them is a constant ΔAtt i.e. ΔAtt = $AttackScore_i - AttackScore_{i+1}$ where $\Delta Att = \frac{1}{2n} + \frac{1}{2p}$.*

We can then use ΔAtt to bring together the functions Pol and $\mathsf{AttackScore}$ to give a combined assessment of the polarity of the graphs in a graph space.

Definition 2.6. Let $(G_1,..,G_m)$ be a sequence of graphs in S s.t. for any two graphs G_i, G_{i+1} it holds that $\mathsf{Pol}(G_i) \geq \mathsf{Pol}(G_{i+1})$ and $\mathsf{AttackScore}(G_i) \geq \mathsf{AttackScore}(G_{i+1})$. We say that $\mathsf{alike}(G_i,G_{i+1})$ holds iff $\mathsf{Pol}(G_i) = \mathsf{Pol}(G_{i+1})$ and $\mathsf{AttackScore}(G_i) = \mathsf{AttackScore}(G_{i+1})$. We define the **aggregate score** of a graph G_i s.t. $i > 1$, as $\mathsf{Agg}(G_i) = \mathsf{Agg}(G_{i-1})$ if $\mathsf{alike}(G_i,G_{i-1})$ and $\mathsf{Agg}(G_i) = \mathsf{Agg}(G_{i-1} - \Delta Att$ otherwise and $\mathsf{Agg}(G_1) = \mathsf{AttackScore}(G_1)$.

To illustrate we can see that in Table 1 the attack scores are non-unique. The *Agg* function enables us distinguish between graphs such as G_{10} and G_{11} which share the same attack score but not the same grounded extension.

We now consider how ratings can be used in identifying a probability distribution for a view. Our proposal for this is a function which maps a rating to an aggregate value which is in turn used in specifying our probability distribution. In order to produce this polynomial we partition the rating scale into three categories which correspond to the three categories of polarity defined in Definition 2.2.

No	Graph	Gr(G)	Attack Score	Agg	P(G)						
					10	9	8	7	6	5	4,3,2,1
G_1	a → b ← c	a,c	2	2	0.16	0.06	0.01	0	0	0	0
G_2	a b ← c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_3	a → b c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_4	a → b ↔ c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_5	a ↔ b ← c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_6	a → b → c	a,c	0.5	0.5	0.08	0.12	0.11	0.05	0.03	0.02	0.02
G_7	a ← b ← c	a,c	0.5	0.5	0.08	0.12	0.11	0.05	0.03	0.02	0.02
G_8	a b ↔ c	a	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_9	a ↔ b c	c	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_{10}	a b c	a,b,c	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_{11}	a ↔ b ↔ c		0.5	-1	0.02	0.03	0.07	0.15	0.11	0.09	0.08
G_{12}	a b → c	a,b	-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{13}	a ← b c	b,c	-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{14}	a ↔ b → c		-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{15}	a ← b ↔ c		-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{16}	a ← b → c	b	-1	-2.5	0	0	0	0.03	0.01	0.13	0.19

Table 1. Breakdown of probability distribution and aggregate graded scores for each graph in a graph with 2 positive arguments and one negative

Definition 2.7. Let $[b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}]$ be the range of possible ratings assignable in a view where $b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}, \in \mathbb{R}$. Within this range we define the positive partition as $[b^{\mathsf{Pos}}_{min}, b^{\mathsf{Pos}}_{max}]$, the neutral partition as $[b^{\mathsf{Ntl}}_{min}, b^{\mathsf{Ntl}}_{max}]$ and the negative partition as $[b^{\mathsf{Neg}}_{min}, b^{\mathsf{Neg}}_{max}]$ s.t $b^{\mathsf{Pos}}_{max} > b^{\mathsf{Pos}}_{min} > b^{\mathsf{Ntl}}_{max} > b^{\mathsf{Ntl}}_{min} > b^{\mathsf{Neg}}_{max} > b^{\mathsf{Neg}}_{min}$.

Example 2.4. Consider a set of views that use a rating scale range $[1, 10]$. In this case $b^{\mathsf{Neg}}_{min} = 1$ and $b^{\mathsf{Pos}}_{max} = 10$. Example boundaries in between this range could be $b^{\mathsf{Neg}}_{min} = 4$, $b^{\mathsf{Ntl}}_{min} = 5$, $b^{\mathsf{Ntl}}_{max} = 7$ and $b^{\mathsf{Pos}}_{min} = 8$.

We can now relate these three partitions to the three sets $\mathsf{Pos}(S), \mathsf{Ntl}(S), \mathsf{Neg}(S)$ in the graph space using a polynomial function that allows us to go from ratings to Agg scores.

Definition 2.8. Let $\sigma \in \{max, min\}$, $polarity \in \{\mathsf{Pos}, \mathsf{Ntl}, \mathsf{Neg}\}$ and $V = \{b^{\mathsf{Pos}}_{max}, b^{\mathsf{Pos}}_{min}, b^{\mathsf{Ntl}}_{max}, b^{\mathsf{Ntl}}_{min}, b^{\mathsf{Neg}}_{max}, b^{\mathsf{Neg}}_{min}\}$. We say the corresponding **aggregate value** for a boundary $b^{polarity}_{\sigma} \in V$ is given by $\Gamma(b^{polarity}_{\sigma}) = \sigma_{G \in polarity(S)}(\mathsf{Agg}(G))$. In the case that $\mathsf{Pos}(S)$ is a singleton set then $\Gamma(b^{\mathsf{Pos}}_{max}) = \mathsf{max}_{G \in \mathsf{Pos}(S)}(\mathsf{Agg}(G)) + \Delta Att$ and likewise if $\mathsf{Neg}(S)$ is a singleton set then $\Gamma(b^{\mathsf{Neg}}_{min}) = \mathsf{min}_{G \in \mathsf{Neg}(S)}(\mathsf{Agg}(G)) - \Delta Att$. We then say that the set of all corresponding **aggregate coordinates** is $AggCoordinates = \{(b, \Gamma(b)) | b \in V\}$.

Example 2.5. In Table 1 we have $AggCoordinates = ((10, 2), (8, -0.25), (7, -1), (5, -1.75), (4, -2.5), (1, -3.25))$.

With the coordinates $AggCoordinates$ we can then fit our polynomial function which will enables us to map a rating to an aggregate score. We experimented with randomly generated graphs of different sizes and found that second order polynomials were sufficient for fitting to these coordinates.

Definition 2.9. Given aggregate coordinates $AggCoordinates$ and a rating b we define a function $\mathsf{ratingToAgg} : b \rightarrow \mathbb{R}$ which is a second-order polynomial function $\mathsf{ratingToAgg}(b) = c_0 b^2 + c_1 b + c_2$ where $c_0, c_1, c_2 \in \mathbb{R}$. The coefficients $c_0, c_1, c_2 \in \mathbb{R}$ are learnt by fitting $AggCoordinates$ to the polynomial using the least squares approximation method.

The $\mathsf{ratingToAgg}$ function provides an aggregate score for a view based on its rating. Using this we calculate the differences between this aggregate score and the aggregate scores of all of the graphs in the graph space. These differences serve as the basis for identifying a probability distribution. In essence we want those graphs that have a similar aggregate score to be assigned a larger probability mass.

Definition 2.10. Given a function $\mathsf{ratingToAgg}$ and a rating $b \in [b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}]$ we define a distance function $\mathsf{AggDist}(G, b) = \frac{1}{1+|\mathsf{Agg}(G)-\mathsf{ratingToAgg(b)}|^2}$. We then define a probability mass function for a graph G in graph space S as $P(G, b) = \frac{\mathsf{AggDist}(G,b)}{\sum_{G \in S} \mathsf{AggDist}(G,b)}$

Example 2.6. Table 1 shows two probability distributions for ratings in range $[1, 10]$. In this example because there are more positive than negative arguments, and hence more graphs with a positive grounded extension, the probability mass is distributed across more graphs.

In this section we have defined a method for identifying a probability distribution for a view using the intuition that the rating provided in a view is a proxy for understanding the agent's belief in the polarity of the winning arguments.

3. Refining a Probability Distribution Using Impacts

In the previous section we identified a probability distribution for a view based on the rating alone. In this section we propose improving this distribution by incorporating real data about arguments derived from a set of views. We propose a simple measure which captures the general influence a particular argument has on a rating when it appears in a review.

Definition 3.1. Given a set of reviews Rev, and boundaries b^{Pos}_{max}, b^{Neg}_{min} an argument a, the set of reviews the argument appears in is given by $\mathsf{App}(a, Rev) = \{rev \in Rev | rev = (A, r) \,\&\, a \in A\}$. We denote the number of reviews it appears in as $N = |\mathsf{App}(a, Rev)|$. The sum of the ratings is then $\mathsf{sum}(a, Rev) = \sum_{(a,r) \in \mathsf{App(a,Rev)}} r - b^{\mathsf{Neg}}_{min}$. The impact of the argument is then given below.

$$\mathsf{Impact}(a, Rev) = \begin{cases} \dfrac{\mathsf{sum}(a, Rev)}{(b^{\mathsf{Pos}}_{max} - b^{\mathsf{Neg}}_{min}) \times N} & \text{if} \quad a \in A^+ \\[2ex] 1 - \dfrac{\mathsf{sum}(a, Rev)}{(b^{\mathsf{Pos}}_{max} - b^{\mathsf{Neg}}_{min}) \times N} & \text{if} \quad a \in A^- \end{cases}$$

The impact of an argument tells us how much the argument caused the ratings of the reviews that it appeared in to move towards its polarity (positive or negative).

Example 3.1. Consider a set of reviews $Rev = \{(\{a,b,c\},9),(\{a,b,c\},8),(\{a,d\},7),(\{b,c\},2)\}$. where $A^+ = \{a,c\}$, $A^- = \{b,d\}$, $b^{\mathsf{Neg}}_{min} = 0$ and $b^{\mathsf{Pos}}_{max} = 10$. The impacts are then $\mathsf{Impact}(a,Rev) = 0.8$, $\mathsf{Impact}(b,Rev) = 0.63$, $\mathsf{Impact}(c,Rev) = 0.36$ and $\mathsf{Impact}(d,Rev) = 0.3$.

We interpret impact as a measure of relative strength of an argument. In the previous section we defined the relative strength of an argument using the normalised grade score. Hence in order to incorporate the impacts we weight those argument graphs whose normalised grade values resemble the impact values we have calculated.

Definition 3.2. Given a set of reviews Rev, a review $(A,r) \in Rev$, the corresponding graph space S for the review and a graph $G \in S$, the **similarity** between the impacts of the arguments A and their grades in graph G is given by $\mathsf{sim}(A,Rev,G) = \sqrt{\sum_{a\in A}}(\mathsf{Impact}(a,Rev) - \mathsf{NormGrade}(G,a))^2$.

Proposition 3.1. *For all A, Rev, G, it holds that* $0 \leq \mathsf{sim}(A,Rev) \leq \sqrt{|A|}$.

There is a natural correspondence between impact and graded score as they both are indicators of the degree of importance an argument plays in a graph/review. Hence when we find graphs where the difference between these values is small for all arguments we want to increase our probability assignment to such graphs.

Definition 3.3. Let (A,r) $Revs$ be a review, and S a graph space. Given a graph $G \in S$ we say that $d_G = \frac{1}{1+\mathsf{sim}(A,Rev,G))}$. The update weight associated with graph G is then $\mathsf{Weight}(G,r) = \frac{d_G \times P(G,r)}{\sum_{F\in S} d_F \times P(F,r)}$.

The weight assigned to each graph is thus the product of the probability of the graph and the inverse distance of the graph's grades to the argument's impacts. The normalising constant in the denominator ensures that the distribution of weights across the graph space is a probability distribution.

Example 3.2. Continuing from Example 3.1 if we now consider a review $(\{a,b,c\},9)$ we find that the largest weights are $\mathsf{Weight}(G_6,9) = 0.3$; this makes sense in this graph a has the highest grade followed by b and then c. We also see that $\mathsf{Weight}(G_6,3) = \mathsf{Weight}(G_6,3) = 0.10$ and that $\mathsf{Weight}(G_2,3) = \mathsf{Weight}(G_5,3) = 0.05$.

In this section we have proposed a method for incorporating data taken from sets of reviews to be able to identify probability distributions that better reflect the ratings in the reviews.

4. Experiment

In this section we demonstrate our framework using a set of reviews taken from the Drugs.com website. The dataset contains 601 reviews pertaining to the condition acne where each review contains a textual review and a rating between 1 and 10. In order to identify positive and negative arguments the first author identified arguments in the text for each review and assigned each argument a label (e.g. 'bearable side effect' etc) that best described it. Each label therefore denotes a different type of argument. In total 41 argument labels were used with a total of 2000 arguments being identified from all reviews. Following this we took 29 reviews and asked two annotators (neither of whom were authors) to provide an argument graph for each review using the identified arguments. We note that our paper is not intended as an argument mining framework and hence we are not focused on evaluating the quality of the argument labels, rather we want to evaluate our proposal for predicting an appropriate argument graph.

To report inter-annotator agreement we measured the degree of overlap between the grounded extensions of between the annotator's graphs. Popular inter-annotators agreement measures, such as Kappa-score, were not used as these measures are suitable for binary/categorical annotations and not graph structures.

Definition 4.1. For an actual graph graph G and a predicted graph $\hat{G}$, the **extension performance** is given by the function $\mathsf{GroundedPerformance} = \frac{|\mathsf{gr}(G) \setminus \mathsf{gr}(\hat{G})| + |\mathsf{gr}(\hat{G}) \setminus \mathsf{gr}(G)|}{|\mathsf{gr}(G)| + |\mathsf{gr}(\hat{G})|}$

The function $\mathsf{GroundedPerformance}$ is 0 when the both graphs have exactly the same extension and 1 when they share no arguments in common. The average $\mathsf{GroundedPerformance}$ between the annotators was 0.16. To produce the final dataset annotators were asked to resolve conflicts in annotation between themselves.

We used this annotated data [1] to evaluate our approach in a two part experiment.. In the first part we trained a machine learning model to predict ratings for reviews using the full dataset and thus illustrate that it is possible to train such models to predict ratings for reviews that do not have any. In the second step we identified a probability distribution over the constellation of possible graphs for each of the 29 dual annotated reviews and sample from the distributions in order to predict a graph for each review. We measured the performance of our approach by comparing our predicted graphs to the graphs acquired through the annotators. Hence we required independent annotators for the argument graphs given a set arguments but not for the identification of those arguments.

4.1. Predicting Ratings for Reviews

We trained a 2-layer feed-forward multi-layer neural network to predict ratings for each review. We modelled each review as a binary vector of arguments. Our architecture consisted of 250 neurons in the hidden layers and a single neuron in the output layer. We chose a softmax activation function for the hidden layer and a linear activation function for the output layer. The model was trained using standard backpropogation with a mean-square-error loss function. All of our code was implemented using the Keras Python library. We used a training: validation split of 80:20 for our dataset.

After 150 iterations of training we achieved a mean absolute percentage error (MAPE) of 30.86 %. MAPE is a standard loss measurement when training regression models; it is defined as $\frac{100\%}{n} \sum_{t-1}^{n} \left| \frac{X_t - Y_t}{X_t} \right|$ where X_t is the ground truth, Y_t the predicted value and n the number of datapoints. Our reported MAPE suggests the model can generally predict near to the correct rating thus suggesting that their is a correlation between the polarity of arguments and the ratings. The hardest ratings to predict were the ratings between 4 and 6. This we believe is partly due to the quality of the original reviews; a number of times it was noted that the arguments in a review did not always match the rating provided.

[1] https://github.com/robienoor/constellationsDataReviews

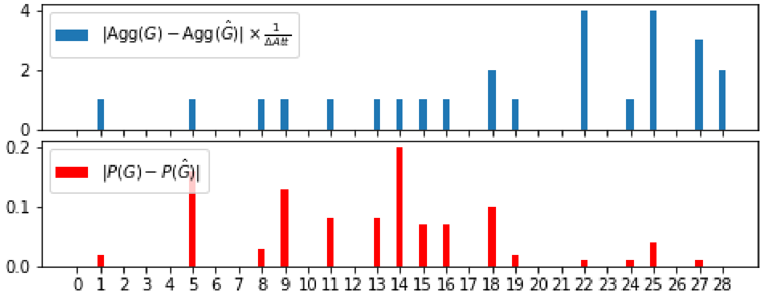


Figure 3. Performance results where each tick on the x-axis represents one of the 29 annotated graphs. The upper graph shows the aggregate distance between actual graph G and predicted graph $\hat{G}$. The bottom graph shows how far G was from $\hat{G}$ in terms of probabilities .

4.2. Predicting Graphs for Reviews

In this section we discuss the process of predicting argument graphs for reviews. We used the 29 argument graphs acquired from the annotators. For each review we identified a probability distribution using our approach and then sampled from this distribution in order to assign a graph to the review. For sampling we took the graph with the highest probability. In the case that we returned multiple graphs we simply randomly sample from the returned set of graphs.

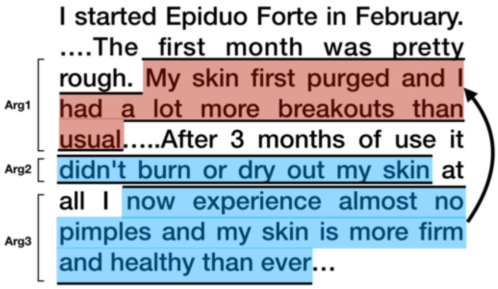
In order to measure the performance of our model we used two measurements in addition to GroundedPerfomance. For the first additional measure, we took the difference between the aggregate score of the predicted graphs and the actual graph. As per Definition 2.6 the aggregate scores for graphs in a graph space differ in units of ΔAtt. Graphs that share the same aggregate score are thus viewed as effectively belonging to the same equivalence class. This is captured by the following function that measures the number of equivalence classes by which the actual and predicted graph differ by.

Definition 4.2. Given a graph space S and set $Aggs = \{\mathrm{Agg}(G) | G \in S\}$ and a ground truth graph G and predicted graph $\hat{G}$ s.t. $G, \hat{G} \in S$ we define an **aggregate distance function** $\mathrm{AggDist}(G, \hat{G}) = \frac{|\mathrm{Agg}(G) - \mathrm{Agg}(\hat{G})|}{\Delta Att}$.

Example 4.1. Consider the example in Table 1 where $\Delta Att = 0.75$ and assume $G = G_1$ and $\hat{G} = G_{10}$. $\mathrm{AggDist}(G, \hat{G}) = 2.25/0.75 = 3$.

For the second additional measure we took the difference between the probability of the predicted graph and the actual graph. The results for the aggregate measurement and the probability measurement are depicted in Figure 3.

We found that the average GroundedPerformance was 0.30. In the cases where we identified an incorrect grounded extension we were either adding an additional argument or removing one and in other words we were not far off from the actual extension. In terms of aggregate distance we were never far off in terms of equivalence class as can be see in Figure 3 and likewise for the probability. Figure 4 depicts a review, in which three argument types where identified, and the attacks where assigned using our probabilistic model.



I started Epiduo Forte in February. ....The first month was pretty rough. My skin first purged and I had a lot more breakouts than usual.....After 3 months of use it didn't burn or dry out my skin at all I now experience almost no pimples and my skin is more firm and healthy than ever...

Figure 4. A shortened review for the acne drug Epiduo with arguments annotated. Three arguments were identified. The graph sampled from the corresponding graph space is depicted above with Arg3 attacking Arg1

We have demonstrated in this section the end-to-end process of using our framework for predicting argument graphs for reviews. We started by demonstrating that ratings could be reasonably predicted for reviews by using off the shelf machine learning algorithms. We then used our framework to identify probability distributions for each review before finally sampling from this distribution to predict the correct graph for the review.

5. Related Works

In another proposal for generating probability distributions over constellations of arguments graph [1] it is assumed that an agent(s) specifies a belief in the acceptability status of arguments. Using this data the paper proposes methods for aggregating, combining and summarising these beliefs. Whilst related to this paper, we have a different starting point which is that we do not have access to such beliefs directly rather we have access to ratings which we process to produce a distribution over a set of argument graphs.

There have been a few proposals for argument graphs learning algorithms when in/out/un labellings are provided by agents. In [3] a learning algorithm is proposed which takes as input a probability distribution over a set of in/un/out labellings. The algorithm is an on-the-fly algorithm to aggregate these labellings into a weighted argumentation graph. In our case we deal with a setting in which we do not have access to such labellings and furthermore we produce a distribution over a constellation of argument graphs. Likewise [4] makes a similar starting assumption in that the algorithm begins which a set of labellings for each argument. A Bayesian approach is proposed in order to learn from these labellings a posterior distribution for a set of arguments being in an extension. Both of these papers differ from our approach in that we do not assume we have such labelled data rather. Another proposal in [2] provides a method for extracting bipolar argument frameworks from a set of movie reviews. Each review contains a textual review and a binary rating indicating whether the reviewer thought the movie was good or bad. The proposed algorithm produces a quantitative bipolar argument per review which differs from our probabilistic output.

Various proposals for capturing and aggregating views taken from the social web have also been made [6][5]. These proposals use judgement aggregation and voting mechanisms to produce the aggregation which differs from our approach which produces probabilistic interpretation of views.

In summary our proposal differs primarily from the existing literature in that it is driven by our interpretation of ratings. The notion of rating is not dealt with explicitly in the literature and certainly not in a probabilistic context.

6. Discussion

In this paper we have proposed a methodology for identifying a probability distribution for a review. In the *identification step* this is done by exploiting the relationship between the rating and the accepted arguments in that review. We considered a situation in which we deal with bipartite argument graphs but this could be generalised to handle multipartite graphs. We further provided a *refinement step* for utilising information extracted from a set of reviews so as to enrich the identified probability distribution. We illustrated our approach using an annotated dataset and highlighted how machine learning models can be employed to provide ratings for reviews without ratings.

In future work we wish to ensure that our proposal is scalable given that the constellations approach can be computationally challenging [11]. We intend to do this by developing an understanding of the underlying combinatorics as well as the potential of approximation techniques. We also wish to experiment with other implementations of the grading function to see if we can improve the distribution. Likewise we wish to explore the use of additional acceptability semantics in order to enrich our function for partitioning the graph space based on polarity.

References

[1] Hunter, Anthony, and Kawsar Noor. “Aggregation of Perspectives Using the Constellations Approach to Probabilistic Argumentation.” Proceedings of the AAAI ‘20.

[2] Cocarascu, Oana, Antonio Rago, and Francesca Toni. “Extracting dialogical explanations for review aggregations with argumentative dialogical agents.” Proceedings of AAMAS ‘19.

[3] Riveret, Régis, and Guido Governatori. “On learning attacks in probabilistic abstract argumentation.” Proceedings of AAMAS ‘16.

[4] Kido, Hiroyuki, and Keishi Okamoto. “A Bayesian Approach to Argument-Based Reasoning for Attack Estimation.” Proceedings of IJCAI ‘17.

[5] Leite, Joao, and Joao Martins. “Social abstract argumentation.” Proceedings of IJCAI ‘11.

[6] Noor, Kawsar, Anthony Hunter, and Astrid Mayer. “Analysis of medical arguments from patient experiences expressed on the social web.” Proceedings of IEA/AIE ‘17.

[7] Dung, Phan Minh. “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games.” Artificial intelligence 77.2 (1995): 321-357.

[8] Li, Hengfei, Nir Oren, and Timothy J. Norman. “Probabilistic argumentation frameworks.” Proceedings of TAFA ‘11.

[9] Hunter, Anthony. “Some foundations for probabilistic abstract argumentation.” Proceedings of COMMA ‘12.

[10] Bonzon, Elise, et al. “A comparative study of ranking-based semantics for abstract argumentation.” Proceeding of AAAI ‘16.

[11] Fazzinga, Bettina, Sergio Flesca, and Francesco Parisi. “Efficiently estimating the probability of extensions in abstract argumentation.” Proceedings of SUM ‘13.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200514

Estimating Stability for Efficient Argument-Based Inquiry

Daphne ODEKERKEN [a,b], AnneMarie BORG [a] and Floris BEX [a,c]
[a] *Department of Information and Computing Sciences, Utrecht University* [1]
[b] *National Police Lab AI, Netherlands Police*
[c] *Tilburg Institute for Law, Technology and Society, Tilburg University*

Abstract. We study the dynamic argumentation task of detecting stability: given a specific structured argumentation setting, can adding information change the acceptability status of some propositional formula? Detecting stability is not tractable for every input, but efficient computation is essential in practical applications. We present a sound approximation algorithm that recognises stability for many inputs in polynomial time and we discuss several of its properties. In particular, we show under which constraints on the input our algorithm is complete. The proposed algorithm is currently applied for fraud inquiry at the Dutch National Police - we provide an English demo version that also visualises the output of the algorithm.

Keywords. dynamic argumentation, structured argumentation, inquiry

1. Introduction

One task of the police is the intake of citizens' reports on crimes: the citizen tells the police what happened; subsequently, additional questions can be asked to determine if the citizen has been the victim or witness of a crime. Certain high-volume crimes can be reported online. This can be as simple as filling out a web form, but can also be a more involved online dialogue with a (possibly artificial) agent. One specific high volume crime that can be reported online at the Dutch National Police is internet trade fraud. This concerns fake web shops and malicious second-hand traders on platforms such as eBay. In [3], an initial sketch was given for an artificial agent handling the intake of internet trade fraud by combining natural language processing with symbolic techniques for reasoning about crime reports. During the subsequent development of the intake agent, we regarded intake as *argument-based inquiry* [4]. In this inquiry, defeasible rules representing the laws and practices surrounding trade fraud are combined with the citizen's knowledge of the specific situation they observed, to build arguments for and against the main claim made by the citizen: that they have been the victim of trade fraud.

The first contribution of this paper is to present the implemented version of the intake agent. It has been released on the web site of the Dutch Police[2] where it handles the intake of hundreds of fraud reports every day. Because the police web site only shows

[1] This research has been partly funded by the Dutch Ministry of Justice and the Dutch National Police.
[2] https://aangifte.politie.nl/iaai-preintake

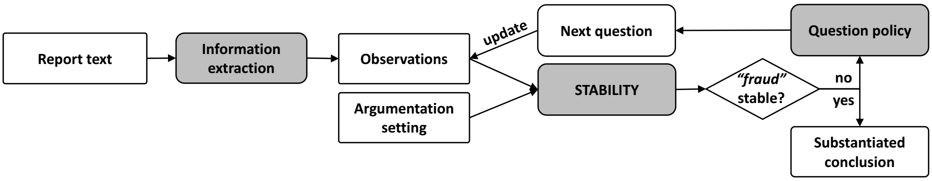


Figure 1. *Overview of the hybrid inquiry agent for the intake of fraud complaints.*

the Dutch user interface, we provide a demo[3] of an English version that gives more insight in the underlying reasoning. The agent's architecture is illustrated in Figure 1. The *information extraction* component uses natural language processing techniques to automatically extract the initial observations from the free text user input [12]. These observations are then combined with rules concerning trade fraud in the argumentation setting to build arguments for and against the claim "fraud". The *stability* component decides if any additional observations that the citizen could possibly add in the future can change the acceptability status of the "fraud" claim. If not, the dialogue terminates; otherwise a *question policy* component finds the best question to ask given current observations. The stability component is thus an important part of the agent's architecture: it provides a termination criterion that prevents the agent from asking unnecessary questions. If, for example, it is already clear from the initial observations that we are not dealing with fraud because the citizen simply received a product they did not like, the agent will not continue to exhaustively inquire [4] about further details of the situation.

The rest of this paper focuses on a more theoretical study of the stability component of the intake agent. Stability in structured argumentation is a form of dynamic argumentation that was introduced in [14]. Informally, a claim is stable if more information cannot change the acceptability of the claim, where this acceptability depends on the acceptability of arguments for this claim in terms of Dung's grounded semantics [7]. Detecting stability is complex: a brute-force approach would involve generating and evaluating all possible future argumentation setups given new observations, which would require far too much time in an applied setting. In this paper, we provide some new insights on the complexity of the stability problem by showing that it is CoNP-hard.

A sound approximation algorithm for stability was provided in [14]. However, the conditions under which the algorithm is complete were not studied in-depth. When investigating these conditions, we identified the nontrivial issues of *irrelevant labels* and *support cycles*, in the presence of which the algorithm from [14] does not detect a stable situation. In this paper, we solve these issues by proposing a new approximation algorithm consisting of an alternative labelling and a preprocessing step. We prove[4] that the refined algorithm has polynomial time complexity and that it is sound. Furthermore, we specify constraints on the input under which the new algorithm is complete.

Section 2 below specifies the structured argumentation setup for which we formally define the stability problem in Section 3. In Section 4 we then identify issues with the algorithm of [14], propose our refined algorithm and study its properties. Section 5 discusses related work in dynamic argumentation and Section 6 concludes the paper.

[3] https://nationaal-politielab.sites.uu.nl/estimating-stability-for-efficient-argument-based-inquiry/

[4] Due to space restrictions, proofs are omitted in this paper. The proofs are available at https://nationaal-politielab.sites.uu.nl/estimating-stability-for-efficient-argument-based-inquiry/

2. Preliminaries

Our argumentation setup is a variation on ASPIC$^+$ [11], albeit simplified in that we only consider axiom premises, defeasible rules and no preferences. From a theoretical perspective, this can be considered to be a limitation; however, from a practical point of view this simplification makes it more feasible for police employees without background in formal argumentation to adapt or create rule sets. We add the notion of queryable literals $\mathcal{Q}$. These literals can be obtained (i.e. added to the knowledge base) by querying the citizen, thus restricting the possibilities of updating the knowledge base.

Definition 1 (Argumentation Setup)**.** An **argumentation setup** AS is a tuple $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ where:

- $\mathcal{L}$ is a finite propositional language, closed under classical negation ($\neg$). The literals will be denoted by lower-case letters. We write $a = -b$ iff $a = \neg b$ or $b = \neg a$.
- $\mathcal{R}$ is a finite set of defeasible rules $a_1, \ldots, a_m \Rightarrow c$ such that $\{a_1, \ldots, a_m, c\} \subseteq \mathcal{L}$. Where $r \in \mathcal{R}$, $\texttt{ants}(r) = \{a_1, \ldots, a_m\}$ are the antecedents of r and $\texttt{cons}(r) = c$ is its consequent. We refer to a rule with consequent c as “a rule for c”.
- $\mathcal{Q} \subseteq \mathcal{L}$ is a set of queryable literals, s.t. $l \in \mathcal{Q}$ iff $-l \in \mathcal{Q}$.
- $\mathcal{K} \subseteq \mathcal{Q}$ is the knowledge base, which must be consistent: if $l \in \mathcal{K}$ then $-l \notin \mathcal{K}$.

Based on an argumentation setup, arguments can be constructed from $\mathcal{R}$ and $\mathcal{K}$.

Definition 2 (Argument)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. We denote by $Arg(AS)$ the set of **arguments** inferred from AS. An argument $A \in Arg(AS)$ is:

- an **observation-based argument** c iff $c \in \mathcal{K}$.
  The conclusion $\texttt{conc}(A)$ of A is c. The set of subarguments $\texttt{sub}(A)$ of A is $\{c\}$.
- a **rule-based argument** $A_1, \ldots, A_m \Rightarrow c$ iff for each $i \in [1 \mathinner{..} m]$: A_i is in $Arg(AS)$ with conclusion c_i and there is a rule $r : c_1, \ldots, c_m \Rightarrow c$ in $\mathcal{R}$.
  The conclusion $\texttt{conc}(A)$ of A is c. The set of subarguments $\texttt{sub}(A)$ of A is $\texttt{sub}(A_1) \cup \ldots \cup \texttt{sub}(A_m) \cup \{A\}$. The top rule $\texttt{top-rule}(A)$ of A is r.

We refer to an argument with conclusion c as “an argument for c”. We refer to a rule-based argument with top rule r as “an argument based on r”.

Definition 3 (Attack)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. For two arguments $A, B \in Arg(AS)$ we say that A **attacks** B on B' iff A’s conclusion is c, there is a subargument $B' \in \texttt{sub}(B)$ such that $\texttt{conc}(B') = -c$ and $-c \notin \mathcal{K}$.

Our definition of attack corresponds to rebuttal in ASPIC$^+$ [11]. From Definition 3 it follows directly that observation-based arguments cannot be attacked.

Example 1 (Online trade fraud)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$, visualised in Figure 2, be an argumentation setup in the domain of online trade fraud. $\mathcal{L}$ consists of the literals $\{b, sm, sp, rp, rm, u, s, t, sd, rd, d, f\}$ and their negations. Squares represent literals from $\mathcal{L}$, rounded squares are queryable literals (from $\mathcal{Q}$) and literals in $\mathcal{K}$ are shaded. Rules are represented by double-lined arrows and attacks as single-lined arrows. $Arg(AS)$ includes an argument for f based on the rule $sd, \neg rd, d \Rightarrow f$ and an argument for $\neg f$ based on $b, t \Rightarrow \neg f$. These arguments attack each other.

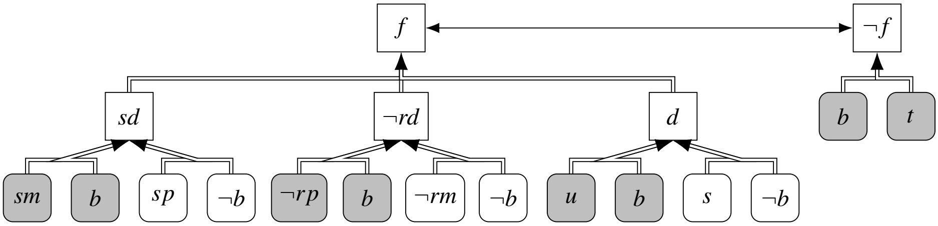


Figure 2. Example of an argumentation setup *AS* from the law enforcement domain. *b*: citizen tried to buy a product (as opposed to selling a product); *sm*: citizen sent money; *sp*: citizen sent product; *rp*: citizen received product; *rm*: citizen received money; *u*: suspicious url; *s*: screenshot of payment; *t*: trusted web shop; *sd*: citizen delivered; *rd*: citizen received delivery; *d*: deception; *f*: fraud. Note that the literals *b* and $\neg b$ are visualised multiple times and attacks between them are omitted for clarity.

Like in ASPIC$^+$, the evaluation of arguments is done using the semantics of [7]. We choose grounded semantics since it is the most skeptical semantics, which fits the application in police investigation. We subsequently use the grounded extension to define the acceptability of literals in an argumentation setup.

Definition 4 (Grounded Extension)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup and let $S \subseteq Arg(AS)$. S is said to be **conflict-free** iff there are no $A, B \in S$ such that A attacks B. S **defends** $A \in Arg(AS)$ iff for each $B \in Arg(AS)$ that attacks A there is a $C \in S$ that attacks B. S is **admissible** iff it is conflict-free and defends all its arguments. S is a **complete extension** iff it is admissible and contains all the arguments it defends. The **grounded extension** $G(AS)$ is the least (w.r.t. $\subseteq$) complete extension.

Definition 5 (Acceptability)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. The acceptability of literal $l \in \mathcal{L}$ given AS is:

- **unsatisfiable** iff there is no argument for l in $Arg(AS)$;
- **defended** iff there exists an argument for l in $Arg(AS)$ that is also in the grounded extension $G(AS)$;
- **out** iff there exists an argument for l in $Arg(AS)$, but each argument for l in $Arg(AS)$ is attacked by an argument in the grounded extension $G(AS)$;
- **blocked** iff there exists an argument for l in $Arg(AS)$, but no argument for l is in the grounded extension $G(AS)$ and at least one argument for l is not attacked by an argument in $G(AS)$.

Note that these acceptability statuses are complementary: e.g. if l is not unsatisfiable, defended or out, then it is blocked. This follows directly from the definition.

Example 2 (Example 1 continued)**.** In argumentation setup AS from Figure 2, $G(AS)$ contains (unattacked) arguments for sm, b, $\neg rp$, u, t, sd, $\neg rd$ and d, so these literals are defended in AS. There are arguments for f and $\neg f$ in $Arg(AS)$ that attack each other, but these are not attacked or defended by any argument in $G(AS)$, so f and $\neg f$ are blocked in AS. Each other literal $l \in \mathcal{L}$ is unsatisfiable in AS: there is no argument for l in $Arg(AS)$.

3. Stability

Using Definition 5, we can determine the acceptability status of a literal $l \in \mathcal{L}$ in a given argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$. However, by adding more information, l's acceptability status may change. Informally, l is *stable* in AS if its acceptability status cannot change by adding any combination of queryables to the knowledge base - provided that the resulting knowledge base is consistent. Note that we restrict the changes on the argumentation setup to adding knowledge, since we expect the citizen to attribute only facts on his/her situation. Next, we define future setups, which specify how information can be added to AS.

Definition 6 (Future setups)**.** The set of **future setups** $F(AS)$ of an argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ consists of all argumentation setups $AS' = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K}')$ with $\mathcal{K} \subseteq \mathcal{K}'$.

Note that the argumentation setup AS always belongs to the set of future setups $F(AS)$. Further recall from Definition 1 that $\mathcal{K}'$ must be consistent since AS' is an argumentation setup. Using the notion of future setups, we now define stability.

Definition 7 (Stability)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. A literal $l \in \mathcal{L}$ is **stable** in AS iff there is an acceptability status $acc \in \{$unsatisfiable, defended, out, blocked$\}$ such that for each $AS' \in F(AS)$, l is acc in AS'.

Example 3 (Example 2 continued)**.** In our running example, the literal f is stable. By querying the client agent, we could obtain more information; $F(AS)$ for example contains an argumentation setup with knowledge base $\mathcal{K}' = \mathcal{K} \cup \{\neg sp\} = \{sm, b, \neg rp, u, t, \neg sp\}$. However, adding information does not influence f's acceptability status: for each AS' in $F(AS)$, f is blocked in AS'. Therefore, f is stable in AS.

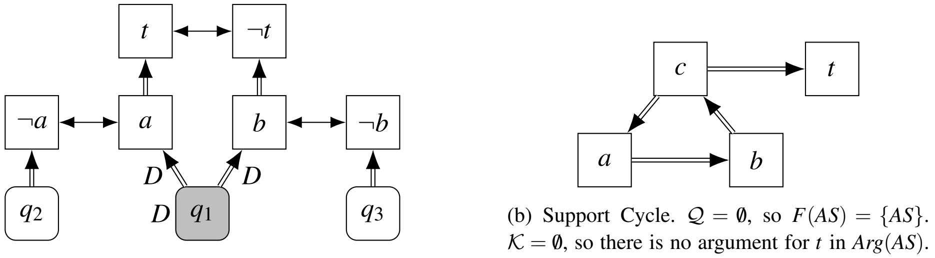
Proposition 1. *Determining stability is CoNP-hard.*

This can be shown by a polynomial-time reduction from the CoNP-complete problem UNSAT. The full proof is available on the website with additional material.

CoNP-hard problems are generally considered intractable (unless P = NP). Given the above results and assuming that P $\neq$ NP, there is no exact polynomial-time algorithm that determines for an arbitrary argumentation setup AS if a literal is stable in AS. This means that an exact algorithm would need exponential time. Since practical applications require fast computation for arbitrary argumentation setups, we consider a sound polynomial-time approximation algorithm in the next section.

4. Approximating stability

A first approximation algorithm for determining stability in formal argumentation was proposed in [14]. This algorithm assigns a label to literals and rules that it considers to be stable. Each label relates to one of the four cases of stability: U (unsatisfiable); D (defended); O (out); or B (blocked). However, the algorithm is not complete: there exist argumentation setups that are stable but are not labelled as such by the approximation algorithm. In [14] we gave an example, but no precise specification of argumentation setups for which the algorithm does not recognise stability. In the next subsection, we give two additional examples which reveal different issues of the method described in [14]. In Sections 4.2 and 4.3, we present a refined algorithm to solve these issues. Subsequently, we will show soundness and conditional completeness and study the computational complexity of this refinement in Section 4.4.



(a) D/B is irrelevant in [14] *Case B lit. A*.

(b) Support Cycle. $\mathcal{Q} = \emptyset$, so $F(AS) = \{AS\}$. $\mathcal{K} = \emptyset$, so there is no argument for t in $Arg(AS)$. However, L does not label t.

Figure 3. Examples of incompleteness of the basic algorithm from [14].

4.1. Examples of incompleteness basic algorithm

Figures 3a and 3b illustrate two different issues of the algorithm from [14].

Example 4 (Irrelevant label problem)**.** Figure 3a represents an argumentation setup AS in which q_1, q_2 and q_3 are queryable. q_1 is in the knowledge base. There is an argument for t based on $a \Rightarrow t$ and an argument for $\neg t$ based on $b \Rightarrow \neg t$ in $Arg(AS)$. So for each $AS' \in F(AS)$, t is blocked in AS'. However, t is not recognised as being stable by the algorithm in [14]. The literal q_1 and rules $q_1 \Rightarrow a$ and $q_1 \Rightarrow b$ are correctly labelled D, but the other literals and rules are not labelled by the algorithm. a and b are not labelled because they may become either defended (if $\neg q_2$ resp. $\neg q_3 \in \mathcal{K}'$) or blocked (if q_2 resp. $q_3 \in \mathcal{K}'$). As a result, the rules $a \Rightarrow t$ and $b \Rightarrow \neg t$ are not labelled because they may become either defended or blocked. In all future setups in $F(AS)$, the argument for a is either defended (if $q_2 \notin \mathcal{K}'$) or blocked (if $q_2 \in \mathcal{K}'$). Similarly, in every future setup, the argument for b is either defended (if $q_3 \notin \mathcal{K}'$) or blocked (if $q_3 \in \mathcal{K}'$). The algorithm in [14] has a labelling rule *Case B literal A* stating that "$l \in \mathcal{L}$ is labelled B iff $l \in \mathcal{Q}$ and a rule for l and a rule for $-l$ are labelled D or B". However, this rule does not apply: although $a \Rightarrow t$ and $b \Rightarrow \neg t$ will be labelled D or B in a future setup in which we have information about q_2 and q_3, we do not know the exact label - which is here irrelevant.

We will refer to the issue illustrated in Figure 3a as the *irrelevant label problem*. It is caused by the fact that L only assigns a label if there is exactly one possible acceptance status for all future setups, but does not take into account that some acceptability statuses are *impossible* in a future setup. The next example reveals another issue of the basic algorithm, which we will refer to as the *support cycle problem*.

Example 5 (Support cycle problem)**.** Figure 3b represents an argumentation setup AS in which a, b, c and t are literals that are not queryable. As a result, there is no other future argumentation setup than the current setup: $F(AS) = \{AS\}$. There is no argument for t in $Arg(AS)$, hence t is unsatisfiable for every $AS' \in F(AS)$. However, no rule or literal is labelled U since the algorithm in [14] only labels a non-queryable literal U if all rules for this literal are labelled U; a rule only gets labelled U if at least one antecedent of that rule is labelled U. Because of this support cycle, there is no place to start labelling.

Due to the irrelevant label problem and the support cycle problem, the algorithm from [14] fails to recognise the stability of some argumentation setups. We present a solution to these problems in Sections 4.2 and 4.3.

4.2. Reasoning with possible future labels

In this section, we present an alternative labelling method that bypasses the irrelevant label problem by reasoning with possible future labels. Whereas the approximation algorithm presented in [14] relies on a partial labelling function L that assigns at most one label to each literal in $\mathcal{L}$ and rule in $\mathcal{R}$ ($L : \mathcal{L} \cup \mathcal{R} \nrightarrow \{U, D, O, B\}$ where $\nrightarrow$ denotes a partial function), we propose a labelling L' that assigns a quadruple of four booleans $\langle u, d, o, b \rangle$ to each literal and rule. Each boolean corresponds to an acceptability status. Intuitively, the truth value of a boolean belonging to a literal or rule represents the possibility that this literal or rule may become unsatisfiable (u), defended (d), out (o) or blocked (b) in a future argumentation setup. Similar to the approach in [14], labels of rules depend on the labels of their antecedent literals and labels of literals depend on the labels of rules for that literal. Literals and rules are labelled incrementally, starting from queryable literals and literals for which there is no rule and relabelling literals and rules based on the resulting new labels, until no new label can be added.

Definition 8 (Quadruple labelling L')**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. The **labelling function** $L' : \mathcal{L} \cup \mathcal{R} \rightarrow \{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1\}$ assigns a label $\langle u, d, o, b \rangle$ to each literal or rule in $\mathcal{L} \cup \mathcal{R}$. Given a literal or rule $x \in \mathcal{L} \cup \mathcal{R}$, we write $\neg u(x)$ [resp. $\neg d(x), \neg o(x), \neg b(x)$] iff the u- [resp. d-, o-, b-] boolean of x's label is False and $u(x)$ [resp. $d(x), o(x), b(x)$] iff the u- [resp. d-, o-, b-] boolean of x's label is True. We say that a rule or literal x is **labelled stable** by L' iff exactly one of the booleans is True: $L'(x)$ is $\langle 1,0,0,0 \rangle$, $\langle 0,1,0,0 \rangle$, $\langle 0,0,1,0 \rangle$ or $\langle 0,0,0,1 \rangle$.
Given a literal $l \in \mathcal{L}$, $L'(l) = \langle u, d, o, b \rangle$ where:

literal cannot become unsatisfiable: $\neg u(l)$ iff:
L-U-a) $l \in \mathcal{K}$; or
L-U-b) there is a rule r for l with $\neg u(r)$.

literal cannot become defended: $\neg d(l)$ iff:
L-D-a) $-l \in \mathcal{K}$; or
L-D-b) $l \notin \mathcal{Q}$ and for each rule r for l: $\neg d(r)$; or
L-D-c) $l \notin \mathcal{Q}$ and there is a rule r' for $-l$ with $\neg u(r')$ and $\neg o(r')$.

literal cannot become out: $\neg o(l)$ iff:
L-O-a) $l \in \mathcal{K}$; or
L-O-b) for each rule r for l: $\neg d(r)$, $\neg o(r)$ and $\neg b(r)$; or
L-O-c) $l \notin \mathcal{Q}$ and for each rule r for l: $\neg o(r)$; or
L-O-d) $l \notin \mathcal{Q}$ and there is a rule r for l with $\neg u(r)$ and $\neg o(r)$.

literal cannot become blocked: $\neg b(l)$ iff:
L-B-a) $l \in \mathcal{Q}$; or
L-B-b) for each rule r for l: $\neg d(r)$ and $\neg b(r)$; or
L-B-c) for each rule r for l: $\neg b(r)$ and for each rule r' for $-l$: $\neg d(r')$ and $\neg b(r')$.
L-B-d) there is a rule r for l with $\neg u(r)$, $\neg o(r)$ and $\neg b(r)$ and for each rule r' for $-l$: $\neg d(r')$ and $\neg b(r')$.

Given a rule $r \in \mathcal{R}$, $L'(r) = \langle u, d, o, b \rangle$ where:

rule cannot become unsatisfiable: $\neg u(r)$ iff:
R-U-a) for each antecedent l of r: $\neg u(l)$.

rule cannot become defended: $\neg d(r)$ iff:
R-D-a) there is an antecedent l of r with $\neg d(l)$.

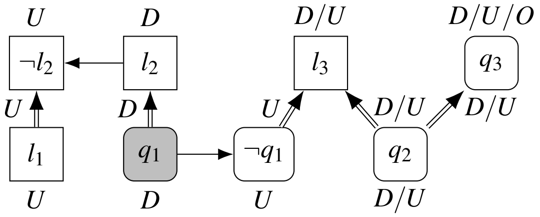


Figure 4. Quadruple labelling example.

rule cannot become out: $\neg o(r)$ iff:
R-O-a) for each antecedent l of r: $\neg o(l)$; or
R-O-b) there is an antecedent l of r with $\neg d(l)$ and $\neg o(l)$ and $\neg b(l)$.
rule cannot become blocked: $\neg b(r)$ iff:
R-B-a) for each antecedent l of r: $\neg b(l)$; or
R-B-b) there is an antecedent l of r with $\neg d(l)$ and $\neg b(l)$.

Example 6. We give some intuition by labelling the AS from Figure 4. Some rules apply if (the negation of) a literal is in $\mathcal{K}$ or $\mathcal{Q}$, e.g. q_1 is labelled $\langle 0,1,0,0\rangle$ by Definition 8 case L-U-a, L-O-b and L-B-a: there is an observation-based argument for q_1 that cannot be attacked in any future setup. The absence of rules for a literal is informative for the acceptability status as well: e.g. l_1 is labelled $\langle 1,0,0,0\rangle$ by L-D-b, L-O-b/c and L-B-b/c.

Other labels are based on the rules for (the negation of) a literal and propagate properties of (attacks on) subarguments. For example, $q_1 \Rightarrow l_2$ is labelled $\langle 0,1,0,0\rangle$ by R-U-a, R-O-a and R-B-a and l_2 is labelled $\langle 0,1,0,0\rangle$ by L-U-b, L-O-c/d and L-B-c/d. Some literals and rules cannot be labelled stable, but still some acceptability status(es) can be excluded: e.g. the rule $q_2 \Rightarrow q_3$ is labelled $\langle 1,1,0,0\rangle$ by case R-O-a and R-B-a.

Example 7 (Alternative labelling Figure 3a)**.** Consider the L' labelling for the argumentation setup from Figure 3a. q_1 is in the knowledge base, so by Definition 8, $L'(q_1) = \langle 0,1,0,0\rangle$. Then $L'(q_1 \Rightarrow a) = L'(q_1 \Rightarrow b) = \langle 0,1,0,0\rangle$ by R-U-a, R-O-a and R-B-a. q_2 and q_3 are queryable but not in the knowledge base and there are no rules for q_2 or q_3, so by Case L-O-b and L-B-a: $L'(q_2) = L'(q_3) = \langle 1,1,0,0\rangle$. For the rules $q_2 \Rightarrow \neg a$ and $q_3 \Rightarrow \neg b$, only the d- and u-booleans are True by R-O-a and R-B-a. As a result, for the literals a and b only the d- and b-booleans are True by L-U-b and L-O-c, which implies by R-U-a and R-O-a that $L'(b \Rightarrow t) = L'(a \Rightarrow t) = \langle 0,1,0,1\rangle$. Finally, t is labelled $L'(t) = \langle 0,0,0,1\rangle$ (by L-U-b, L-D-c and L-O-c/d), so t is labelled stable by L'.

In Example 7 we saw that t is labelled stable by our labelling function L', but its stability was not detected by [14]'s labelling function L. In general, each literal or rule labelled stable by L is also labelled stable by L', but L' covers more stable setups than L.

4.3. Preprocessing

The new labelling proposed in the previous section does not solve the support cycle problem: if we would apply the labelling L' from Definition 8 to the argumentation setup from Figure 3b, all literals l (including literal t) would be labelled $\langle 1,1,1,1\rangle$. In order to solve this issue, we add a preprocessing step, which is specified in Algorithm 1. The

Algorithm 1 Preprocessing step

```
1: procedure PREPROCESS($\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K}$)
2:     Label each literal $l$ s.t. $l \in \mathcal{Q} \wedge -l \notin \mathcal{K}$ as $\langle 1,1,1,1 \rangle$
3:     Label all other literals as $\langle 1,0,0,0 \rangle$
4:     Label each $r \in \mathcal{R}$ as $\langle 1,0,0,0 \rangle$
5:     while a label changed in the previous loop do
6:         for Rule $r$ in $\mathcal{R}$ do
7:             if $L(r) = \langle 1,0,0,0 \rangle$ and for each $l \in \mathtt{ants}(r)$: $L(l) \neq \langle 1,0,0,0 \rangle$ then
8:                 Label $r$ as $\langle 1,1,1,1 \rangle$
9:                 Label $\mathtt{cons}(r)$ as $\langle 1,1,1,1 \rangle$
```

idea of this algorithm is that initially, all literals that cannot be in the knowledge base in a future setup and all rules are labelled $\langle 1,0,0,0 \rangle$ (i.e. unsatisfiable). Then, the algorithm incrementally removes unsatisfiable labels of rules for which all antecedents are not labelled $\langle 1,0,0,0 \rangle$, and of the consequents of these rules, based on the intuition that there may be an argument based on these rules in a future setup.

Example 8 (Alternative labelling Example 5)**.** We reconsider Figure 3b, assuming that the preprocessing step has been executed. In Line 3, all literals (a, b, c and t) are labelled $\langle 1,0,0,0 \rangle$. Since the if-statement in Line 7 never returns true, no rule or literal gets another label, so the while loop is executed only once. After termination of Algorithm 1, all literals are still (correctly) labelled $\langle 1,0,0,0 \rangle$.

4.4. Properties of the proposed algorithm

In this subsection, we present properties of STABILITY, our proposed algorithm, which runs PREPROCESS on the argumentation setup and then labels all literals and rules by repeatedly applying Definition 8. First, we consider STABILITY's soundness.

Proposition 2 (Soundness stability labelling)**.** *Given an argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ and labelling L' after executing the* STABILITY *algorithm, if a literal $l \in \mathcal{L}$ is labelled stable in AS, then l is stable in AS.*

Soundness can be proven by systematically analysing all argumentation setups in which a literal l is labelled stable (e.g. $l \in \mathcal{K}$ or [$l \notin \mathcal{Q}$ and there is no rule for l in $\mathcal{R}$]) and proving that l is stable in each of them. Next, we consider completeness. As illustrated in Example 9, STABILITY is not complete for all argumentation setups.

Example 9 (Example 3 continued)**.** Consider the argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ where $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{Q}$ are as in Figure 2, but $\mathcal{K} = \{\neg sm, rm\}$. STABILITY does not label f stable: it expects a future argument for f based on $sd, \neg rd, d \Rightarrow f$, where the argument for sd is based on $sp, \neg b \Rightarrow sd$ and the argument for $\neg rd$ is based on $\neg rp, b \Rightarrow rd$. However, this argument would require both b and $\neg b$ to be in the knowledge base, which violates the consistency criterion. In fact, for each AS' in $F(AS)$ there is no argument for f in $Arg(AS')$, so f should be labelled $\langle 1,0,0,0 \rangle$.

Example 9 shows that there are argumentation setups where the STABILITY algorithm wrongfully takes the possibility into account that there exists an argument for a literal in a future argumentation setup. Specifically, this issue is caused by *inconsistent potential arguments*, which we define next.

Definition 9 (Potential argument)**.** Let $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ be an argumentation setup. A **potential argument** A^p inferred from AS is:

- c iff $c \in \mathcal{Q}$ and $-c \notin \mathcal{K}$. $\texttt{prem}(A^p) = \{c\}$; $\texttt{conc}(A^p) = c$; and $\texttt{sub}(A^p) = \{c\}$.
- $A_1, \ldots, A_m \Rightarrow c$ iff there is a rule $c_1, \ldots, c_m \Rightarrow c$ in $\mathcal{R}$ and for each $i \in [1 \mathinner{..} m]$: A_i is a potential argument inferred from AS and $\texttt{conc}(A_i) = c_i$. $\texttt{prem}(A^p) = \texttt{prem}(A_1) \cup \ldots \cup \texttt{prem}(A_m)$; $\texttt{conc}(A^p) = c$; and $\texttt{sub}(A^p) = \texttt{sub}(A_1) \cup \ldots \cup \texttt{sub}(A_m) \cup \{A\}$.

We denote the set of potential arguments by $P(AS)$. Given some $A^p, B^p \in P(AS)$, A^p **p-attacks** B^p iff there is a $B' \in \texttt{sub}(B^p)$ s.t. $\texttt{conc}(A^p) = -\texttt{conc}(B')$ and $-\texttt{conc}(B') \notin \mathcal{K}$; A^p is **inconsistent** with B^p iff $\{a, -a\} \in \texttt{prem}(A^p) \cup \texttt{prem}(B^p)$ for some $a \in \mathcal{L}$.

Note that, for any argumentation setup AS, each argument inferred from AS or some future setup is a potential argument in $P(AS)$. However, there may be a potential argument $A^p \in P(AS)$ such that there is no $AS' \in F(AS)$ with $A^p \in Arg(AS')$, but then A^p must be inconsistent *with itself*, like in Example 9. Example 10 reveals another issue, where stability is not detected due to an inconsistency of two *different* potential arguments.

Example 10 (Mutual inconsistency issues)**.** Given the argumentation setup AS illustrated in Figure 5, for each $AS' = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K}')$ in $F(AS)$, l_1 is blocked in AS': if $\neg q_2 \notin \mathcal{K}'$ then there is an argument for l_1 based on $q_2 \Rightarrow l_1$; otherwise there is an argument for l_1 based on $\neg q_2 \Rightarrow l_1$. However, l_1 is not labelled $\langle 0,0,0,1 \rangle$ because STABILITY wrongfully anticipates a future setup AS' in which each argument for l_1 is attacked by an argument in $G(AS')$ (thus $o(l_1)$). For the same reason, $\neg l_1$ is labelled $d(\neg l_1)$ and therefore $L'(\neg l_1) \neq \langle 0,0,0,1 \rangle$, although $\neg l_1$ is blocked in each $AS' \in F(AS)$.

The issues illustrated in Examples 9 and 10 can be generalised to the following two situations. Given an argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ and a literal $l \in \mathcal{L}$:

- l is **inconsistently supported** in AS iff there are $A^p, B^p \in P(AS)$ such that $\texttt{conc}(A^p) = \texttt{conc}(B^p) = l$ and A^p is inconsistent with B^p.
- l is **inconsistently attacked** in AS iff there is a $C^p \in P(AS)$ such that $\texttt{conc}(C^p) = l$ and there are $A^p, B^p \in P(AS)$ such that A^p p-attacks C^p, B^p p-attacks C^p and A^p is inconsistent with B^p.

If a potential argument is inconsistent with itself, its conclusion l can be incorrectly labelled $d(l)$ or $b(l)$ (e.g. f in Example 9). Similarly, if two potential arguments with the same conclusion are inconsistent, their conclusion l can be incorrectly labelled $o(l)$ (e.g. l_1 in Example 10). Moreover, if l is inconsistently attacked, l may be incorrectly labelled $d(l)$ (e.g. $\neg l_1$ in Example 10) or $b(l)$. Otherwise, l is labelled stable if it is stable in AS.

Proposition 3 (Conditional completeness stability labelling)**.** *Given an argumentation setup $AS = (\mathcal{L}, \mathcal{R}, \mathcal{Q}, \mathcal{K})$ and a labelling L' after executing* STABILITY*. If $l \in \mathcal{L}$ is stable in AS and l is not inconsistently supported or attacked, then l is labelled stable by L'.*

Finally, the proposed algorithm runs in polynomial time, which makes it suitable for practical applications such as human-computer inquiry dialogues.

Proposition 4. *The time complexity of* STABILITY *is $\mathcal{O}(|\mathcal{L}|^2 \cdot |\mathcal{R}| + |\mathcal{L}| \cdot |\mathcal{R}|^2)$.*

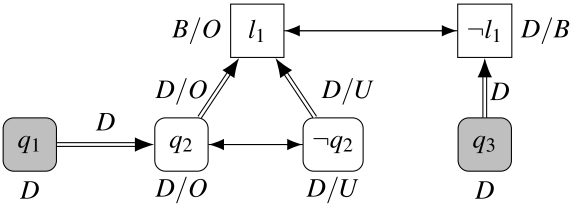


Figure 5. For each $AS' \in F(AS)$: l_1 and $\neg l_1$ are blocked in AS', but not labelled as such due to inconsistent support (l_1) and attack ($\neg l_1$).

5. Related work

We study stability [14]: given a specific structured argumentation setting, can adding information change the acceptability status of some propositional formula? This is a relatively new task within dynamic argumentation, which studies the acceptability of (sets of) arguments or their conclusions in relation to changes on the argumentation framework. Research on dynamic argumentation includes work on the impact of a change operation [5,1], enforcement [2,6], resolution [10] and the relation with belief revision [8,13].

Most research on dynamic argumentation, e.g. [2,5,6], only considers the effect of changes in the *abstract* framework, such as adding an argument. This approach does not take into account dependencies between arguments. For example, adding a new argument A often introduces more arguments having A as a subargument. Conversely, we study the effect of changes in the underlying *structured* argumentation framework.

None of the existing work in structured dynamic argumentation specifically studies stability. We briefly discuss some related research. [10] study resolutions in structured argumentation: they show how the acceptability of *arguments* changes due to a change of *preferences* in the underlying structured argumentation framework. Another related study [13] shows how the acceptability of a specific set of *arguments* can be altered by a minimal number of changes on the premises and/or rules in an argumentation framework, relating dynamic argumentation to belief revision. However, they do not focus on a specific task, such as stability; they do not consider *computational complexity* or provide an efficient (approximation) algorithm. One of the few papers that take computational complexity into account is [1]. The authors propose an efficient algorithm to minimise re-computations after a change in a DeLP program. However, whereas they study acceptability status after a *specific change*, we study the status after *any possible change* in the structured argumentation framework.

Finally, [9] apply a similar strategy to ours for efficiently determining the acceptability status of literals: they create a graph representing (the relation between) literals and rules and incrementally label its nodes and edges. However, they only determine the *current* acceptability status without considering changes.

6. Discussion and conclusion

We have studied the task of detecting stability: given a specific structured argumentation setup, based on a variation on ASPIC$^+$, can adding information result in a changed acceptability status of a specific literal? We have shown that the task is CoNP-hard. This is problematic in practical applications, such as identifying the termination criterion in human-computer inquiry dialogue. We proposed an algorithm for estimating stability that improves on the algorithm in [14]. We have shown that the refined algorithm is sound and runs in polynomial time. Thanks to these properties, the algorithm has been taken into use as part of an agent handling intake of fraud reports at the Dutch National Police – we provide an English demo that also visualises the agent's stability component.

There are examples of argumentation setups for which the algorithm does not detect that a literal is stable; in our application, this can result in the agent asking unnecessary questions. This issue could be resolved by a further refinement of the algorithm, which lists the knowledge bases $\mathcal{K}'$ of all future setups and checks that each $\mathcal{K}'$ is consistent. However, such an algorithm would have exponential time complexity.

In future work, we plan to extend the argumentation framework and the allowed updates. Furthermore, our demo applies a heuristic to select relevant questions; we plan to specify this formally. Finally, we will extensively evaluate the fraud intake agent.

References

[1] Gianvincenzo Alfano, Sergio Greco, Francesco Parisi, Gerardo Ignacio Simari, and Guillermo Ricardo Simari. An incremental approach to structured argumentation over dynamic knowledge bases. In *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning*, pages 78–87, 2018.

[2] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Proceedings of the 2010 conference on Computational Models of Argument: Proceedings of COMMA 2010*, pages 75–86. IOS Press, 2010.

[3] Floris Bex, Joeri Peters, and Bas Testerink. AI for online criminal complaints: From natural dialogues to structured scenarios. In *Artificial Intelligence for Justice Workshop (ECAI 2016)*, pages 22–29, 2016.

[4] Elizabeth Black and Anthony Hunter. An inquiry dialogue system. *Autonomous Agents and Multiagent Systems*, 19(2):173–209, 2009.

[5] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, 38:49–84, 2010.

[6] Sylvie Doutre and Jean-Guy Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument and Computation*, 9:223–248, 11 2018.

[7] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[8] Marcelo Alejandro Falappa, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. Belief revision and argumentation theory. *Argumentation in Artificial Intelligence*, pages 341–360, 2009.

[9] Abdelraouf Hecham, Pierre Bisquert, and Madalina Croitoru. On a flexible representation for defeasible reasoning variants. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 1123–1131, 2018.

[10] Sanjay Modgil and Henry Prakken. Resolutions in structured argumentation. In *Proceedings of the 4th International Conference on Computational Models of Argument*, pages 310–321, 2012.

[11] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[12] Marijn Schraagen and Floris Bex. Extraction of semantic relations in noisy user-generated law enforcement data. In *13th International Conference on Semantic Computing*, pages 79–86. IEEE, 2019.

[13] Mark Snaith and Chris Reed. Argument revision. *Journal of Logic and Computation*, 27(7):2089–2134, 2016.

[14] Bas Testerink, Daphne Odekerken, and Floris Bex. A method for efficient argument-based inquiry. In *Proceedings of the 13th International Conference on Flexible Query Answering Systems*, 2019.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200515

Argumentative Relation Classification with Background Knowledge

Debjit PAUL [a], Juri OPITZ [a], Maria BECKER [a] Jonathan KOBBE [b] Graeme HIRST [c] Anette FRANK [a]

[a] *Department of Computational Linguistics, Heidelberg University*
[b] *Data and Web Science, University of Mannheim*
[c] *Department of Computer Science, University of Toronto*

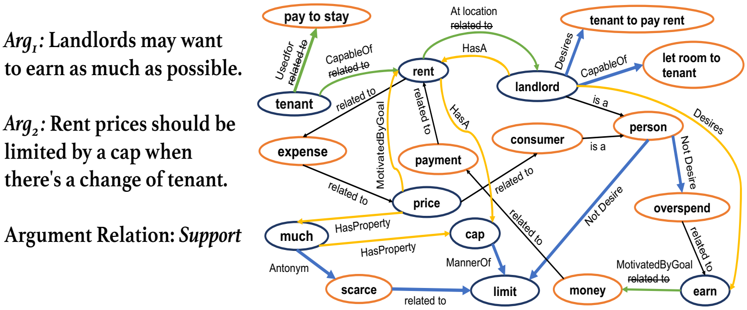
Abstract. A common conception is that the understanding of relations that hold between argument units requires knowledge beyond the text. But to date, argument analysis systems that leverage knowledge resources are still very rare. In this paper, we propose an unsupervised graph-based ranking method that extracts relevant multi-hop knowledge from a background knowledge resource. This knowledge is integrated into a neural argumentative relation classifier via an attention-based gating mechanism. In contrast to prior work we emphasize the selection of *relevant* multi-hop knowledge, and apply methods to *automatically enrich* the knowledge resource with missing knowledge. We assess model performance on two datasets, showing considerable improvement over strong baselines.

Keywords. argumentative relation classification, commonsense knowledge relations, multihop knowledge paths, knowledge graph completion, graph-based ranking

1. Introduction

Automatically identifying relations between argumentative text units (e.g., *support* and *attack* relations) has attracted much attention [1,2,3,4]. *Argumentative relation classification* (henceforth *ARC*) is the task of determining the type of relation that holds between two argumentative units (AUs, for short). This task has some overlap with *stance detection*, but differs in important aspects: while stance detection aims at determining the relation of AUs *towards a topic* or conclusion, argumentative relation classification analyzes relations *between argumentative units*. In this work we consider both *argument-topic relations* and *argument-argument relations* – since only a system that captures both types of relations can be applied in a real debate. We propose a ranking-based knowledge-knowledge-enhanced argumentative relation classification approach that we successfully apply to both (closely related) argumentative relation classification tasks.

Defining abstract semantic patterns is one way to explain argumentative relations [5]. In Fig. 1 Arg_1 implies that x is ***good for*** landlords, while Arg_2 implies that x is ***bad for*** tenants, with x = *'rise in price'*. This pattern can indicate *attack*. But Arg_2 states that *x should be limited* and thus the correct relation is *support* (Arg_1,Arg_2). Hence, we not only need good analysis of the text, but also further, so-called commonsense knowledge about the events, entities and relations mentioned in it, in order to gain true understanding



Arg$_1$: Landlords may want to earn as much as possible.

Arg$_2$: Rent prices should be limited by a cap when there's a change of tenant.

Argument Relation: *Support*

Figure 1. A subgraph extracted from ConceptNet. Blue edges portray relevant knowledge paths from ConceptNet. Concepts from the text in blue; intermediate nodes in orange. Yellow color edges: Our *on-the-fly* knowledge-base completion method infers ConceptNet relations. Green color edges: The knowledge-base completion feature of our method replaces *'related to'* relations with more specific ConceptNet relations.

of an argument. For example, we need to know that *landlords* and *tenants* are in a relation where one pays the other, with conflicting interests in the amount to be paid.

In this work we propose to leverage commonsense knowledge from ConceptNet [6] in order to connect pairs of concepts in argumentative units with implicit background knowledge relations. Fig. 1 shows a semantic (sub)graph with nodes representing concepts and edges (e.g., *'not desire'*) indicating relations between them. The graph captures semantic relations between entities (*tenant – landlord*) and properties (*much – limited*).

Our hypothesis is that capturing commonsense knowledge relations within and between AUs is essential for deeper understanding of arguments, especially for aspects of practical reasoning, cf. [7]. We investigate this hypothesis by devising a system that constructs subgraphs over pairs of AUs based on relevant concepts and multi-hop knowledge from the ConceptNet graph [6]. We propose a graph-based ranking method to extract relevant paths from these subgraphs that connect the argumentative units. Further, we dynamically enrich these graphs to counter sparsity problems when analyzing texts. Finally, we leverage knowledge from WordNet definitions to expand the meaning of words. E.g., a *tenant "... pays rent to use ... a building ... that is owned by someone else."*

Our contributions are: (i) We show that our graph-based method that extracts relevant commonsense knowledge and selectively integrates it into the model improves over a strong neural *and* a linear argumentative relation classification system on two datasets with different relation types; (ii) we show that enriching knowledge resources 'on the fly' can further improve results; and (iii) we provide an enhanced dataset for *support/attack* classification derived from Debatepedia. Our code and datasets will be made public.

2. Related Work

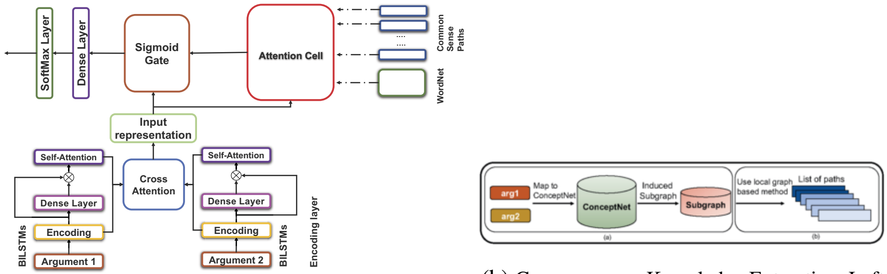
Argumentative Relation Classification (ARC) has been addressed in various works: [3] identify argument component types (*premise*, *claim*, *major-claim*) and argumentative relations (*support*, *non-support*) using structural, lexical and syntactic features using production rules, similar to [8]. Their system is extended by [9], who exploit the context of argumentative statements. [10] use a joint approach that, given a pre-segmented text, reconstructs the argument structure. This includes identifying the argumentative role (*pro* or *opposing*) of each segment and the argumentative function of each relation (*support* or *attack*). [11] propose the first end-to-end approach to solve argument component and relation identification, comparing a joint model to a pipeline system. [4] propose an end-to-end approach for argument structure reconstruction. Similar to [10], they predict whether there is an argument relation between AU pairs and whether it is *support* or *attack*. While they predict relations jointly with the argument component type, they predict the relation label independently. We also classify relations between AUs into *support* and *attack*. However, [12] show that systems applied to this task tend to focus on discourse clues instead of the content and can be easily fooled when relying on discourse indicators. We therefore adopt an experimental setting that focuses only on the content of AUs.

Recent approaches to argumentative relation classification (ARC) have been built on Siamese networks [13,14,15]. In our work we devise a strong neural system with self- and cross-attention as a novel baseline for ARC. But in contrast to previous work, we leverage commonsense knowledge for ARC and extend this system with a mechanism to inject full-fledged knowledge paths that we select from a background knowledge graph.

Background Knowledge for Argumentation When humans are debating (*Should rent prices be capped?*), they make use of background knowledge. Often, this knowledge belongs to the "content [that] is not expressed explicitly but resides in the mind of communicator and audience" [16,17]. Yet, few approaches have tried to leverage such knowledge in computational argumentation models, especially when it comes to commonsense knowledge (CSK). Previously, [14] investigate the impact of CSK in argumentative relation classification using linguistic and knowledge graph features derived from DBpedia and ConceptNet. They connect AUs via the knowledge graph, they use quantitative features that they derive from the established knowledge paths (edges only, i.e., deprived from concepts) to predict the argumentative relation between them. They extract a huge number of connecting paths, which they aggregate to patterns of relation types occurring in them. While [14] use only (features over) isolated relations (edges) that connect pairs of AUs, without filtering them by relevance, our work will filter and weight the knowledge paths and will include concepts (nodes) on the paths.

Besides CSK, other knowledge sources have been leveraged for argumentation. For example, Wikipedia articles [18], SNLI data [19] or sentiment lexica [20] have proven to be effective. [21] shows that the *Generative Lexicon* [22] captures relevant commonsense knowledge for argument mining in its qualia roles, such as physical or telic properties. However, such lexicons are hard to create and existing resources are little. [23] derive embeddings for FrameNet frames and entities from Wikidata to solve the Argument Reasoning Comprehension task [24]. They find small improvements from adding this knowledge and conclude that external knowledge alone is insufficient for improving argumentative reasoning. We are solving a related but different task and use different resources for injecting *commonsense knowledge*. Most importantly, while [23] integrate pre-trained embeddings computed over FrameNet and WikiData graphs at the token level, we pursue targeted knowledge selection from ConceptNet by inducing knowledge subgraphs between AUs that we extract from case-specific multi-hop knowledge paths using graph-based ranking.

Our goal is to take a step beyond the prior work by (i) studying how *relevant* knowledge can be selected that is tailor-cut to solving the relation classification task, by (ii) refining the extracted knowledge and leveraging an in-depth encoding of the paths, and finally by (iii) efficiently integrating this knowledge in a strong ARC approach.



(a) **ARK:** Argumentative relation classification (ARC) with self-attention and knowledge (ARK)

(b) Commonsense Knowledge Extraction. Left: Subgraph Construction. Right: Ranking & Path Selection

Figure 2. (a) ARC model with knowledge (ARK) and (b) Commonsense Knowledge Extraction

3. Argumentative Relation Classification with Commonsense Knowledge

We propose a neural Argumentative Relation Classification (ARC) system that (i) encodes pairs of argumentative units (AUs) using a cross-sentence attention mechanism over attentive BiLSTM encoders to understand their contextual features and structures; (ii) we leverage commonsense knowledge by linking concepts from the AUs to concepts from ConceptNet, and construct instance-specific subgraphs from which we extract relevant knowledge paths using graph-based ranking methods; finally (iii), we incorporate lexical knowledge from WordNet – *Synonyms* and definitions – to expand the meaning of terms in the AUs. Recently, [25] and [26] proposed methods to select multi-hop knowledge paths for reading comprehension and human needs classification: the former use heuristics, the latter graph-based measures for selection. In our work, we construct a knowledge subgraph over AUs and use local graph measures to select relevant knowledge for predicting the correct argumentative relation class. The selected knowledge paths along with *Synonyms* and definitional knowledge are encoded and incorporated into the relation prediction component. We use an attention cell that jointly encodes the encoded argument pair representations and the selected knowledge paths to predict implicit knowledge relations during inference. Figure 2 gives an overview of the model.

3.1. Argumentative Relation Classifier

The core of our model consists of three components: (1) encoding layer, (2) attention layer with *self-attention and cross-attention*, (3) output layer. The BiLSTM encoder takes two AUs $arg \in [arg1, arg2]$ as inputs: sequences of tokens $w_1^{arg}, ..., w_n^{arg}$ (or $w_{1:n}^{arg}$).

Encoding Layer We map the sequence of tokens of both AUs to sequences of word representations using word embeddings, and encode them with a single-layer BiLSTM.[1]

Attention Layer We apply ***self-attention*** to capture the contribution of each token in the argument [27]. We obtain argument representations x^{arg1} and x^{arg2} by taking the weighted sum of the attention scores and the hidden states that were generated by the BiLSTM.

We capture the relevance of the hidden representations of the arguments with ***cross-attention***. We calculate soft attention weights, this time across arguments and taking into account the self-attention weighted token representations from (1) and (2):

[1] The final state of the forward and backward pass is composed by taking the max over each dimension.

$$\hat{h}_i^{arg1} = \frac{\sigma(x_i^{arg2} h_i^{arg1})}{\sum_{j=1}^{N} \sigma(x_j^{arg2} h_j^{arg1})} \qquad \hat{h}_i^{arg2} = \frac{\sigma(x_i^{arg1} h_i^{arg2})}{\sum_{j=1}^{M} \sigma(x_j^{arg1} h_j^{arg2})} \tag{1, 2}$$

$$x_i^{arg1} = \sum_{j=1}^{N} \hat{h}_j^{arg1} h_i^{arg1}; \quad x_i^{arg2} = \sum_{j=1}^{M} \hat{h}_j^{arg2} h_i^{arg2} \tag{3}$$

with N, M the number of tokens in $arg1$ and $arg2$.

Output Layer We apply a final dense layer followed by softmax to predict the classes *support* or *attack*. As input y_i to this final layer we concatenate the output representations x_i^{arg1} and x_i^{arg2} from the cross-attention layer, and their difference vector $x_i^{arg1} - x_i^{arg2}$ and feed them through a projection layer: $y_i = ReLU(W_y[x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] + b_y)$.

3.2. Commonsense Knowledge Extraction for Argumentative Relation Classification

Models for ARC will often require knowledge that is not overtly stated in the AUs or their context [28]. We aim to solve this issue by leveraging commonsense and lexical knowledge from resources such as ConceptNet and WordNet.

We begin by extracting connections between concepts mentioned in pairs of AUs from ConceptNet. For each pair we (i) collect all potentially relevant relations and concepts in a subgraph and (ii) select the top-ranked paths using local graph measures. Figure 2b, gives an overview of the extraction method.

Subgraph Construction For each pair $arg1$, $arg2$ we construct a subgraph $G' = (V', E')$ from ConceptNet $G = (V, E)$ by initializing V' with all concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$. To do so, we remove stop words, lemmatize tokens and perform n-gram matching of the remaining tokens to concepts in G. Similar to the subgraph construction in [25] and [26], we extend G' by including all concepts contained in the shortest paths between all concepts $c_i \in V'$ as well as all neighbouring nodes of concepts c_{arg} from $arg1$ and $arg2$. The final subgraph G' collects all edges E' from E that have both endpoints in V'.

Ranking and Selecting Paths We apply a two-step method: (i) **Collect top-n concepts**: Although most concepts in the AUs may be useful, considering all of them may introduce noise. For example, in Figure 1, the concept *possible* in arg_0 is not especially relevant in the given context. Therefore, we filter and collect the top-n concepts from each AU arg_i by ranking all the concepts $c_{arg_i} \in arg_i$ using personalized page rank [29] given the subgraph G' and all concepts $c_{arg_j} \in arg_j$ $(i \neq j)$, i.e., the concepts mentioned in the other argumentative unit. (ii) **Select top-k paths**: We then collect all shortest paths between the remaining concepts (of length $\leq$ 4 hops). We rank each node in the path with *closeness centrality* [30] scores. We select the top-k paths that connect any pair of filtered concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$, which we denote as **Selected Knowledge Paths (SKP)**.

3.3. Knowledge Graph Completion (KGC)

Knowledge graphs are incomplete, so we expect them to be more effective after a knowledge base completion step. For ConceptNet, this task has been addressed using link prediction [31,32]. [33] apply a pre-trained transformer model that learns to generate concepts as *phrase objects*, given an existing seed phrase (subject) and a ConceptNet relation label. On ConceptNet, they generate phrase objects with up to 91.7% precision. Human evaluations shows that the produced knowledge is novel and of high quality.

In contrast, [34] propose an open-world multi-label relation classification system[2] to predict ConceptNet *relation types* for given pairs of concepts. This system addresses specific properties of ConceptNet, such as the complexity of argument types and relation ambiguity. It encodes pairs of arguments (here, concepts) using word embedding inputs and an RNN component. The model constructs a joint representation that is projected to an output layer to predict one or several of 14 ConceptNet relation types (or none).

We adjust this classifier by: (1) refining the relation space and (2) pre- and postfiltering of concepts. Analyses in [34] show that the relation types *HasPrerequisite*, *HasSubevent* and *HasFirstSubevent* often co-occur, which indicates their ambiguity. To enhance the separation of these classes, we restructure the relation inventory. We retrain the classifier on the adapted dataset and perform pre- and postfiltering of concepts to reduce uninformative instances. Filtering steps include (i) TF-IDF filtering of concepts; (ii) excluding concepts covering more than two words; and (iii) type-based PoS sequence filtering on argument phrases. This enhances the performance by +9 pp. to 77 F1 score.

We apply the adapted KGC system to our data to predict and label *direct* links between any concepts detected in the AU pairs. We denote the predicted **extended knowledge relations EK** (for enhanced knowledge) and add it to the system (ARK+EK). In addition, we replace any *RelatedTo* triple in the Selected Knowledge Paths (SKP) with the predicted ConceptNet relations from EK and denote the result as **SKP***. E.g., an original triple is *umpire RelatedTo call* while the predicted triple is *umpire HasA call*. We update SKP to SKP*and combine it with additional predicted relations: ARK+SKP*+EK.

Lexical Knowledge WordNet[3] [35] is a widely used lexical resource. It defines the meaning of words and their relations for English. We employ WordNet's lexical knowledge by mapping each lemmatized token from the AUs to the WordNet graph, selecting the most frequent sense. We extract its SYNONYMS and sense definition. We denote WordNet knowledge as WN and knowledge acquired from WN as **Lexical Knowledge LK**.

3.4. Injecting Knowledge for ARC

We leverage commonsense knowledge for the ARC task from three sources: structured knowledge from ConceptNet via *Selected Knowledge Paths (SKP)* and *Enriched Knowledge (EK)*, and unstructured *Lexical Knowledge (LK)* from WordNet. SKP, EK and LK (SYNONYMS & Definitions) can all be represented as sets of (multi- or single-hop) paths $p_{1:l}$, i.e., sequences (of length l) of nodes (concepts) and edges (relation types). For LK, each path $p_{1:l}$ consists of the sequence of words from the sense definition of word w.[4]

Encoding Layer We use a single-layer BiLSTM to obtain encodings ($h^{k,i}$) for each knowledge path (h^k the encoded knowledge path, i the path index).

[2] https://gitlab.cl.uni-heidelberg.de/mbecker/corec—commonsense-relation-classifier

[3] https://wordnet.princeton.edu/

[4] We use the most frequent sense of w, as defined in WordNet. We embed each path $p^{k,i}_{1:l}$ with pretrained GloVe [36] embeddings ($k \in \{\text{SKP, EK, LK}\}$).

Attention Cell We define a cell that allows the model to attentively encode the knowledge paths (see Figure 2a). We use an attention layer, where each encoded knowledge path interacts with the argument representations x^{arg} (4) (to receive attention weights ($\hat{h}^{k,i}$) from (5). In (5) we use sigmoid to calculate attention weights,

$$x_i^{arg} = [x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] \qquad \widetilde{h}^{k,i} = \sigma(x_i^{arg} h^{k,i}), \quad \hat{h}^{k,i} = \frac{\widetilde{h}^{k,i}}{\sum_{j=1}^{N} \widetilde{h}^{k,j}} \tag{4, 5}$$

To obtain the argument-aware commonsense knowledge representation x_i^k, we pass the output of the attention layer through a feedforward layer. W_k, b_k are trainable parameters.

$$x_i^k = ReLU(W_k(\sum_{j=1}^{N} \hat{h}^{k,j} h^{k,i}) + b_k) \qquad o_i = sigmoid(W_z[x_i^{arg}; x_i^k] + b_z) \tag{6, 7}$$

To distill the selected and weighted knowledge into the model, we concatenate the argument x_i^{arg} and the knowledge x_i^k representation and process it by a dense layer (Eq. 8), with $\odot$ element-wise multiplication, $b_{\widetilde{y}_z}$ and $W_{\widetilde{y}_z}$ trainable parameters, y_i from *Output Layer*. Then, a sigmoid gate helps the model select when to incorporate knowledge x_i^k (Eq. 8).

$$z_i = softmax(W_{\widetilde{y}_z}(o_i \odot y_i + (1 - o_i) \odot x_i^k) + b_{\widetilde{y}_z}) \tag{8}$$

We finally pass the representation to a softmax classifier to form a probability distribution over the two classes *attack* and *support*.

4. Experiments

4.1. Data There is are only few datasets for the ARC task. We use these two datasets:[5]
Student Essays. This well-established dataset comprises argumentative essays in English written by students. We use the extended v.02 with 402 essays [4]. An issue with this data is that many of the relations can be easily identified by observing shallow discourse clues (*however*, *moreover*). Therefore, we we use the more difficult *content-based* setup [12], where the relations between argumentative units have to be determined without looking at the textual discourse context of unit clauses.
Debatepedia The Debatepedia website[6] collects user-generated debates that each contain several arguments in favor of or opposed to the debate's topic. Topics are usually formulated as polar questions. [1] created a small dataset from Debatepedia consisting of 200 pairs of topics (questions) and associated pro vs. con arguments, as well as further dependent pairs of pro and con arguments among each topic. But the pairing of coherent pro and con arguments is difficult to establish automatically. We thus restrict ourselves to pairs of directly connected questions and pro/con arguments. To construct high-quality data, we manually reformulate the questions to statements. If an argument is in favor of the debated topic, the claim *supports* the topic. Else it *attacks* it.

[5] Below we summarize the data statistics:

Student Essay	train:	2803 / 273 (support / attack)	dev:	1017 / 132 (support / attack)
Debatepedia	train:	3240 / 3251 (support / attack)	dev:	1121 / 1042 (support / attack)

[6] http://www.debatepedia.org

4.2. Linear Classifier Baseline Among other text classification tasks, linear SVMs have been successfully applied to ARC [37,38,4,39]. Next to our neural system we thus implement an SVM model w/ and w/o knowledge enhancement. Below we describe text classification features used by our baseline SVM and explain ways of modeling and abstracting the knowledge paths to make them accessible for the SVM.
Text features. We feed the SVM a concatenation of the uni- and bigram (TF-IDF) representation of (i) source, (ii) target and (iii) the text overlap of source and target. We also concatenate averaged GloVe vectors to the bag-of-words feature representation; the vectors are separately averaged over (i), (ii) and (iii). We further concatenate to the vector the element-wise subtraction and multiplication of the averaged source from the averaged target GloVe vector, to model the argumentative relation as a directional vector.
Modeling paths as features. We investigate whether the extracted and selected knowledge paths (SKP) can improve the SVM classifier. But encoding paths is not straightforward for an SVM compared to encoding sequential paths with a recurrent NN. We thus apply the following steps: we represent every selected path as the mean vector of the token-wise GloVe vectors in a path. We then retrieve different path selections, e.g., the mean vector of all paths or the path-vector with the maximum and minimum norm. To determine the optimal selection jointly with the optimal SVM margin, we run a greedy hyper-parameter search on the development data. Details will be provided with the code.

4.3. Training Details **Objective** During training we minimize the cross-entropy loss between the predicted and the actual distribution. We use Adam optimizer [40] with an initial learning rate of 0.001, and batch size of 8/32 for Student Essays/Debatepedia. We use pretrained GloVe [36], ELMo [41] embeddings, a hidden size of 100 for all Dense Layers and L2 regularization with $\lambda = 0.01$. We use $k = 3$ for selecting top-ranked paths. For filtering the number of concepts with *personalized page rank* we use $n \leq 5$ concepts per AU. **Metrics** We report macro-averaged Precision (P), Recall (R), F1 scores.

5. Results

We examine 8 different systems: **random** baseline guesses labels according to the training data label distribution. **SVM** is a knowledge-agnostic linear classifier baseline. When we add selected knowledge paths via aggregation features, we denote this as **SVM+CN** (w/ knowledge from ConceptNet, including SKP* and EK) and as **SVM+CWN** (for the latter (+CN) extended with WordNet). **BiLSTM** is a neural knowledge-agnostic baseline and **Bi-ATT** denotes the BiLSTM with self- and co-attention (see Fig. 2a *w/o* Attention Cell and Sigmoid Gate). By further enriching Bi-ATT with knowledge paths through the Attention Cell, we obtain our main model: **ARK** (again in different varieties: **+CN**, etc.).

Table 1a reports our experiment results in averaged scores over five runs. Our models enhanced with knowledge (including SVM) perform significantly better ($p<0.05$) compared to their baselines, and similarly for ARK+CWN vs. KOB2019.

Knowledge helps The results show that adding selected knowledge to any of our baseline models improves their overall performance on both datasets and for both types of embeddings. Our full model **ARK** profits most from the added knowledge when compared to its knowledge-agnostic counterpart **Bi-ATT** (using ELMo: +4.27 pp. (percentage points) macro F1 in Student essays; +4.6 in Debatepedia; when using GloVe: +4.33

Table 1. (a) Classification results and (b) ablation study over K-path selection methods & K-graphs.

(a) Classification results. Bi-ATT = BiLSTM+Attention model, ARK = ARC model + Knowledge, where CN = ConceptNet (incl. SKP* + EK); WN = WordNet; CWN = ConceptNet (with SKP* + EK) + WordNet. Superscripts mark significant improvement (✓) or not (✗) of the result relative to the model the index names.

		Student essays			Debatepedia		
Model	**WE**	**P**	**R**	**F1**	**P**	**R**	**F1**
(1) random	-	49.68	49.66	49.65	50.04	50.03	50.01
(2) BiLSTM	G_{300d}	53.53	52.89	53.13	55.67	55.68	55.63
(3) KOB2019	G_{300d}	52.79	51.85	52.05 $^{(2)}$✗	58.06	57.75	57.04 $^{(2)}$✓
(4) KOB2019	ELMo	55.72	53.16	54.37 $^{(2)}$✓	59.16	59.17	59.11 $^{(2)}$✓
(5) SVM	G_{300d}	54.11	52.59	52.95	54.73	54.71	54.52
(6) SVM + CN	G_{300d}	54.11	54.23	54.17	56.12	56.00	55.58
(7) SVM + CWN	G_{300d}	55.80	56.38	**56.06** $^{(5, 3)}$✓	56.60	56.57	**56.37** $^{(5)}$✓$^{(3)}$✗
(8) Bi-ATT	G_{300d}	54.46	53.31	53.70	56.20	56.19	56.18
(9) ARK + WN	G_{300d}	57.68	55.71	56.44	57.49	57.48	57.48
(10) ARK + CN	G_{300d}	57.64	57.71	57.67	57.38	57.25	57.31
(11) ARK + CWN	G_{300d}	60.70	55.55	**58.03** $^{(8, 3)}$✓	58.78	58.43	**58.60** $^{(8, 3)}$✓
(12) Bi-ATT	ELMo	56.44	54.77	55.16	59.10	59.08	59.09
(13) ARK + WN	ELMo	57.13	56.26	56.69	63.00	62.70	62.85
(14) ARK + CN	ELMo	59.13	**58.68**	58.89 $^{(12)}$✓	63.64	63.45	63.50 $^{(12)}$✓
(15) ARK + CWN	ELMo	**63.43**	55.90	**59.43** $^{(12, 4)}$✓	63.72	63.65	**63.69** $^{(12, 4)}$✓

(b) Ablation study over KnowledgePath (KPATH) selection methods & Knowledge Graphs (K). Models: random: 3 randomly chosen paths (= no selection); LK: Lexical Knowledge; SKP = Selected Knowledge Paths; SKP* = SKP w/ (*RelatedTo* → EK) and all WE=ELMo.

		Student essays			Debatepedia		
KPath selection	**K**	**P**	**R**	**F1**	**P**	**R**	**F1**
random KPaths	CN	56.73	57.80	57.16	60.50	60.16	60.33
SKP	CN	58.22	58.64	58.25	63.38	63.04	63.12
EK	CN	**59.58**	54.95	56.11	**63.90**	62.79	63.34
SKP* + EK	CN	59.13	58.68	**58.89**	63.64	**63.45**	**63.50**
LK	WN	57.13	56.26	56.69	63.00	62.70	62.85
SKP*+EK+LK	CWN	**63.43**	55.90	**59.43**	63.72	63.65	**63.69**

pp. in Student essays; +2.42 in Debatepedia). This finding not only applies to the global F1 metric, but also to macro Precision and Recall: we obtain considerable gains in Recall on Student essays of over 4 pp., i.e., a relative increase of more than 8%. Deeper analysis in §6 will show that knowledge helps especially for classifying rare *attack*-examples. We compare our knowledge representation and extraction method with the method in [14]. We empirically show that across two datasets and different embeddings we gain +4 F1 (on average) improvement. Knowledge also helps the linear SVM baseline (**SVM** vs. **SVM+CN/+CWN**). For both datasets we see gains. Adding only knowledge from ConceptNet improves over SVM by +1.22 pp. macro F1 in Student essays; +1.06 in Debatepedia. With access to the full knowledge we observe a more notable gain: +3.11 pp. macro F1 in Student essays; +1.85 pp. in Debatepedia (**SVM+CWN**). The fact that a linear classifier profits less from added knowledge compared to the neural system (**Bi-ATT** vs. **ARK**) is expected: the knowledge paths are sequential and thus easier to model with recurrent computations of the neural model. When computing path aggregates to make knowledge paths accessible for the SVM, we lose important structural information.

Ablation Study To gain better insight into the effects of different kinds of knowledge and selection methods, we run an ablation study over variants of ARK, where the number of paths is constant: In Table 1b row 1 we **randomly select** from the set of shortest knowledge paths between concepts appearing in AUs; row 2 uses knowledge selected using graph measures (**SKP**); row 3 shows model performance when using extended knowledge predicted on the fly with knowledge completion (**EK**); row 4 uses **both SKP*** **and EK**. Table 1b shows that using selected knowledge paths (SKP) improves F1 macro scores over models that use randomly selected knowledge paths. The effect is smaller for Student essays (+1.09 pp. F1), but considerable for Debatepedia (+2.79 pp. F1). Models that include automatically predicted knowledge for specific items (EK+SKP*) yield a further improvement of 0.64 and 0.38 pp. F1 macro scores for Student Essays and Debatepedia. This demonstrates that both knowledge selection and the instance-specific enrichment of the knowledge graph is important, and that EK complements SKP*.

Table 2. Example of knowledge paths used for prediction of argumentative relations.

Source	Relation	Predicted	Argumentative Unit 1	Argumentative Unit 2	Knowledge Paths
Essay	attack	attack	online classes have many advantages	traditional learning still has many benefits to the students	*benefit* RELATEDTO *advantage*; *online* ANTONYM *brick_and_mortar* SYNONYM *traditional*
Debate	support	support	Trans fats can be replaced w/o changing taste/price.	Trans fats should be banned.	*ban* ISA *action* RELATEDTO *change* RELATEDTO *replace*
Debate	support	attack	Instant replay call reviews should be implemented in baseball.	Instant replay makes game more about players, less about umps	*umps* FORM OF *ump* SYNONYM *umpire* RELATED TO *call*, *player* RELATEDTO *game* RELATEDTO *baseball* HASCONTEXT *ump* FORMOF *umps*

6. Analysis and Discussion

Minority Class Classification Data examples labeled with *attack* are scarce. This situation is extreme in Student Essays, where less than 10% of the data carry the *attack*-label. Therefore, systems usually struggle with this class (cf. [4]) and compensate for bad classification of *attack* examples with very good classification of *support* examples. Nevertheless, the minority class (*attack*) is at least as important as the *support* class. Thus, it is notable that our knowledge-enhanced systems **ARK+CN/+CWN** obtain a +53.8%/+93.3% relative increase in detecting the *attack* class (compared to the **Bi-ATT** baseline). WordNet, when used as sole source of knowledge (**ARK+WN**), leads to a lower but still remarkable improvement of +15.4%. A similar outcome is observed for the *linear model*: SVM+CN obtains a +66% increase for the *attack* class, while experiencing a -3.56% loss for *support*. To summarize, our selected paths greatly improve the results with respect to successfully predicting examples of the minority class (*attack*).

Knowledge Path Examples for Improved ARC To shed some light on how knowledge helps our ARC system, we analyze cases where the knowledge-enhanced neural model (**ARK**) corrects a mis-classification of the knowledge-agnostic model (**Bi-ATT**) with high probability. Some cases are displayed in Table 2. In the first case, a system lacking deeper knowledge can easily be fooled: both argument units contain phrases which are highly similar and carry positive sentiment (*advantages*; *benefits*) – yet, they are in an *attack* relation. A knowledgeable system, by contrast, would understand that 'online classes' and 'traditional learning' are opposites of each other. This valuable information is reflected in the retrieved two-hop path (right column): *online* ANTONYM brick-and-mortar SYNONYM *traditional*. To get from the *online*-concept to the *traditional*-concept, we have to traverse an ANTONYM-edge. This may signal to the system that despite the semantically highly similar content, the units are in fact attacking each other. In the second example, the system needs to understand that the word 'replace' in unit 1 has an implicit relation with 'banned' in unit 2 – again, this is captured by the selected path.

'Gay rights' or 'environment' – where does knowledge help? While our results indicate that knowledge is important for ARC, we found that the system needs more topic-specific common-sense knowledge. In the 3rd example (Table 2), although we extract and identify the relation between *players* and *umps* given the context, the missing knowledge is that in the sports domain, for replays *players* are more important than *umpires* – knowledge which we neither find in domain-specific KBs nor in commonsense KBs. We investigate the impact of knowledge infusion for different debate topics by clustering all topics in the dev set into 18 major areas (details will be released). '*Trans fats should be banned.*', e.g., appears in FOOD & NUTRITION; GAY RIGHTS includes debates such as *'Gay marriage should be legalized.'*. Figure 3 shows the comparative model perfor-

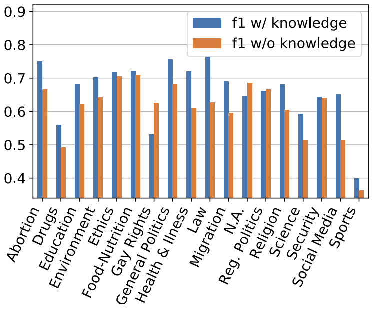


Figure 3. Macro F1 results of Bi-ATT vs. ARK+CWN model across 18 debate topic clusters (on DevSet).

mance over these topics. In 15 out of 18 topics injection of knowledge helps, especially in HEALTH, SOCIAL MEDIA and LAW, with great gains in macro F1 of more than 10 pp. By contrast, adding knowledge incurs a loss in GAY RIGHTS.

7. Conclusions

Determining relations between arguments requires knowledge beyond the text. In this work, we investigate ways of improving linear *and* neural systems by feeding knowledge paths that link concepts from two argumentative units. We extract the paths from background knowledge graphs and filter them with graph algorithms. Our experiments show that our method for incorporating commonsense knowledge is efficient for improving overall ARC results across two datasets. We show that extending the knowledge *on the fly* improves model performance – which further emphasizes the impact of knowledge for the task. An in-depth analysis shows that knowledge improves the performance across many topics, with very few exceptions. Finally, we provide an enhanced dataset for *support/attack* classification based on Debatepedia, which we will publicize.

References

[1] Cabrio E, Villata S. Natural language arguments: A combined approach. In: ECAI; 2012. p. 205–210.

[2] Stab C, Gurevych I. Annotating Argument Components and Relations in Persuasive Essays. In: COLING; 2014. p. 1501–1510.

[3] Stab C, Gurevych I. Identifying Argumentative Discourse Structures in Persuasive Essays. In: EMNLP; 2014. p. 46–56.

[4] Stab C, Gurevych I. Parsing Argumentation Structures in Persuasive Essays. Computational Linguistics. 2017;43(3):619–659.

[5] Reisert P, Inoue N, Okazaki N, Inui K. Deep Argumentative Structure Analysis as an Explanation to Argumentative Relations. In: ACL; 2017. p. 38–41.

[6] Speer R, Havasi C. Representing General Relational Knowledge in ConceptNet 5. In: LREC; 2012. p. 3679–3686.

[7] Walton D. Goal-based Reasoning for Argumentation. Cambridge University Press; 2015.

[8] Palau RM, Moens MF. Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In: ICAIL; 2009. p. 98–107.

[9] Nguyen H, Litman D. Context-aware Argumentative Relation Mining. In: ACL; 2016. p. 1127–1137.

[10] Peldszus A, Stede M. Joint Prediction in MST-style Discourse Parsing for Argumentation Mining. In: EMNLP; 2015. p. 938–948.

[11] Persing I, Ng V. End-to-End Argumentation Mining in Student Essays. In: NAACL; 2016. p. 1384–1394.

[12] Opitz J, Frank A. Dissecting Content and Context in Argumentative Relation Analysis. In: Workshop on Argument Mining; 2019. p. 25–34.

[13] Cocarascu O, Toni F. Identifying Attack and Support Argumentative Relations Using Deep Learning. In: EMNLP; 2017. p. 1374–1379.

[14] Kobbe J, Opitz J, Becker M, Hulpus I, Stuckenschmidt H, Frank A. Exploiting Background Knowledge for Argumentative Relation Classification. In: LDK; 2019. p. 1–8.

[15] Opitz J. Argumentative Relation Classification as Plausibility Ranking. In: KONVENS; 2019. p. 193–202.

[16] Moens MF. Argumentation Mining: How Can a Machine Acquire Common Sense and World Knowledge? Argument & Computation. 2018 01;9:1–14.

[17] Lawrence J, Reed C. Argument Mining: A Survey. Computational Linguistics. 2019:1–55.

[18] Potash P, Bhattacharya R, Rumshisky A. Length, Interchangeability, and External Knowledge: Observations from Predicting Argument Convincingness. In: IJCNLP; 2017. p. 342–351.

[19] Choi H, Lee H. GIST at SemEval-2018 Task 12: A Network Transferring Inference Knowledge to Argument Reasoning Comprehension Task. In: SemEval; 2018. p. 773–777.

[20] Chen Z, Song W, Liu L. TRANSRW at SemEval-2018 Task 12: Transforming Semantic Representations for Argument Reasoning Comprehension. In: SemEval; 2018. p. 1142–1145.

[21] Saint-Dizier P. Knowledge-driven argument mining based on the qualia structure. Argument & Computation. 2017;8:193–210.

[22] Pustejovsky J. The generative lexicon. Computational linguistics. 1991;17(4):409–441.

[23] Botschen T, Sorokin D, Gurevych I. Frame- and Entity-Based Knowledge for Common-Sense Argumentative Reasoning. In: Workshop on Argument Mining; 2018. p. 90–96.

[24] Habernal I, Wachsmuth H, Gurevych I, Stein B. SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. In: SemEval; 2018. p. 763–772.

[25] Bauer L, Wang Y, Bansal M. Commonsense for Generative Multi-Hop Question Answering Tasks. In: EMNLP; 2018. p. 4220–4230.

[26] Paul D, Frank A. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. In: NAACL; 2019. p. 3671–3681.

[27] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical Attention Networks for Document Classification. In: NAACL; 2016. p. 1480–1489.

[28] Rajendran P, Bollegala D, Parsons S. Contextual Stance Classification of Opinions: A Step towards Enthymeme Reconstruction in Online Reviews. In: Workshop on Argument Mining; 2016. p. 31–39.

[29] Haveliwala TH. Topic-Sensitive PageRank. In: WWW; 2002. p. 517–526.

[30] Bavelas A. Communication Patterns in Task-Oriented Groups. Journal of the Acoustical Society of America. 1950;22(6):725–730.

[31] Li X, Taheri A, Tu L, Gimpel K. Commonsense Knowledge Base Completion. In: ACL; 2016. p. 1445–1455.

[32] Saito I, Nishida K, Asano H, Tomita J. Commonsense Knowledge Base Completion and Generation. In: CoNLL; 2018. p. 141–150.

[33] Bosselut A, Rashkin H, Sap M, Malaviya C, Asli C, Yejin C. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In: ACL; 2019. p. 4762–4779.

[34] Becker M, Staniek M, Nastase V, Frank A. Assessing the Difficulty of Classifying ConceptNet Relations in a Multi-Label Classification Setting. In: RELATIONS - IWCS Workshop; 2019. p. 1–14.

[35] Miller GA. WordNet: A Lexical Database for English. Communications of the ACM. 1995;38(11):39.

[36] Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation. In: EMNLP; 2014. p. 1532–1543.

[37] Pradhan S, Hacioglu K, Krugler V, Ward W, Martin JH, Jurafsky D. Support Vector Learning for Semantic Argument Classification. Machine Learning. 2005;60(1–3):11–39.

[38] Kim Y. Convolutional Neural Networks for Sentence Classification. In: EMNLP; 2014. p. 1746–1751.

[39] Aker A, Sliwa A, Ma Y, Lui R, Borad N, Ziyaei S, et al. What Works and What does not: Classifier and Feature Analysis for Argument Mining. In: Workshop on Argument Mining; 2017. p. 91–96.

[40] Kingma DP, Ba JL. Adam: A Method for Stochastic Optimization. In: ICLR; 2014. p. 1–15.

[41] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: NAACL; 2018. p. 2227–2237.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200516

Increasing the Naturalness of an Argumentative Dialogue System Through Argument Chains

Niklas RACH [a] Wolfgang MINKER [a] and Stefan ULTES [b]
[a] *Institute of Communications Engineering, Ulm University, Germany*
[b] *Mercedes-Benz Research & Development, Sindelfingen, Germany*

Abstract. This work introduces chained arguments into a dialogue game for argumentation to allow a more natural and intuitive interaction with a respective system. Thus, the turn taking rules of the game are improved while still preserving the general consistency that is ensured by the framework. The improved system is used to generate artificial dialogues between two virtual agents which are assessed in a user study. The results show a significant improvement in the perceived naturalness without violating the logical consistency.

Keywords. Dialogue Games for Argumentation, Argumentative Dialogue, Argumentation Strategies

1. Introduction

Empowering virtual agents with the ability to exchange arguments and to engage in argumentative dialogue is a desirable, yet challenging task. Due to the complexity of suchlike conversations, respective systems often utilize formal modelling of the dialogue in order to achieve consistency and reasonableness in the interaction [1, 2, 3, 4].

However, despite the logical advantage of dialogue models, the resulting interactions are restricted by the formalism and can be perceived as significantly less natural than human discussion [5]. The task at hand is thus to find the balance between reasonable restrictions and a freedom of choice that allows a natural and intuitive interaction. The difficulty lies in the implications that come with this freedom, as the possibility of a more natural response may also include the possibility of responses that are neither natural nor consistent and violate the basic principles of the desired interaction. Especially modifications to an established formalism have thus to preserve the general properties of the model and extend the respective regulations rather than simply drop them.

Within this work we address this task in view of the dialogue game for argumentation introduced in [6, 7]. This choice is due to the fact that the model allows for multiple as well as postponed responses to an utterance which means that it provides a certain freedom of choices for the players by design. We built upon this freedom and increase it by modifying the underlying game protocol to allow for a chaining of multiple arguments. This is done in a way that is in line with the remaining regulations and does thus not violate the logical consistency of the resulting dialogues. Our approach is tested by generating artificial dialogues between two virtual agents that are rated in a user study with respect to their logical consistency as well as their naturalness. The setup is similar to [5], where the unmodified version of the dialogue game was applied. We show that by adapting the protocol, the perceived consistency remains whereas the naturalness is significantly increased.

The remainder of this paper is as follows: Section 2 covers the related work on argumentative dialogue systems whereas Section 3 discusses the theoretical background of dialogue games for argumentation with an emphasis on the original framework. Section 4 addresses the applied modifications and their implications whereas Section 5 includes details on the evaluation setup, the user study and the corresponding results. The work is closed in Section 6 with a conclusion and an outlook on future work.

2. Related Work

Multiple approaches to human-machine argumentation have been discussed that utilize different models to structure the interaction. In the recently introduced IBM Debater[1], the boundaries of the interaction are given by the debating rules, meaning that speaking time and turn taking are fixed by the overall setup. As a consequence, the main task of the system consists of the automatic analysis of opponent utterances with fixed length and the generation of a suitable response.

One approach to address formal issues like turn taking is to limit the system response to one argument per turn [8, 9]. Consequently, additional options like questioning the validity of an argument or chaining multiple arguments are not considered. In addition, a generative approach to argumentative chat bots was discussed in [10]. Although in this case no rules in view of the interaction are imposed, the system capacities are limited to strategies that can be derived from the training data.

Similar to our setup, dialogue games for argumentation were previously considered as an approach to model argumentative dialogue. Overviews over existing dialogue games for argumentation were presented in [11, 12] and a framework to facilitate their implementation and the development of respective applications was introduced in [13]. Even though several systems utilizing these or similar frameworks in different domains were introduced [1, 2, 3, 14, 15], the main focus of the underlying formal models is usually to preserve logical coherence. Therefore, these models enforce restrictions that can lead to interactions that are not perceived as natural when compared to human discussions [5]. In order to address this issue, the herein introduced extension focuses on this explicit property in order to increase the freedom of choices within the framework and enable a more intuitive and natural argumentation.

3. Dialogue Games for Argumentation

Within this section we recall the theoretical background on dialogue games for argumentation, following the formal description introduced in [7]. Dialogue games in general are a model of conversation, meaning that they extend the formal approach of speech acts to their effect on the listener [16]. A dialogue game for argumentation can be described as

[1] https://www.research.ibm.com/artificial-intelligence/project-debater/

Table 1. Communication language L_c of the original framework [7]. Upper-case variables denote arguments out of *Args*, lower-case variables denote elements of L_t.

Speech Act	Attacks	Surrenders
claim(a)	*why(a)*	*concede(a)*
why(a)	*argue(A)* ($conc(A) = a$)	*retract(a)*
concede(a)	-	-
retract(a)	-	-
argue(A)	*why(a)* ($a \in prem(A)$), *argue(B)* (B defeats A)	*concede(a)* ($a \in prem(A)$ or $a = conc(A)$)

tuple $(\mathscr{L}, D)$, with $\mathscr{L}$ a logic for defeasable argumentation [17] and D the so called dialogue system proper. $\mathscr{L}$ includes the set of arguments *Args* on which a (binary) defeat relation is defined. Arguments are AND-trees with nodes out of a logical language L_t. The AND-links are instantiations of inference rules out of a set R defined over L_t. The set of leaves of an argument A is called its premises (*prem(A)*) and the root is called conclusion (*conc(A)*). We call an argument B an extending argument of A if $conc(B) \in prem(A)$.

The dialogue system proper D structures the interaction and consists of a communication language L_c, a game protocol P and commitment rules C. A game is played in turns, whereas each turn includes at least one *move*. A temporally ordered sequence of moves is called a *dialogue*.

In the following, we focus on Prakken's framework for *relevant dialogues* defined in [7]. The corresponding communication language L_c is shown in Table 1, ordered by attacking and surrendering replies. Each move in the corresponding game includes one speech act out of L_c as well as a temporal identifier and replies to one specific earlier move. The game is played by two players P1 (proponent) and P2 (opponent) and is initiated by the proponent with either a *claim* or an *argue* move.

In order to ensure consistency in the responses, the protocol P determines the legality of moves in each dialogue based on a relevance criterion. In order to determine this relevance, a binary status is assigned to each played move, defining it as either *in* or *out*. A move is *out* if the dialogue includes an attack on it that is *in*. Otherwise the move is *in*. If an attack on a move m_i would change the status of the initial move, m_i is a relevant target. The player to move can only address relevant targets in his or her turn. The turn of each player ends once he or she manages to switch the status of the initial move in his or her favour. If a player has no legal move left and thus cannot switch the status of the

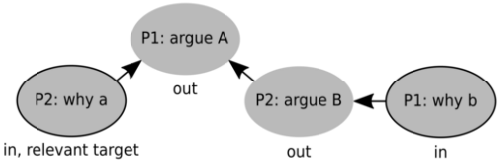


Figure 1. Illustration of the relevance criterion. Both *why* moves are not attacked and therefore *in* (indicated by black margins of the circles). Consequently, their targets are *out*. Only the *why(a)* move is a relevant target since an attack on it would change the status of the opening move *argue(A)*. Consequently, it is the turn of P1.

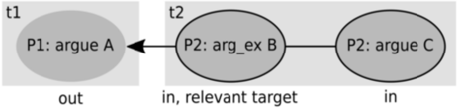


Figure 2. Illustration of a chain consisting of one *argue_extend* and one *argue* move. Grey boxes indicate the corresponding turns (t1 and t2) in the game.

initial move, he or she loses. An abstract example dialogue between two players P1 and P2 is shown in Figure 1 in order to illustrate the turn taking and the relevance criterion.

4. Extension to Chained Arguments

The main restriction of the above discussed formalism in view of naturalness lies in the inability of introducing more than one argument per turn. More precisely, a player is only allowed to extend an argument, if the corresponding move was challenged by a *why* move. This section introduces an extension which allows players to chain multiple arguments in a single turn without violating the logical consistency of the dialogue. In order to do so, we introduce an additional speech act and modifications to the protocol.

In the original formalism, a player has to move until he or she switched the status of the initial move in his or her favour, meaning that he or she plays an unspecified number of surrendering moves, followed by a single attack. After the status of the initial move is switched, the turn ends immediately. We modify this rule by allowing both players to *extend* their attack under the condition, that the attack includes an argument. An extended attack generally allows the player to introduce additional arguments to undermine his or her current move even before it was challenged. This extension does thus not reply to an actual attack but to an anticipated one. Formally, the extension is represented by an additional speech act (*argue_extend(A)*) which has the same properties as the *argue(A)* act in view of allowed attacking and surrendering replies but does not end the turn of the corresponding player. An *argue_extend* move can only be played if an extending argument is available. We call a series of *argue*(*_extend)* moves an (argument) *chain*.

In the following, we discuss implications and changes in the game that arise from this modification. When introduced, each move in a chain is *in*. The first move in a chain is also a relevant target since an attack on it changes the status of the initial move. The remaining moves on the other hand are not relevant targets. Moreover, challenging the relevant target in a chain only switches the status of the initial move if this challenge is not anticipated in the chain. Otherwise, the responding move in the chain becomes a relevant target and the current player is obliged to play another move. An example of a chain consisting of two moves, including status and relevance is shown in Figure 2. Attacking replies to a chain can have multiple forms as illustrated in Figure 3:

- A chain can be attacked by a series of anticipated *why* moves, followed by a *why* move that is not anticipated (Response 1).
- A chain can be attacked by a combination of anticipated *why* and *argue(_extend)* moves (Response 2).
- A chain can be attacked by an attacking reply to its first move that is not anticipated in the chain (Response 3).

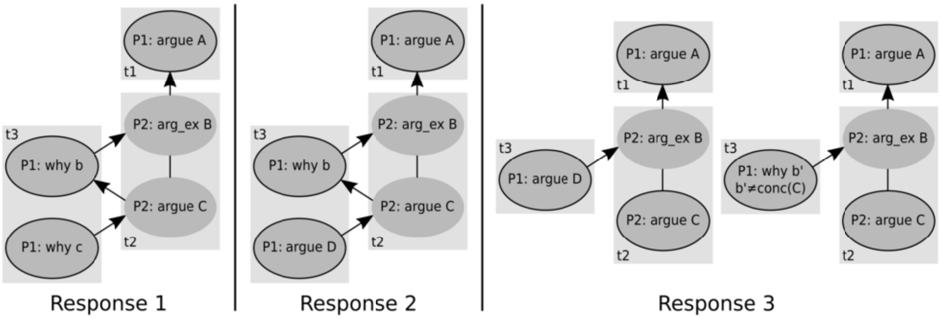


Figure 3. Illustration of the three possible response types to a chain. The arrow from the *argue C* move indicates the implicit response of the argument chain to the anticipated attack.

Generally, responses to a chain may also include a (new) chain, thus giving the players far more freedom in their choices. Nevertheless, since the legality of moves is still determined by the same principles as in the original framework, the resulting dialogues have the same formal consistency.

5. Evaluation

In order to evaluate the discussed extensions, we generate artificial dialogues between two virtual agents Alice and Bob and assess them in a user study. The evaluation setup was chosen in order to compare the results directly with the ratings for the original framework in an earlier work [5]. In order to ensure a fair comparison, the setup is as similar as possible to the original one. Thus, we employ the same multi-agent setup, including the same arguments, the same dialogue manager model and a similar natural language generation (NLG) as well as the same questionnaire for the survey.

5.1. Multi-Agent Setup

The set of arguments is derived from 72 argument components on the topic "Marriage is an outdated institution" annotated on a debate from idebate.com[2] following the argument annotation scheme introduced in [18]. The annotation scheme includes three kinds of argument components (Major Claim - MC, Claim - C, and Premise - P) and two directed relations (support and attack) between them. Each component apart from the Major Claim targets exactly one other component with a relation. Consequently, the resulting structure can be represented as a tree from which we derive the arguments of the form $A = a, so\ b$ (a supports b) and $A' = a', so\ \neg b'$ (a' attacks b').

During the interaction, the agents select their next move from the list of available options provided by the dialogue game. In order to ensure a competitive but reasonable strategy, each agent first prefers attacking moves over surrendering moves, then *argue(_extend)* moves over *why* moves and finally immediate to postponed responses. If there are multiple options with the same preference, the selection between them is random. Moreover, both agents extend their line of argumentation as long as possible. As in

[2] https://idebate.org/debatabase (last accessed 06 May 2020)

Table 2. Excerpt of an artificial discussion on the topic *Marriage is an outdated institution* including speech acts and NLG output. Italic text indicates the annotated sentences of the argument components.

Speech Acts	Utterance
argue_ex(C_1, *so* ¬*MC*)	It seems to me that *marriage is an important institution to religious people.*
argue(P_1, *so* C_1)	I would like to go into that a little further. You see, *there are still such huge numbers of people who practice religions to which marriage is integral.*
why(C_1)	Unfortunately I didn't find that entirely convincing. Would you mind elaborating a little further?
argue(P_2, *so* ¬P_1)	In particular, there's one aspect of your argumentation that I have some doubts about. You said that *there are still such huge numbers of people who practice religions to which marriage is integral.* It seems to me that *religion as a whole is becoming less important and, with it, marriage is becoming less important.*

the original work, we allow an *argue*(a, *so* ¬*MC*) attacking reply to *claim(MC)* to cover all available arguments. This attack can also be extended in the modified framework.

The NLG is done turn wise, meaning that all moves of a turn are merged into one utterance. As in the reference work, the natural language representation of arguments is gained from the annotated sentences of the argument components. Postponed *argue(_extend)* replies include the premise and the conclusion. In the case of a direct reply, the conclusion is left implicit. For the remaining moves, a list of templates is used from which the system selects randomly. Again, the natural language representation of the argument component in the move is left implicit for direct replies and explicitly included in the case of a postponed reply. In addition, we generate a new list of connecting and opening phrases in order to concatenate multiple moves into a single utterance. This part of the NLG is an extension to the original version and may influence the user perception of the resulting dialogues. However, since this extension is only possible due to the extended framework, the advanced NLG template is a direct result of the formal extensions. An example of two utterances[3] and the speech acts of the corresponding turns is shown in Table 2.

5.2. User Study

To compare our approach with the original framework, we generated ten virtual discussions between the agents Alice (proponent) and Bob (opponent) with the new framework and evaluated them in a user survey with the same study setup as in the referenced work. In the original case, 20 dialogues were required in order to cover a majority of the available arguments, which was mainly due to the extensive use of isolated *why* moves. As those are merged into a single utterance within the modified framework, ten dialogues were sufficient to present a similar amount of arguments.

The questionnaire consists of ten questions related to the strategy, the line of argumentation and the naturalness of the dialogue. Each question was rated on a five point scale (1 completely disagree, 5 completely agree) by 61 participants from the UK with an age between 18 and 99. The survey was realized by clickworker[4] and each participant

[3] Material reproduced from www.iedebate.org with the permission of the International Debating Education Association. Copyright ©2005 International Debate Education Association. All Rights Reserved

[4] https://marketplace.clickworker.com (last accessed 06 May 2020)

Table 3. Results for the original framework (Original) and extended one (Modified). Bold lines indicate a significant difference.

Question	Original	Modified	p
The arguments presented by Bob are logically consistent responses to the utterances they refer to.	4.0	4.0	0.36
The arguments presented by Alice are logically consistent responses to the utterances they refer to.	3.5	3.0	0.81
Bob's line of argumentation is not logically consistent.	2.0	2.0	0.85
Alice's line of argumentation is not logically consistent.	2.0	2.0	0.74
It was difficult to follow the line of argumentation throughout the debate.	**3.0**	**2.0**	**0.02**
The whole debate is natural and intuitive.	**2.0**	**4.0**	**0.02**

was assigned a single randomly selected discussion in order to avoid a bias. The wording of questions that are relevant for the herein discussed topic together with the corresponding median (original and modified framework) as well as the p value achieved with a Mann-Whitney-U test are shown in Table 3. We see that the four questions related to the logical consistency of the argumentation show no significant difference to the original results, whereas the p value for both questions related to the naturalness is below the threshold of 0.05. For the sake of completeness, we report that no significant difference was found for the questions omitted in Table 3. We conclude that the herein discussed modification significantly improves the perceived naturalness of the resulting dialogues without lowering the consistency.

6. Conclusion

This work discussed the extension of an existing dialogue game for argumentation in order to enable a more natural interaction with a respective system. Our approach allows for chained arguments from both sides while preserving the regulations that ensure consistency. We evaluated the new framework by generating artificial discussions between two virtual agents that were rated in a user study. In a direct comparison to the results achieved in the reference work with the original framework, we see a significant improvement in ratings related to the naturalness and intuitiveness of the dialogue. On the other hand the perceived consistency of the dialogue remains the same.

Future work will focus mainly on exploring the generated freedom by means of optimization techniques like reinforcement learning [19]. We will also investigate if and how the game protocol can be additionally modified, in order to increase the freedom of choices for respective agents further. Finally, we want to explore the new framework in the interaction between a dialogue system and real users.

Acknowledgements: This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ”How to Win Arguments - Empowering Virtual Agents to Improve their Persuasiveness”, Grant Number 376696351, as part of the Priority Program ”Robust Argumentation Machines (RATIO)” (SPP-1999).

References

[1] Yuan T, Moore D, Grierson A. A human-computer dialogue system for educational debate: A computational dialectics approach. International Journal of Artificial Intelligence in Education. 2008;18(1):3–26.

[2] Yuan T, Moore D, Reed C, Ravenscroft A, Maudet N. Informal logic dialogue games in human–computer dialogue. The Knowledge Engineering Review. 2011;26(2):159–174.

[3] Rach N, Weber K, Pragst L, André E, Minker W, Ultes S. EVA: A Multimodal Argumentative Dialogue System. In: Proceedings of the 2018 on International Conference on Multimodal Interaction. ACM; 2018. p. 551–552.

[4] Bench-Capon TJ. Specification and implementation of Toulmin dialogue game. In: Proceedings of JURIX. vol. 98; 1998. p. 5–20.

[5] Rach N, Langhammer S, Minker W, Ultes S. Utilizing argument mining techniques for argumentative dialogue systems. In: 9th International Workshop on Spoken Dialogue System Technology. Springer; 2019. p. 131–142.

[6] Prakken H. On dialogue systems with speech acts, arguments, and counterarguments. In: JELIA. Springer; 2000. p. 224–238.

[7] Prakken H. Coherence and flexibility in dialogue games for argumentation. Journal of logic and computation. 2005;15(6):1009–1040.

[8] Rosenfeld A, Kraus S. Strategical Argumentative Agent for Human Persuasion. In: ECAI; 2016. p. 320–328.

[9] Rakshit G, Bowden KK, Reed L, Misra A, Walker M. Debbie, the debate bot of the future. In: Advanced Social Interaction with Agents. Springer; 2019. p. 45–52.

[10] Le DT, Nguyen CT, Nguyen KA. Dave the debater: a retrieval-based and generative argumentative dialogue agent. In: Proceedings of the 5th Workshop on Argument Mining. ACL; 2018. p. 121–130.

[11] Prakken H. Formal systems for persuasion dialogue. The knowledge engineering review. 2006;21(2):163–188.

[12] Prakken H. Historical overview of formal argumentation. vol. 1. College Publications; 2018.

[13] Bex F, Lawrence J, Reed C. Generalising argument dialogue with the Dialogue Game Execution Platform. In: COMMA; 2014. p. 141–152.

[14] Hunter A, Chalaguine L, Czernuszenko T, Hadoux E, Polberg S. Towards computational persuasion via natural language argumentation dialogues. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Springer; 2019. p. 18–33.

[15] Sakai K, Higashinaka R, Yoshikawa Y, Ishiguro H, Tomita J. Hierarchical Argumentation Structure for Persuasive Argumentative Dialogue Generation. IEICE TRANSACTIONS on Information and Systems. 2020;103(2):424–434.

[16] Mann WC. Dialogue games: Conventions of human interaction. Argumentation. 1988;2(4):511–532.

[17] Prakken H, Vreeswijk G. Logics for defeasible argumentation. In: Handbook of philosophical logic. Springer; 2001. p. 219–318.

[18] Stab C, Gurevych I. Annotating Argument Components and Relations in Persuasive Essays. In: COLING; 2014. p. 1501–1510.

[19] Rach N, Minker W, Ultes S. Markov Games for Persuasive Dialogue. In: COMMA; 2018. p. 213–220.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200517

Semantics Hierarchy in Preference-Based Argumentation Frameworks

Rafael SILVA [1], Samy SÁ and João ALCÂNTARA
Department of Computer Science, Federal University of Ceará, Brazil

Abstract. We define the pref-complete semantics for the Preference-Based Argumentation Frameworks (PAFs) of Amgoud and Vesic. The new semantics generalizes Dung's complete semantics for Argumentation Frameworks (AFs) in the same way that their original semantics (called pref-grounded, pref-stable, pref-preferred) respectively generalize the grounded, stable and preferred semantics for AFs. Additionally, we show that the pref-grounded/stable/preferred semantics are particular cases of the newly defined pref-complete semantics, therefore preserving the semantic hierarchy observed for AF semantics. This yields new ways for computing the semantics of PAFs, since the particular cases can be obtained from the pref-complete semantics with straightforward operations. Our contributions reinforce their thesis of backwards compatibility towards Dung's AF semantics.

Keywords. Abstract Argumentation, Preferences, Argumentation Semantics

1. Introduction

This work contributes to the thesis that the *Preference-Based Argumentation Frameworks* (PAFs) of Amgoud and Vesic [1] are backwards compatible to Dung's *abstract argumentation frameworks* (AFs) [2] concerning semantics. Their work can be found among several others [1,3,4,5,6,7,8,9] advocating that arguments do not always have the same strength and that, in some cases, the confidence one has in an argument could be enough to accept it despite reasons not to. In each case, these works (as well as [10,11]) approached how preferences over arguments in an AF should affect their evaluation, leading to different results.

To their advantage, [1] only retrieves *conflict-free* [2] sets of arguments in their preferential semantics. To that matter, they developed three preferential semantics respectively called *pref-grounded*, *pref-stable* and *pref-preferred* semantics, which respectively retrieve Dung's *grounded*, *stable* and *preferred* semantics [2] when the preferences over arguments cope with the attacks, but the *complete* semantics [2], commonly understood as the core AF semantics, was not addressed. The missing semantics is known to subsume the ones they approached, in the sense that the grounded, stable and preferred semantics are all particular cases of the complete semantics [12] for AFs. For this reason, had they defined the pref-complete semantics in [1], one would expect it to subsume the pref-grounded, pref-stable and pref-preferred semantics, and also to coincide with

[1] Corresponding Author: rafaels@lia.ufc.br

Dung's complete semantics when the preferences cope with the attacks. Our work means to close that gap, therefore we start by properly defining the *pref-complete* semantics for PAFs. Based on the new definition, we will prove that our preferential semantics generalizes Dung's complete semantics (following criteria from [1]) as expected. Further, we confirm that the pref-complete, pref-grounded, pref-stable and pref-preferred semantics for PAFs preserve the exact same hierarchy found between their corresponding AF semantics, even when the preferences influence the attack relation. Our results would also allow the results for the pref-grounded, pref-stable and pref-preferred semantics to be computed from the pref-complete extensions with straightforward operations, based on the confirmed hierarchy.

2. Preliminaries

We briefly review Argumentation Frameworks [2] and Preference-based Argumentation Frameworks as in [1] along with their semantics.

Definition 1 (Dung's framework)**.** *[2] An argumentation framework (AF) is a pair* $(\mathtt{A}r, att)$ *where* $\mathtt{A}r$ *is a set of arguments and* $att \subseteq \mathtt{A}r \times \mathtt{A}r$*.*

Arguments are related to others by the attack relation *att*: an argument a attacks b iff $(a,b) \in att$. An argumentation framework can be seen as a directed graph where the arguments are nodes and each attack is an arrow.

Definition 2 (defense/conflict-free)**.** *[2] Let* $\mathscr{F} = (\mathtt{A}r, att)$ *be an AF and* $\mathtt{E} \subseteq \mathtt{A}r$*. We say* $\mathtt{E}$ *is conflict-free iff* $\nexists a, b \in \mathtt{E}$ *such that* $(a,b) \in att$*. We will refer to* $\mathtt{CF}(\mathscr{F}) = \{\mathtt{E} \subseteq \mathtt{A}r \mid \mathtt{E}$ *is a conflict-free set of arguments w.r.t.* $\mathscr{F}\}$ *as the set of all conflict-free sets of arguments w.r.t.* $\mathscr{F}$*. We say* $\mathtt{E}$ defends *a iff every argument attacking a is attacked by some argument in* $\mathtt{E}$*. We define the characteristic function* $f : 2^{\mathtt{A}r} \to 2^{\mathtt{A}r}$ *of* $\mathscr{F}$ *as* $f(\mathtt{E}) = \{a \in \mathtt{A}r \mid \forall b \in \mathtt{A}r,\ if\ (b,a) \in att,\ then\ \exists c \in \mathtt{E}\ such\ that\ (c,b) \in att\}$ *to determine the set of all arguments defended by* $\mathtt{E}$*. We write* $\mathtt{E}^+ = \{a \in \mathtt{A}r \mid \exists b \in \mathtt{E}\ \ such\ that\ (b,a) \in att\}$ *to refer to the set of arguments attacked by* $\mathtt{E}$*.*

Traditional approaches to argumentation semantics are based on extensions of arguments. Some of the mainstream approaches are summarised below:

Definition 3 (Argumentation Semantics)**.** *[2,13] Let* $\mathscr{F} = (\mathtt{A}r, att)$ *be an AF,* $\mathtt{E}$ *be a conflict free subset of* $\mathtt{A}r$*, and f the characteristic function of* $\mathscr{F}$*. Then*

- $\mathtt{E}$ *is a complete extension of* $\mathscr{F}$ *iff* $f(\mathtt{E}) = \mathtt{E}$*.*
- $\mathtt{E}$ *is the grounded extension of* $\mathscr{F}$ *iff* $\mathtt{E}$ *is the* $\subseteq$*-minimal complete extension of* $\mathscr{F}$*.*
- $\mathtt{E}$ *is a preferred extension of* $\mathscr{F}$ *iff* $\mathtt{E}$ *is a* $\subseteq$*-maximal complete extension of* $\mathscr{F}$*.*
- $\mathtt{E}$ *is a stable extension of* $\mathscr{F}$ *iff* $\mathtt{E}$ *is a complete extension of* $\mathscr{F}$ *s.t.* $\mathtt{E} \cup \mathtt{E}^+ = \mathtt{A}r$*.*

Several works generalizing Dung's *AF* to handle preferences over arguments have been proposed [14,3,4,15,16,5,1]. In the so-called Preference-based Argumentation Frameworks (*PAF*s), preferences are used to represent the comparative strength of arguments. In *PAF*s, a critical scenario is what to do when the attacked argument b is stronger than its attacker a. In [16,7,9], they ignore those attacks and evaluate the arguments of *PAF* based only on the remaining attacks. This approach has been criticised by Amgoud and Vesic [1] as it leads to non-conflict-free extensions. As an alternative, they propose that the frameworks should instead be repaired by reversing the direction of those attacks.

Definition 4 (*PAF*). *[1] A Preference-based Argumentation Framework* (*PAF*) *is a tuple* $(\mathtt{Ar}, att, \geq)$ *s.t.* $\mathtt{Ar}$ *is a set of arguments,* $att \subseteq \mathtt{Ar} \times \mathtt{Ar}$*, and* $\geq$ *is a (partial/total) preorder.*

As in [1], we assume in this paper and without loss of generality that for a *PAF* $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$, $\mathtt{Ar}$ is finite and *att* does not contain self-attacking arguments. By $\mathtt{CF}(\mathfrak{T}) = \{\mathtt{E} \subseteq \mathtt{Ar} \mid \mathtt{E}$ *is conflict-free*$\}$, we denote the set of all conflict-free sets of arguments in $\mathfrak{T}$.

A distinguishing aspect of this approach is how a set of arguments defends an argument from other sets of arguments.

Definition 5 (Defense). *Let* $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ *be a PAF and* $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{Ar}$*. We say* $\mathtt{E}$ *defends* $a \in \mathtt{Ar}$ *from* $\mathtt{E}'$*, denoted by* $d(a, \mathtt{E}, \mathtt{E}')$*, iff* $\forall b \in \mathtt{E}'$ *if* $((b,a) \in att$ *and* $a \not> b)$ *or* $((a,b) \in att$ *and* $b > a)$*, then* $\exists c \in \mathtt{E}$ *s. t.* $((c,b) \in att$ *and* $b \not> c)$ *or* $((b,c) \in att$ *and* $c > b)$*.*

Still in [1], a semantics for evaluating arguments of a *PAF* is defined as a dominance relation $\succeq$ on $2^{\mathtt{Ar}}$. For $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{Ar}$, writing $\mathtt{E} \succeq \mathtt{E}'$ means that $\mathtt{E}$ is at least as good as $\mathtt{E}'$. By $\mathtt{E} \succ \mathtt{E}'$ we say that $\mathtt{E}$ is strictly better than $\mathtt{E}'$, i.e., that $\mathtt{E} \succeq \mathtt{E}'$ and $\mathtt{E}' \not\succeq \mathtt{E}$.

An *acceptability* semantics for a *PAF* $(\mathtt{Ar}, att, \geq)$ is defined by a dominance relation $\succeq \subseteq 2^{\mathtt{Ar}} \times 2^{\mathtt{Ar}}$ satisfying the postulates P_1, P_2 and P_3 that follow. Below, $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{Ar}$ and $a, a' \in \mathtt{Ar}$ and $\frac{X_1, \ldots, X_n}{Y}$ means that whenever $X_1, \ldots,$ and X_n hold, Y holds.

$$\frac{\mathtt{E} \in \mathtt{CF}(\mathfrak{T}) \quad \mathtt{E}' \notin \mathtt{CF}(\mathfrak{T})}{\mathtt{E} \succeq \mathtt{E}'} \qquad \frac{(a,a') \in att \quad (a',a) \notin att \quad \neg(a' > a)}{\{a\} \succ \{a'\}} \qquad \frac{(a,a') \in att \quad (a' > a)}{\{a'\} \succ \{a\}}$$

Postulate P_1 **Postulate P_2** **Postulate P_3**

The authors also defined three semantics for *PAF*s in [1]: pref-grounded, pref-stable, pref-preferred. We proceed by recalling the notion of strong defense, which will be employed in the characterisation of pref-grounded:

Definition 6 (Strong Defense). *[1] Let* $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ *be a PAF and* $\mathtt{E} \subseteq \mathtt{Ar}$*. We say* $\mathtt{E}$ *strongly defends an argument a from attacks of a set* $\mathtt{E}'$*, denoted by* $sd(a, \mathtt{E}, \mathtt{E}')$*, iff* $\forall b \in \mathtt{E}'$ *if* $((b,a) \in att$ *and* $a \not> b)$ *or* $((a,b) \in att$ *and* $b > a)$*, then* $\exists c \in \mathtt{E} \backslash \{a\}$ *such that* $((c,b) \in att$ *and* $b \not> c)$ *or* $((b,c) \in att$ *and* $c > b)$ *and* $sd(c, \mathtt{E} \backslash \{a\}, \mathtt{E}')$*.*

Intuitively, an argument is strongly defended when it is preferred to its attackers or it is defended by another argument that is strongly defended without the argument in question. In [1], the extensions of a semantics $\succeq$ are then given by its maximal elements:

Definition 7 ((Maximal) Upper Bounds). *Let* $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ *be a PAF,* $\mathtt{E} \subseteq \mathtt{Ar}$ *and* $\succeq \subseteq 2^{\mathtt{Ar}} \times 2^{\mathtt{Ar}}$ *a semantics for PAF. We say* $\mathtt{E}$ *is an upper bound wrt* $\succeq$ *iff* $\forall \mathtt{E}' \in 2^{\mathtt{Ar}}, \mathtt{E} \succeq \mathtt{E}'$*. Besides, if no strict superset of* $\mathtt{E}$ *is an upper bound wrt* $\succeq$*, then* $\mathtt{E}$ *is a maximal wrt* $\succeq$*. Let* $\succeq_{ub}$ *and* $\succeq_{max}$ *denote respectively the set of upper bound and maximal sets w.r.t.* $\succeq$*.*

We are ready to define the pref-grounded, pref-stable and pref-preferred semantics:

Definition 8 (Pref-grounded semantics). *[1] Let* $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ *be a PAF and* $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{Ar}$*. It holds that* $\mathtt{E} \succeq_g \mathtt{E}'$ *iff a)* $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ *and* $\mathtt{E}' \notin \mathtt{CF}(\mathfrak{T})$*, or b)* $\forall a \in \mathtt{E}$*, it holds* $sd(a, \mathtt{E}, \mathtt{E}')$*.*

Definition 9 (Pref-stable semantics)**.** *[1] Let $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ be a PAF and $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{A}r$. It holds that $\mathtt{E} \succeq_s \mathtt{E}'$ iff a) $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\mathtt{E}' \notin \mathtt{CF}(\mathfrak{T})$, or b) $\mathtt{E}, \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$ and $\forall b \in \mathtt{E}' \backslash \mathtt{E}$, $\exists a \in \mathtt{E} \backslash \mathtt{E}'$ s.t. $((a,b) \in att$ and $b \not> a)$ or $(a > b)$.*

Definition 10 (Pref-preferred semantics)**.** *[1] Let $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ be a PAF and $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{A}r$. It holds that $\mathtt{E} \succeq_p \mathtt{E}'$ iff a) $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\mathtt{E}' \notin \mathtt{CF}(\mathfrak{T})$, or b) $\mathtt{E}, \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$ and $\forall a \in \mathtt{E}, \forall b \in \mathtt{E}'$, if $((b,a) \in att$ and $a \not> b)$ or $((a,b) \in att$ and $b > a)$, then $\exists c \in \mathtt{E}$ such that $((c,b) \in att$ and $b \not> c)$ or $((b,c) \in att$ and $c > b)$.*

We say $\mathtt{E} \subseteq \mathtt{A}r$ is a pref-grounded, pref-stable or a pref-preferred extension iff it is respectively maximal (Definition 7) with respect to $\succeq_g$, $\succeq_s$ and $\succeq_p$. By $\succeq_{g,max}$, $\succeq_{s,max}$ and $\succeq_{p,max}$, we denote respectively the set of maximal sets w.r.t. $\succeq_g$, $\succeq_s$ and $\succeq_p$.

Example 1. *Let $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ be a PAF such that $\mathtt{A}r = \{a,b,c,d,e,f\}$, $att = \{(a,b), (b,c), (c,a), (d,e), (d,f), (e,a), (f,d)\}$ and $(a > e)$. Its sets of extensions are $\succeq_{g,max} = \{\emptyset\}$, $\succeq_{s,max} = \emptyset$ and $\succeq_{p,max} = \{\{d\}, \{f\}\}$.*

According to [1], pref-grounded, pref-stable and pref-preferred coincide respectively with grounded, stable and preferred when the available preferences do not conflict with the attacks. Note also that instead of partitioning the powerset of the set of arguments into extensions and non-extensions as usual in the definition of the semantics for *AF*, this approach is more informative as it compares all the subsets of arguments.

Next, we define a new semantics for *PAF*, namely the *pref-complete* semantics. We will proceed to show that the relations between the pref-complete extensions and the pref-grounded, pref-stable and pref-preferred extensions are respectively the same as the relations between complete extensions and grounded, stable and preferred extensions.

3. Complete Semantics for *PAF*s

In this section, we will define the pref-complete semantics $\succeq_c$ for *PAF*s, designed to coincide with the complete semantics for *AF* when preferences are ignored. The challenge behind this goal is that, differently from $\succeq_g$, $\succeq_p$, and $\succeq_s$, the extensions of $\succeq_c$ cannot be defined in terms of only its maximal elements. For instance, the complete extensions of $AF = (\{a,b\}, \{(a,b),(b,a)\})$ are $\emptyset$, $\{a\}$, $\{b\}$, amongst which $\emptyset$ fails maximality. As we will show, the extensions of $\succeq_c$ are instead characterized by its upper bounds.

Definition 11 (Pref-complete semantics)**.** *Let $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ be a PAF and $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{A}r$. It holds that $\mathtt{E} \succeq_c \mathtt{E}'$ iff a) $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\mathtt{E}' \notin \mathtt{CF}(\mathfrak{T})$ or b) $\mathtt{E}, \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$ and $\mathtt{E} \subseteq \{a \in \mathtt{A}r | d(a, \mathtt{E}, \mathtt{E}')\}$ and if $\mathtt{E} \subseteq \mathtt{E}'$, then $(\{a \in \mathtt{A}r | d(a, \mathtt{E}, \mathtt{A}r)\} - \mathtt{E}) \subseteq (\{a \in \mathtt{A}r | d(a, \mathtt{E}', \mathtt{A}r)\} - \mathtt{E}')$.*

We define $\succeq_{c,ub} = \{\mathtt{E} \subseteq \mathtt{A}r \mid \mathtt{E}$ is an upper bound w.r.t. $\succeq_c\}$. A set $\mathtt{E}$ is a pref-complete extension of $\mathfrak{T}$ iff $\mathtt{E} \in \succeq_{c,ub}$.

Note that when $\mathtt{E}, \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$, it holds $\mathtt{E} \succeq_c \mathtt{E}'$ iff $\mathtt{E}$ defends all its elements from the attacks of $\mathtt{E}'$, and if $\mathtt{E} \subseteq \mathtt{E}'$, those extra elements defended by $\mathtt{E}$ beyond the elements in $\mathtt{E}$ are also defended by $\mathtt{E}'$. In particular, if $\mathtt{E}$ is conflict-free and the set of elements it defends is exactly $\mathtt{E}$, then $\mathtt{E}$ is a pref-complete extension. Recalling the *PAF* $\mathfrak{T}$ in Example 1, we obtain the set of its pref-complete extensions is $\succeq_{c,ub} = \{\emptyset, \{d\}, \{f\}\}$.

It is clear that $\succeq_c$ is an acceptability semantics.

Proposition 1. *The relation $\succeq_c$ satisfies postulates P_1, P_2 and P_3.*

The next definition describes semantics generalization. We will employ it to prove that the pref-complete semantics generalizes Dung's complete semantics.

Definition 12 (Generalizing a semantics). *A semantics $\succeq$ for PAF generalizes a semantics S for AF iff for all PAF $(\mathtt{Ar}, att, \geq)$, such that $\nexists a, b \in \mathtt{Ar}$ with $(a,b) \in att$ and $b > a$, it holds $\mathtt{E} \in \succeq_{ub}$ iff $\mathtt{E}$ is an extension of $\mathscr{F} = (\mathtt{Ar}, att)$ according to S.*

As expected, Proposition 2 guarantees pref-complete extensions are conflict-free:

Proposition 2. *Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a PAF and $\mathtt{E} \subseteq \mathtt{Ar}$. If $\mathtt{E} \in \succeq_{c,ub}$, then $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$.*

The next result will help us prove that the pref-complete semantics generalizes Dung's complete semantics.

Lemma 1. *Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a PAF, in which $\nexists a, b \in \mathtt{Ar}$ such that $(a,b) \in att$ and $b > a$. For any $\mathtt{E} \subseteq \mathtt{Ar}$, it holds $\{a \in \mathtt{Ar} \mid d(a, \mathtt{E}, \mathtt{Ar})\} = f(\mathtt{E})$. Besides, for each $\mathtt{E}' \subseteq \mathtt{Ar}$, it holds $f(\mathtt{E}) \subseteq \{a \in \mathtt{Ar} \mid d(a, \mathtt{E}, \mathtt{E}')\}$.*

Theorem 1. *The relation $\succeq_c$ generalises complete semantics.*

Proof. (*sketch*) Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a *PAF*, in which $\nexists a, b \in \mathtt{Ar}$ such that $(a,b) \in att$ and $b > a$. We will prove $\mathtt{E}$ is a complete extension of $\mathscr{F} = (\mathtt{Ar}, att)$ iff $\mathtt{E} \in \succeq_{c,ub}$: it holds $\mathtt{E}$ is a complete extension of $\mathscr{F}$ iff $\mathtt{E} \in \mathtt{CF}(\mathscr{F})$ and $f(\mathtt{E}) = \mathtt{E}$ iff (Lemma 1) $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\forall \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$, $\mathtt{E} \subseteq \{a \in \mathtt{Ar} | d(a, \mathtt{E}, \mathtt{E}')\}$ and $\{a \in \mathtt{Ar} | d(a, \mathtt{E}, \mathtt{Ar})\} = \mathtt{E}$ iff $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\forall \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$, $\mathtt{E} \subseteq \{a \in \mathtt{Ar} | d(a, \mathtt{E}, \mathtt{E}')\}$ and if $\mathtt{E} \subseteq \mathtt{E}'$, then $(\{a \in \mathtt{Ar} | d(a, \mathtt{E}, \mathtt{Ar})\} - \mathtt{E}) = \emptyset \subseteq (\{a \in \mathtt{Ar} | d(a, \mathtt{E}', \mathtt{Ar})\} - \mathtt{E}')$ iff $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ and $\forall \mathtt{E}' \subseteq \mathtt{Ar}$, $\mathtt{E} \succeq_c \mathtt{E}'$ iff $\mathtt{E} \in \succeq_{c,ub}$. □

4. The pref-Semantics Satisfies the Classical AF Semantics Hierarchy

In this section, we show that the pref-grounded, pref-stable and pref-preferred semantics are particular cases of the pref-complete semantics in the same way that the grounded, stable and preferred AF semantics are particular cases of the complete AF semantics. Therefore, we show that the semantic hierarchy of AFs is entirely preserved by the semantics defined for PAFs. Timely, we highlight this result holds for all PAFs, independently of what preferences one has over arguments.

Regarding the successful attacks (defeats), we have the *AF* corresponding to a *PAF*:

Definition 13 (Defeat). *[1] Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a PAF and $a, b \in \mathtt{Ar}$. We say a defeats b in $\mathfrak{T}$ if $((a,b) \in att$ and $b \not> a)$ or $((b,a) \in att$ and $a > b)$. We will refer to $(\mathtt{Ar}, \mathscr{D})$ as the AF corresponding to $\mathfrak{T}$, in which $\mathscr{D} = \{(a,b) | a, b \in \mathtt{Ar}$ and a defeats b in $\mathfrak{T}\}$.*

We show the arguments defended by a set of arguments $\mathtt{E}$ via d operator in a *PAF* $\mathfrak{T}$ are the same as those defended by $\mathtt{E}$ via f operator in the *AF* corresponding to $\mathfrak{T}$:

Lemma 2. *Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a PAF, $a \in \mathtt{Ar}$ and $(\mathtt{Ar}, \mathscr{D})$ the corresponding argumentation framework to $\mathfrak{T}$. We have $d(a, \mathtt{E}, \mathtt{Ar})$ in $(\mathtt{Ar}, att, \geq)$ iff $a \in f(\mathtt{E})$ in $(\mathtt{Ar}, \mathscr{D})$.*

Lemma 3 shows a pref-complete extension is equal the set of arguments it defends:

Lemma 3. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF and* $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$. *It holds* $\mathtt{E} \in \succeq_{c,ub}$ *if and only if* $\{a' \in \mathtt{A}r | d(a', \mathtt{E}, \mathtt{A}r)\} = \mathtt{E}$.

Now we ensure the pref-complete extensions of a *PAF* are the complete extensions of the corresponding *AF*:

Theorem 2. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF and* $\mathscr{T}^d = (\mathtt{A}r, \mathscr{D})$ *the corresponding argumentation framework. We have that* $\mathtt{E} \in \succeq_{c,ub}$ *iff* $\mathtt{E}$ *is a complete extension of* $\mathscr{T}^d$.

Proof. $\mathtt{E}$ is a complete extension of $\mathscr{T}^d$ iff $f(\mathtt{E}) = \mathtt{E}$ in $\mathscr{T}^d$ and $\mathtt{E} \in \mathtt{CF}(\mathscr{T}^d)$ iff (Lemma 2) $\mathtt{E} = \{a \in \mathtt{A}r \mid d(a, \mathtt{E}, \mathtt{A}r)\}$ and $\mathtt{E} \in \mathtt{CF}(\mathscr{T}^d)$ iff (Lemma 3) $\mathtt{E} \in \succeq_{c,ub}$ and $\mathtt{E} \in \mathtt{CF}(\mathscr{T}^d)$ iff (Proposition 2) $\mathtt{E} \in \succeq_{c,ub}$. □

Theorems 3, 4 and 5 show respectively pref-grounded, pref-stable and pref-preferred extensions can be depicted via pref-complete extensions in the same way grounded, stable and preferred extensions can be depicted via complete extensions. Theorem 3 follows immediately from Theorem 2 and the fact $\mathtt{E}$ is the pref-grounded extension of a *PAF* iff $\mathtt{E}$ is the grounded extension of the corresponding *AF* (see [1]):

Theorem 3. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF. It holds* $\mathtt{E}$ *is the minimal (w.r.t. set inclusion) pref-complete extension of* $\mathfrak{T}$ *iff* $\mathtt{E}$ *is the pref-grounded extension of* $\mathfrak{T}$.

In the remaining of this section, for a dominance order $\succeq$ in the context of a *PAF* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$, we will write $\succeq^{\mathfrak{T}}$ to indicate the reference framework. By $\mathscr{T}^d = (\mathtt{A}r, \mathscr{D})$ we mean the corresponding argumentation framework and we assume $\mathfrak{T}^r = (\mathtt{A}r, \mathscr{D}, \geq)$.

Theorem 4. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF,* $\mathscr{T}^d = (\mathtt{A}r, \mathscr{D})$ *be the corresponding argumentation framework,* $\mathtt{E} \subseteq \mathtt{A}r$ *and* $\mathtt{E}^+ = \{a \in \mathtt{A}r \mid \exists b \in \mathtt{E}$ *s. t.* $(b, a) \in \mathscr{D}\}$. *It holds* $\mathtt{E}$ *is a pref-complete extension of* $\mathfrak{T}$ *such that* $\mathtt{E} \cup \mathtt{E}^+ = \mathtt{A}r$ *iff* $\mathtt{E}$ *is a pref-stable extension of* $\mathfrak{T}$.

Proof. It holds $\mathtt{E} \in \succeq_{s,max}$ iff (Theorem 11 in [1]) $\mathtt{E}$ is stable in $\mathscr{T}^d$ iff (according to [2]) $\mathtt{E}$ is complete in $\mathscr{T}^d$ and $\mathtt{E} \cup \mathtt{E}^+ = \mathtt{A}r$ iff (Theorem 2) $\mathtt{E} \in \succeq_{c,ub}$ and $\mathtt{E} \cup \mathtt{E}^+ = \mathtt{A}r$. □

The next lemmas are employed to prove Theorem 5:

Lemma 4. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF and* $\mathtt{E}, \mathtt{E}' \subseteq \mathtt{A}r$. *Then* $\mathtt{E} \succeq_p^{\mathfrak{T}} \mathtt{E}'$ *iff* $\mathtt{E} \succeq_p^{\mathfrak{T}^r} \mathtt{E}'$.

Proof. As $\mathtt{E} \in \mathtt{CF}(\mathfrak{T})$ iff $\mathtt{E} \in \mathtt{CF}(\mathfrak{T}^r)$, it is sufficient to consider the case where $\mathtt{E}, \mathtt{E}' \in \mathtt{CF}(\mathfrak{T})$. We have $\mathtt{E} \succeq_p^{\mathfrak{T}} \mathtt{E}'$ iff $\forall a \in \mathtt{E}, \forall b \in \mathtt{E}'$ if $((b, a) \in att$ and $a \not> b)$ or $((a, b) \in att$ and $b > a)$, then $\exists c \in \mathtt{E}$ such that $((c, b) \in att$ and $b \not> c)$ or $((b, c) \in att$ and $c > b)$ iff $\forall a \in \mathtt{E}, \forall b \in \mathtt{E}'$, if $(b, a) \in \mathscr{D}$ then $\exists c \in \mathtt{E}$ such that $(c, b) \in \mathscr{D}$ iff $\forall a \in \mathtt{E}, \forall b \in \mathtt{E}'$ if $((b, a) \in \mathscr{D}$ and $a \not> b)$ (or the impossible case where $(a, b) \in \mathscr{D}$ and $b > a)$ then $\exists c \in \mathtt{E}$ such that $((c, b) \in \mathscr{D}$ and $b \not> c)$ (or the impossible case where $(b, c) \in \mathscr{D}$ and $c > b)$ iff $\mathtt{E} \succeq_p^{\mathfrak{T}^r} \mathtt{E}'$. □

Lemma 5. *Let* $\mathfrak{T} = (\mathtt{A}r, att, \geq)$ *be a PAF and* $\mathtt{E} \subseteq \mathtt{A}r$. *We have* $\mathtt{E}$ *is a preferred extension of* $\mathscr{T}^d$ *iff* $\mathtt{E} \in \succeq_{p,max}^{\mathfrak{T}^r}$ *iff* $\mathtt{E} \in \succeq_{p,max}^{\mathfrak{T}}$.

Proof. We have $\mathtt{E}$ is a preferred extension of $\mathscr{T}^d$ iff (Theorem 3 from [1]) $\mathtt{E} \in \succeq_{p,max}^{\mathfrak{T}^r}$ iff (Lemma 4) $\mathtt{E} \in \succeq_{p,max}^{\mathfrak{T}}$. □

Theorem 5. *Let $\mathfrak{T} = (\mathtt{Ar}, att, \geq)$ be a PAF. A pref-complete extension $\mathtt{E}$ of $\mathfrak{T}$ is $\subseteq$-maximal among all $\mathtt{E} \in \succeq_{c,ub}$ iff $\mathtt{E}$ is a pref-preferred extension of $\mathfrak{T}$.*

Proof. Let $\mathscr{T}^d = (\mathtt{Ar}, \mathscr{D})$ be the corresponding argumentation framework to $\mathfrak{T}$. We have $\mathtt{E}$ is a $\subset$-maximal pref-complete extension in $\mathfrak{T}$ iff (Theorem 2) $\mathtt{E}$ is a $\subset$-maximal complete extension of $\mathscr{T}^d$ iff (according to [2]) $\mathtt{E}$ is a preferred extension of $\mathscr{T}^d$ iff (Lemma 5) $\mathtt{E}$ is a pref-preferred extension of $\mathfrak{T}$. □

5. Conclusion

The literature on preferences in argumentation is rich with different approaches, lacking consensus on a standard. The disagreement can be backtracked to a critical scenario where an attacked argument (in the sense of [2]) is deemed stronger or preferred over its attackers. Here, we contributed to the debate showing that a prominent approach, namely that of Amgoud and Vesic [1], retains the hierarchy of admissibility-based semantics established in [12]. This result is not straightforward, since [1] did not provide a preferential semantics corresponding to Dung's complete semantics, which is often considered the core AF semantics. For this reason, we started by defining the pref-complete semantics $\succeq_c$ for the Preference-based Argumentation Frameworks (*PAF*s) of [1]. Here, we associated the pref-complete extensions with the upper bounds of $\succeq_c$ and showed that it adequately generalizes Dung's complete semantics for *AF*s. The new semantics allowed us to establish a proper hierarchy among preferential semantics for *PAF*s from [1], showing they preserve the same subsumption relations as the *AF* semantics.

While there is a general agreement that conflict-freeness should be respected by the semantics of AFs with preferences, works such as [11,17,10] criticized the solution of [1]. In [11], Kaci et. al. propose that the attack (A,B) should be ignored only if it is symmetric, i.e., if B also attacks A, otherwise it should remain unchanged. This choice leaves room for an attack from a less preferred argument to still be successful, which is debatable. For comparison, the PAF $(\{A,B\}, \{(A,B)\}, \{(B,A)\})$, has the unique complete extension $\{A\}$ according to [11] and $\{B\}$ according to [1]. In [17], Wakaki ensures that extensions of a *PAF* $(\mathtt{Ar}, att, \geq)$ are extensions of its base *AF* $(\mathtt{Ar}, att)$. Instead of changing the attack relation, they simply select what extensions of *AF* respect the preferences. For comparison, the *PAF* $(\{A,B\}, \{(A,B),(B,A)\}, \{(B,A)\})$ has the complete extensions $\emptyset$ and $\{B\}$ according to [17] (notice $\emptyset$ is grounded) and only $\{B\}$ according to [1]. In [10], Modgil and Prakken focused on preferences in ASPIC$^+$ [18]. They argue the structure of arguments and the nature of attacks should be considered when applying preferences, adding more conditions to the reversal of the attacks that do not satisfy preferences.

Here, we do not advocate that Amgoud and Vesic's approach [1] would be the best available, but instead that the special cases of [11,17,10] are also worthwhile investigating. In our view, the divergences between them occur simply because they model different notions of preferences, each deserving attention on its own. In future works we will extend our investigation to verify whether other proposals of preference-based argumentation also preserve the semantic hierarchy observed among Dung's semantics. Another promising venture inspired by Wakaki's work [17] involves adapting other approaches of preferences from logic programming (see [19] for a survey) based on the mappings between abstract argumentation frameworks and logic programs found in [20].

References

[1] Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 63(2):149–183, 2011.

[2] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[3] Guillermo R Simari and Ronald P Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial intelligence*, 53(2-3):125–157, 1992.

[4] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Uncertainty in Artificial Intelligence*, pages 411–419. Elsevier, 1993.

[5] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of applied non-classical logics*, 7(1-2):25–75, 1997.

[6] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.

[7] Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[8] Leila Amgoud and Srdjan Vesic. Repairing preference-based argumentation frameworks. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, page 665–670, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[9] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial intelligence*, 173(9-10):901–934, 2009.

[10] Sanjay Modgil and Henry Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[11] Souhila Kaci, Leendert van der Torre, and Serena Villata. Preference in abstract argumentation. In *7th International Conference on Computational Models of Argument (COMMA)*, volume 305, pages 405–412. IOS Press, 2018.

[12] Martin WA Caminada and Dov M Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109, 2009.

[13] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 Conference on Computational Models of Argument: Proceedings of COMMA 2006*, page 121–130, NLD, 2006. IOS Press.

[14] Claudette Cayrol, Véronique Royer, and Claire Saurel. Management of preferences in assumption-based reasoning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 13–22. Springer, 1992.

[15] Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48(3):730–751, 2008.

[16] Leila Amgoud, Claudette Cayrol, and Daniel Le Berre. Comparing arguments using preference orderings for argument-based reasoning. In *Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence*, pages 400–403. IEEE, 1996.

[17] Toshiko Wakaki. Preference-based argumentation built from prioritized logic programming. *Journal of Logic and Computation*, 25(2):251–301, 2015.

[18] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[19] James Delgrande, Torsten Schaub, Hans Tompits, and Kewen Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 20(2):308–334, 2004.

[20] Martin Caminada, Samy Sá, João Alcântara, and Wolfgang Dvořák. On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning*, 58:87–111, 2015.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200518

Abstract Argumentation Frameworks with Fallible Evidence

Kenneth SKIBA [a,1], Matthias THIMM [a], Andrea COHEN [b], Sebastian GOTTIFREDI [b], and Alejandro J. GARCÍA [b]

[a] *University of Koblenz-Landau, Germany*
[b] *Universidad Nacional del Sur, CONICET, Argentina*

Abstract. We consider a generalisation of abstract argumentation frameworks where arguments need to be backed by pieces of evidence in order to be actually present in the argumentation framework. These pieces of evidence come with an associated cost for retrieval and may not be available at any given time. We model an information-seeking agent in this scenario that aims at deciding whether a certain argument is acceptable while minimising the total evidence retrieval cost. We investigate the computational complexity of decision variants of this optimisation problem and find that, depending on the underlying classical argumentation semantics, complexity rises one level in the polynomial hierarchy compared to the classical case.

Keywords. abstract argumentation, computational complexity, evidence retrieval cost

1. Introduction

Computational models of argumentation [1] aim at modelling rational decision-making through the representation of arguments and their relationships. In particular, abstract argumentation frameworks [6] provide a simple representation formalism of such situations by focusing on the representation of arguments and a conflict relation between arguments through modeling this setting as a directed graph. Here, arguments are identified by vertices and an *attack* from one argument to another is represented as a directed edge. This simple model already provides an interesting object of study, see [2] for an overview. Several extensions of this model have been investigated as well, such as considering an additional support relation [5], recursive interactions [9], attacks by sets of arguments [10], and others.

In this paper, we investigate yet another extension of abstract argumentation. In many real-life application scenarios for argumentation, arguments are not standalone entities but rely on pieces of *evidence* in order to be *active* in an argumentation [4, 12]. Consider an online discussion forum and a discussion about the

[1] Corresponding Author: Kenneth Skiba, University of Koblenz-Landau, Germany; E-mail: kennethskiba@uni-koblenz.de.

spread of the COVID-19 virus. A specific argument in this context could be "The COVID-19 virus is a serious danger because (1) its incubation phase is quite long and (2) infected people can easily infect other people during this phase. Moreover, although (3) the mortality rate is quite small, (4) the impact on public health systems and the elderly can be severe". In order for this argument to be believable at all, facts (1), (2), (3), and (4) need to be backed by some evidence. For example, the author of that argument can link its argument to some articles from the World Health Organisation (WHO) or other resources of authority, also providing concrete numbers to the imprecise facts of the argument. If the recipient of the argument wishes to assess the validity of the argument, she can visit the linked pieces of evidence and verify the claims. However, this verification step involves time and effort, so pieces of evidence usually come with an associated cost (such as time) that need to be spent in order to verify the argument. Moreover, retrieval of pieces of evidence may fail, because a web server may be down or the article is no longer available.

Here, we model scenarios such as the one outlined above by extending abstract argumentation frameworks through the addition of pieces of evidence, a function associating arguments with pieces of evidence and a function determining their associated cost, generalising the formalisation of [4]. More precisely, the contributions of this paper are as follows:

1. We present Abstract Argumentation Frameworks with Fallible Evidence (AAFE) as an extension to Dung's abstract argumentation frameworks that take evidence for arguments into account (Section 3).
2. We investigate the computational complexity of a certain optimisation problem within our new setting (Section 4).

We also provide necessary preliminaries on abstract argumentation in Section 2, discuss related works in Section 5, and conclude in Section 6.

2. Abstract Argumentation

Following [6], an *(abstract) argumentation framework* AF is a pair $(\mathcal{A}, \mathcal{R})$, where $\mathcal{A}$ is a finite set of arguments and $\mathcal{R}$ is a set of attacks between arguments, i. e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. An argument a is said to *attack* b if $(a,b) \in \mathcal{R}$. We call an argument a *acceptable with respect to a set* $S \subseteq \mathcal{A}$ if for each $b \in \mathcal{A}$ with $(b,a) \in \mathcal{R}$, there is an argument $c \in S$ with $(c,b) \in \mathcal{R}$. An argumentation framework $(\mathcal{A}, \mathcal{R})$ can be illustrated by a directed graph with vertex set $\mathcal{A}$ and edge set $\mathcal{R}$.

For an argumentation framework $AF = (\mathcal{A}, \mathcal{R})$ and a set $\mathcal{A}' \subseteq \mathcal{A}$ we define the *projection* $AF_{\mathcal{A}'}$ of AF onto $\mathcal{A}'$ via $AF_{\mathcal{A}'} = (\mathcal{A}', \mathcal{R} \cap (\mathcal{A}' \times \mathcal{A}'))$.

Semantics are given to argumentation frameworks by means of *extensions*, i. e., sets of mutually acceptable arguments. A set $S \subseteq \mathcal{A}$ is *conflict-free* (CF) if there are no arguments a and b in S such that $(a,b) \in \mathcal{R}$. We call a conflict-free set S *admissible* (AD) if every argument $a \in S$ is acceptable with respect to S.

Definition 1. Let $(\mathcal{A}, \mathcal{R})$ be an argumentation framework and $S \subseteq \mathcal{A}$.

- S is a *complete extension* (CP) if it is admissible and contains every argument that is acceptable with respect to S.

- S is the *grounded extension* (GR) if it is the minimal complete extension (wrt. set inclusion).
- S is a *preferred extension* (PR) if it is a maximal complete extension (wrt. set inclusion).
- S is a *stable extension* (ST) if it is conflict-free and for each $b \in \mathcal{A} \setminus S$, there is at least one argument $a \in S$ such that $(a, b) \in \mathcal{R}$.

Note that the grounded extension is uniquely defined and stable extensions may not exist [6]. Given an argumentation framework $(\mathcal{A}, \mathcal{R})$ and an argument $a \in \mathcal{A}$, we say that a is *cred*ulously (*skep*tically) accepted under the semantics σ if it is contained in at least one σ-extension (all σ-extensions) of $(\mathcal{A}, \mathcal{R})$, respectively.

3. Abstract Argumentation with Fallible Evidence

We now define abstract argumentation frameworks with fallible evidence (abbreviated AAFE) as a generalisation of abstract argumentation frameworks. In this generalisation, each argument is associated with a set of evidence, which models observations needed to be present in order to make the argument "active". Each evidence comes with an associated cost that needs to be paid in order to attempt to retrieve the evidence. This cost must also be paid if it turns out that the evidence is not available.

Example 1. Consider the following arguments exchanged by doctors trying to diagnose a patient:

a: If the patient shows the set S_1 of signs and symptoms, there are indications that the patient has the disease D.

b: If the patient also shows the set S_2 of signs and symptoms, he could have disease D' instead. We can perform a high sensitivity test T_1 for D'. If the result is positive, then we have reasons to diagnose the patient with D'.

c: If T_1 is positive, a high specificity test T_2 for D' has to be performed. If T_2's result is negative, then we can refrain from diagnosing the patient with D'.

d: We can also run a high sensitivity test T_3 for D. If we get a negative result, then we can refrain from diagnosing the patient with D.

The arguments are based on different sets of signs and symptoms and/or the results of different tests being performed over the patient. Then, these observations are the pieces of evidence the arguments are based on and, furthermore, they come with an associated cost. On the one hand, the doctors have to spend some time in order to identify the sets S_1 and S_2. On the other hand, the cost for performing each test has to be paid (both time and money), regardless of whether the result is as expected (as specified by the corresponding piece of evidence) or not; in cases where the tests' outcome are not as expected, we can consider that the corresponding pieces of evidence are unavailable and cannot be retrieved.

We define *abstract argumentation frameworks with fallible evidence* as follows.

Definition 2. An *abstract argumentation framework with fallible evidence* (AAFE) F is a tuple $F = (\mathcal{A}, \mathcal{R}, \mathcal{E}, \delta, \mu)$ where $\mathcal{A}$ is a set of arguments, $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is the attack relation, $\mathcal{E}$ is a set of evidence, $\delta : A \to 2^{\mathcal{E}}$ assigns to each argument a set of evidence, and $\mu : \mathcal{E} \to \mathbb{N}$ is the evidence cost function.

The function μ is extended to sets of evidence $E \subseteq \mathcal{E}$ via $\mu(E) = \sum_{e \in E} \mu(e)$. Furthermore, by abusing notation, for a set of evidence $E \subseteq \mathcal{E}$, we write $\delta^{-1}(E) = \{a \in \mathcal{A} \mid \delta(a) \subseteq E\}$.

Given a concrete set of evidence $E \subseteq \mathcal{E}$, an AAFE $F = (\mathcal{A}, \mathcal{R}, \mathcal{E}, \delta, \mu)$ is instantiated to an abstract argumentation framework $F_E = (\mathcal{A}_E, \mathcal{R}_E)$ by projecting on the arguments activated by the set of evidence, i. e. $F_E = (\mathcal{A}, \mathcal{R})_{\delta^{-1}(E)}$. We assume that not all pieces of evidence in $\mathcal{E}$ can actually be retrieved, let $\hat{\mathcal{E}} \subseteq \mathcal{E}$ denote the set of evidence which is actually available. We now consider an agent Ag that has to inquire whether a certain argument $a_{query} \in \mathcal{A}$ is credulously or skeptically accepted in $F_{\hat{\mathcal{E}}}$ (i. e. in the framework obtained by projecting on the set of active arguments whose evidence is available) with respect to a semantics σ while $\hat{\mathcal{E}}$ is not known to Ag.

We denote by $\Delta^{\circ}_{\sigma}(F, a) \in \{y, n, na\}$ (yes, no, not active) the acceptance status of a in F with respect to the semantics σ and the problem $\circ \in \{\text{cred}, \text{skep}\}$; the first two labels correspond to active arguments belonging to $\delta^{-1}(\hat{\mathcal{E}})$, whereas the latter corresponds to arguments for which the acceptance status cannot be determined because they are inactive.

The agent can ask for every piece of evidence $e \in \mathcal{E}$ by paying its cost $\mu(e)$ and, if $e \in \hat{\mathcal{E}}$, then he also retrieves e and activates the corresponding arguments. Of course, Ag should be economic and only ask for as little evidence as required (set $E_{\mathsf{Ag}} \subseteq \mathcal{E}$) in order to make sure that for the final set of evidence E that he actually collected (i. e. $E = E_{\mathsf{Ag}} \cap \hat{\mathcal{E}}$) we have that both frameworks $F_{\hat{\mathcal{E}}}$ and F_E yield the same answer regarding a_{query} and he paid as little cost as possible. Formally, given $F = (\mathcal{A}, \mathcal{R}, \mathcal{E}, \delta, \mu)$ and an argument $a_{query} \in \mathcal{A}$, Ag must solve the following optimisation problem:

Minimise $\mu(E_{\mathsf{Ag}})$ such that for every argument $a \in \mathcal{A} \setminus \delta^{-1}(E_{\mathsf{Ag}} \cap \hat{\mathcal{E}})$ with

$$\Delta^{\circ}_{\sigma}((\mathcal{A}, \mathcal{R})_{\delta^{-1}(E_{\mathsf{Ag}} \cap \hat{\mathcal{E}})}, a_{query}) \neq \Delta^{\circ}_{\sigma}((\mathcal{A}, \mathcal{R})_{\delta^{-1}(E_{\mathsf{Ag}} \cap \hat{\mathcal{E}}) \cup \{a\}}, a_{query})$$

we have $E_{\mathsf{Ag}} \cap \delta(a) \cap (\mathcal{E} \setminus \hat{\mathcal{E}}) \neq \emptyset$.

In other words, Ag seeks to determine the cheapest set of evidence E_{Ag} such that the acceptances status of a_{query} does not change. For that Ag made sure that for every argument a that cannot be constructed from this set of evidence (specifically, from its subset of available evidence) and would change the acceptance status of a_{query}, he asks for at least one piece of evidence of a which cannot be retrieved because it is not available. We denote the optimal solution of the above problem as $OPT^{\circ,\sigma}_{F,\hat{\mathcal{E}}}(a_{query})$.

Example 2. Consider the argumentation framework with fallible evidence $F = (\mathcal{A}, \mathcal{R}, \mathcal{E}, \delta, \mu)$ depicted in Figure 1 and defined via

$$\mathcal{A} = \{a, b, c, d, e, f\}$$

$$\mathcal{R} = \{(a, b), (b, c), (d, c), (e, c), (e, d), (f, e)\}$$

$$\mathcal{E} = \{e_1, e_2, e_3, e_4\}$$

$$\delta(a) = \{e_1\} \qquad \delta(b) = \{e_1, e_2\} \qquad \delta(c) = \{e_2\}$$

$$\delta(d) = \{e_3\} \qquad \delta(e) = \{e_3, e_4\} \qquad \delta(f) = \{e_3, e_4\}$$

$$\mu(e_1) = 2 \qquad \mu(e_2) = 6 \qquad \mu(e_3) = 3 \qquad \mu(e_4) = 9$$

For example, we have that argument b can only be considered if the pieces of evidence e_1 and e_2 are available and these have a cost of 2 and 6, respectively. In Figure 1 we use dotted edges to indicate which piece of evidence is needed for which argument.

Let us consider grounded semantics (under credulous reasoning) and observe that in the complete framework where all pieces of evidence can be retrieved, we have that argument c is not accepted, i. e., $\Delta^{\text{cred}}_{\text{GR}}(F_{\mathcal{E}}, c) = n$. Let us now assume that $\hat{\mathcal{E}} = \{e_1, e_2\}$, i. e., it is not possible to retrieve e_3 and e_4; then we have $\Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, c) = y$. The question now is, which pieces of evidence must Ag attempt to retrieve in order to come to the same conclusion as $F_{\hat{\mathcal{E}}}$ (recall that $\hat{\mathcal{E}}$ is not known to the agent)? Let us consider some scenarios:

- Ag can attempt to retrieve the entire set of evidence $E_1 = \mathcal{E} = \{e_1, e_2, e_3, e_4\}$. He would then retrieve e_1 and e_2 and learn that e_3 and e_4 are unavailable. Thus, Ag knows that $F_{E_1 \cap \hat{\mathcal{E}}} = F_{\hat{\mathcal{E}}}$ and therefore trivially $\Delta^{\text{cred}}_{\text{GR}}(F_{E_1 \cap \hat{\mathcal{E}}}, c) = \Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, c)$. The cost for his attempts of retrieval then is $\mu(E_1) = 2 + 6 + 3 + 9 = 20$.
- Ag can retrieve $E_2 = \{e_1, e_2\}$ to obtain $F_{E_2 \cap \hat{\mathcal{E}}} = F_{E_1 \cap \hat{\mathcal{E}}}$ with cost only $\mu(E_2) = 2+6 = 8$. However, as Ag does not know whether any of the other pieces of evidence e_3 and e_4 are actually unavailable, he cannot be sure that $\Delta^{\text{cred}}_{\text{GR}}(F_{E_2 \cap \hat{\mathcal{E}}}, c) = \Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, c)$. For example, e_3 could be available, changing the acceptability status of c due to the presence of d.
- Ag can attempt to retrieve $E_3 = \{e_1, e_2, e_3\}$ to again obtain the same framework $F_{E_3 \cap \hat{\mathcal{E}}} = F_{E_1 \cap \hat{\mathcal{E}}}$ with cost $\mu(E_3) = 2 + 6 + 3 = 11$. As he learns that e_3 is not available, he can be sure that $\Delta^{\text{cred}}_{\text{GR}}(F_{E_3 \cap \hat{\mathcal{E}}}, c) = \Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, c)$, independently of whether e_4 is available. Obviously searching for E_3 is better for Ag than searching for E_1 due to the lower cost.
- Ag can further minimise the cost while still being sure that the answer remains the same. In fact, Ag can attempt to retrieve $E_4 = \{e_2, e_3\}$ with cost $\mu(E_4) = 6+3 = 9$. As only e_2 can be retrieved from E_4, the framework $F_{E_4 \cap \hat{\mathcal{E}}}$ consists only of the argument c and we have $\Delta^{\text{cred}}_{\text{GR}}(F_{E_4 \cap \hat{\mathcal{E}}}, c) = y$. However, if e_1 would be searched for and retrieved, the result stays the same as both the attacker b of c and its defender a could be constructed.

In conclusion, the best strategy for Ag is to attempt to retrieve $E_4 = \{e_2, e_3\}$ with cost $\mu(E_4) = 9$ in order to learn that c is credulously accepted under the grounded semantics given the available evidence. So we have $OPT^{\text{cred,GR}}_{F,\hat{\mathcal{E}}}(c) = 9$.

Example 3. The situation introduced in Example 1 can be modelled with the AAFE $F = (\mathcal{A}, \mathcal{R}, \mathcal{E}, \delta, \mu)$, where

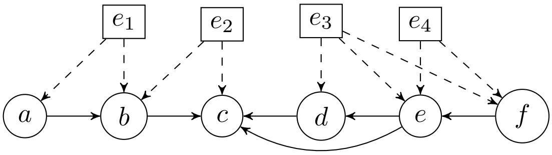


Figure 1. The argumentation framework with fallible evidence from Example 2.

$$\mathcal{A} = \{a, b, c, d\} \qquad \mathcal{R} = \{(b, a), (c, b), (d, a)\} \qquad \mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$$

$$\delta(a) = \{e_1\} \qquad \delta(b) = \{e_1, e_2, e_3\} \qquad \delta(c) = \{e_3, e_4\} \qquad \delta(d) = \{e_1, e_5\}$$

$$\mu(e_1) = 1 \qquad \mu(e_2) = 2 \qquad \mu(e_3) = 5 \qquad \mu(e_4) = 10 \qquad \mu(e_5) = 5$$

and the pieces of evidence are: e_1 (the patient has the set S_1 of signs and symptoms), e_2 (the patient has the set S_2 of signs and symptoms), e_3 (test T_1 for D' is positive), e_4 (test T_2 for D' is negative), e_5 (test T_3 for D is negative).

Here, we have $\Delta^{\text{cred}}_{\text{GR}}(F_{\mathcal{E}}, a) = n$. For instance, if we consider $\hat{\mathcal{E}} = \{e_1, e_2, e_3\}$, we have that $\Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, a) = n$. So, $E_{\mathsf{Ag}} = \{e_1, e_2, e_3, e_4\}$ (with cost $\mu(E_{\mathsf{Ag}}) = 1 + 2 + 5 + 10 = 18$) is the cheapest set of evidence such that $\Delta^{\text{cred}}_{\text{GR}}(F_{E_{\mathsf{Ag}} \cap \hat{\mathcal{E}}}, a) = \Delta^{\text{cred}}_{\text{GR}}(F_{\hat{\mathcal{E}}}, a)$ is guaranteed; moreover, it holds that $F_{E_{\mathsf{Ag}} \cap \hat{\mathcal{E}}} = F_{\hat{\mathcal{E}}}$. Note that there is no need to attempt to retrieve e_5 since, if d was active, the acceptance status of a would not change. Therefore, $OPT^{\text{cred},\text{GR}}_{F,\hat{\mathcal{E}}}(a) = 18$.

4. Computational Complexity

In the following we are interested in the computational complexity of determining $OPT^{\circ,\sigma}_{F,\hat{\mathcal{E}}}(a_{query})$ for an arbitrary AAFE F and an argument a_{query} with respect to a semantics σ and $\circ \in \{\text{cred}, \text{skep}\}$, given an arbitrary set of available evidence $\hat{\mathcal{E}}$. For that we consider the following decision problem variant:

σ-$\circ$-UAAFE	**Input:**	An AAFE F, an argument a_{query}, a set of available evidence $\hat{\mathcal{E}}$, $K \in \mathbb{N}$
	Output:	YES if $OPT^{\circ,\sigma}_{F,\hat{\mathcal{E}}}(a_{query}) \leq K$ and NO otherwise

We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P and NP as well as the polynomial hierarchy, see [11] for an introduction.

Table 1 gives an overview on our technical results. It can be seen that our analysis mirrors classical complexity results for abstract argumentation semantics [7], but most problems are lifted one level up in the polynomial hierarchy. The only exception to this is the problem AD-skep-UAAFE, which remains trivial[2]. We leave the case of preferred semantics for future work. Also, we omit the proofs due to space restrictions, but these can be found in an online appendix[3].

[2] This is because the empty set is always admissible and no argument can be skeptically accepted wrt. admissible semantics. Therefore $OPT^{\text{skep},\text{AD}}_{F,\hat{\mathcal{E}}}(a) = 0$ for every F, $\hat{\mathcal{E}}$, and a.

[3] http://mthimm.de/misc/aafe20_app.pdf

σ	σ-CRED-UAAFE	σ-SKEP-UAAFE
GR	NP-c	NP-c
AD	Π_2^P-c	trivial
CP	Π_2^P-c	NP-c
ST	Π_2^P-c	Σ_2^P-c

Table 1. Summary of complexity results where NP-c stands for "NP-complete".

5. Related works

The idea of using evidence to determine active arguments is not new in the computational argumentation community. In [3] one of the first approaches that uses evidence for this purpose was presented. There, the authors present a structured argumentation system based on DeLP [8] where arguments are pre-compiled (i. e. built beforehand) and then the available evidence is used to activate some of these pre-compilations. The work in [12] further generalises this idea of using evidence to activate pre-existing argument structures in the context of abstract argumentation. Like in the AAFEs, [12] characterises a set of evidence as a set of pieces of information, and establishes which pieces of evidence are required to activate an argument. However, differently from us, that work mostly focuses on providing a formal characterisation of the dynamics of the elements of the framework.

More closely related to our work is [4], where the problem being tackled there motivated our research. In [4] the authors use a simplified version of the framework presented in [12] but extended to consider (like in this paper) that the evidence associated with the arguments has to be retrieved, and such retrieval comes at a cost. Similarly to us, they aim to minimise the evidence retrieval cost incurred for determining the acceptance status of an argument. However, they do not provide a formal characterisation of the task as an optimisation problem nor study its complexity. Instead, they present an algorithm adopting a heuristic-based pruning technique for the construction of argumentation trees. Also, differently from us, they focus on a single semantics which is derived from the one adopted by DeLP and is also similar to the grounded semantics.

6. Summary and Conclusion

We introduced abstract argumentation frameworks with fallible evidence as a generalisation of Dung's abstract argumentation framework that models arguments being backed by pieces of evidence, which in turn may be available or not and have an associated cost when being attempted to be retrieved. As the main contribution, we formulated an optimisation problem characterising reasoning with AAFEs wrt. admissible, complete, grounded and stable semantics, studied its computational complexity, and showed that complexity rises one level in the polynomial hierarchy compared to the classical case.

Our findings can be particularly used to justify the choice of [4] to develop a heuristic algorithm for solving problems very similar to ours. As finding the optimal choice of which pieces of evidence to attempt to retrieve is intractable even for grounded semantics, there is little hope of finding a polynomial-time algorithm; thus, heuristic approaches akin to [4] could be developed for AAFEs with the aim of decreasing the evidence retrieval cost. Part of current work is to establish the computational complexity of our problem wrt. preferred semantics, which we conjecture to be on the third level of the polynomial hierarchy.

Acknowledgments

The work reported here has been partially supported by the Deutsche Forschungsgemeinschaft (grant KE 1413/11-1) and by Universidad Nacional del Sur (grant 24/N046).

References

[1] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, and S. Villata. Toward artificial argumentation. *AI Magazine*, 38(3):25–36, October 2017.

[2] P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors. *Handbook of Formal Argumentation.* College Publications, 2018.

[3] M. Capobianco, C. I. Chesñevar, and G. R. Simari. Argumentation and the dynamics of warranted beliefs in changing environments. *Autonomous Agents and Multi-Agent Systems*, 11(2):127–151, 2005.

[4] A. Cohen, S. Gottifredi, and A. J. García. A heuristic pruning technique for dialectical trees on argumentation-based query-answering systems. In *Proceedings of FQAS'19*, pages 101–113, 2019.

[5] A. Cohen, S. Gottifredi, A. J. García, and G. R. Simari. A survey of different approaches to support in argumentation systems. *Knowledge Engineering Review*, 29(5):513–550, 2014.

[6] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[7] W. Dvořák and P. E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation.* College Publications, 2018.

[8] A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[9] S. Gottifredi, A. Cohen, A. J. Garcia, and G. R. Simari. Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. *Artificial Intelligence*, 262:336–368, 2018.

[10] S. Nielsen and S. Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Proceedings of ArgMAS'06*, pages 54–73, Berlin, Heidelberg, 2007. Springer.

[11] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[12] N. D. Rotstein, M. O. Moguillansky, A. J. García, and G. R. Simari. A dynamic argumentation framework. In *Proceedings of COMMA'10*, pages 427–438, 2010.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200519

An Argument-Based Framework for Selecting Dialogue Move Types and Content

Mark SNAITH [1]
Centre for Argument Technology, University of Dundee, UK

Abstract. Choosing what to say is an integral part of multi-party dialogue, whether to ensure a natural flow, or advance a strategy. This paper presents an argument-based framework for selecting dialogue move types and content. The framework first builds and evaluates arguments in favour of move types and content being preferred, before determining whether or not there is an *optimal* outcome - where both the move type and content are preferred - or a *sub-optimal* outcome - where either the type or content is preferred, but not both.

Keywords. dialogue, dialogue move selection, strategies, argumentation

1. Introduction

During the course of a dialogue, it is important that participants say the right thing at the right time. Formalised dialogue games (such as those specified by [1,2,3]), subsequently implemented as dialogue protocols (e.g. [4,5,6,7]), assist with this by mandating what types of move can follow others. This does however only partially address the problem: first, many dialogue games allow multiple valid move types at any given point, raising the question of exactly which move type to choose. Second, protocols do not influence the specific content of a move (i.e. the actual utterance made by the participant). While some protocols do place constraints on the content, this is only in an abstract sense; for instance, the content of a move must support a certain proposition p, but several different pieces of content might fulfil this criterion.

Dialogue move selection is a widely-studied topic, especially in the context of strategy [8,9,10]. Participants in a dialogue consider their strategic objectives before selecting a move type and appropriate content that they believe will stand the best chance of achieving those objectives.

Choosing what to say in a dialogue is fundamentally a decision problem. Such problems are well-suited to being solved using argumentation [11], with arguments being constructed for and against each possible alternative. Using argumentation to select dialogue moves and content was previously studied by [12], where the selection of a move type and content is a two-stage process: first by selecting a single move type based on *strategic goals* (the meta-level goals of the participant, such as “minimising the dialogue time”), then finding and selecting content to populate it based on *functional goals* (subject-specific goals, such as what the participant wants to achieve). If content cannot

[1] E-mail: m.snaith@dundee.ac.uk.

be found for the selected move type, it is assumed that the strategic and functional goals are incompatible.

A drawback of this approach is that it does not consider any alternative move types should no content be found. While this is justified in terms of a participant remaining true to their strategic goals, it assumes that a participant is always prepared to be bound by those goals even if it means they cannot speak in the dialogue. Considering the choice of content alongside the choice of move, and using the availability or otherwise of content to influence the set of available moves increases the chance of finding *something* to say. A participant then has the choice between saying this (while violating or ignoring some or all of their strategic goals), or saying nothing (and staying true to their strategic goals). This drawback was also identified by [13], who instead propose an algorithm that considers move types and content independently and concurrently, before choosing the “best” combination of type and content on the basis of these determinations.

Common to both [12] and [13] is that the determination is based on two sets of goals - strategic (relating to move types) and functional (relating to content). While considering goals is important, especially from a strategic point of view, limiting the determination to *only* considering goals is somewhat restrictive because there may be additional external factors that influence whether or not a move type and/or content to fill it is available. As an example, a software agent designed to recommend exercise might base its advice on the weather (i.e. outdoor vs indoor exercise), but the weather itself does not influence the agent’s goals. Furthermore, an agent’s goals might not be explicit, but rather determined by a variety of factors based on what the agent values in general and not just related to this current specific dialogue.

This paper presents an argument-based framework that selects between possible move types and content. As well as move types and associated content, the framework takes as further inputs: a set of properties that are assigned to move types and content; a set of values that move types and content can promote[2]; and a preference ordering over those values, determined by the current dialogue participant. The output is a move type and content selection that is either **optimal**, where both the type and content are more preferred with respect to other types and content, or **sub-optimal**, where either the type or content is preferred, but not both.

2. Preliminaries

2.1. Inputs to the framework

2.1.1. Move types and content

For the purposes of this work, a specific dialogue framework is not used, nor a specific protocol. Instead, the only assumption made is that a dialogue is taking place, at a certain point in that dialogue a set of move types M is available to a participant, and that there exists a set of content C that can be used to populate those move types; a content function that collects valid content for each move type is defined thus:

Definition 1 *Content:* $M \rightarrow 2^C$; $Content(m) = \{c \in C \mid c \text{ is valid content for } m\}$

[2] These values while similar in principle to those found in value-based argumentation [14] are not used in the same way in the present work.

$C_{Valid} = \bigcup_{m \in M} Content(m)$ represents all valid content across all move types.

2.1.2. Properties

Properties allow applications of the framework to consider external conditions that must be true for a certain move type and/or piece of content to be available. To again use the example of an agent recommending exercise, content that references walking a dog should only be considered if the person to whom the advice is being given has a pet dog. A *property function* is defined to assign properties to move types and content:

Definition 2 *Let P be a set of properties and $X \in \{M, C\}$, where M is the set of move types and C is the set of content: $Prop_X$: $X \rightarrow 2^P$; $Prop_X(\alpha) = \{p \mid \alpha$ has property $p\}$*

$Prop_D \subseteq P$ represents the set of properties that are currently true for the dialogue D.

2.1.3. Values and preferences

Values, and an associated preference ordering over them, are used to determine move types and content that are preferred. Values themselves are assigned to move types and content, while the preference ordering is individual to the dialogue participant. As with properties, a function is defined to assign values to move types and content.

Definition 3 *Let V be a set of values and $X \in \{M, C\}$, where M is the set of move types and C is the set of content: Val_X: $X \rightarrow 2^V$; $Val_X(\alpha) = \{v \mid v$ is promoted by $\alpha\}$*

Dialogue participants assign a (possibly partial) preference ordering, $<$, over values, which in turn will allow for a preference ordering over move types and content to be determined. This determination is defined in Section 3.

2.2. Argumentation - ASPIC+

ASPIC+ [15] is used as the basis the argumentation model. ASPIC+ combines the work of [16] with that of [17] to provide an account of structured argumentation from which an abstract argumentation framework [18] can be obtained and evaluated for acceptability. The three core elements of ASPIC+ are an *argumentation system*, a *knowledge base*, and an *argumentation theory*.

Definition 4 *An argumentation system is a tuple $AS = \langle \mathcal{L}, cf, \mathcal{R}, \leq \rangle$ where: $\mathcal{L}$ is a logical language; cf: $\mathcal{L} \rightarrow 2^{\mathcal{L}}$, a contrariness function; $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules; and $\leq$ is a partial preorder on $\mathcal{R}_d$.*

In the present work, a Prolog-style language for $\mathcal{L}$ is used, whose formal definition is left implicit, but informally contains terms that consist of: **atoms**, that begin with a lowercase character (e.g. x, y, $some\_string$); **variables**, that begin with an uppercase character (e.g. X, Y, $SomeVariable$); and **compound terms**, consisting of an atom and a (parameterised) list of variables and/or atoms (e.g. $term(SomeVariable)$, $another\_term(SomeVariable, some\_string)$). For brevity in presentation rules are also expressed in a Prolog-style such that a single rule defined over variables is provided in place of multiple concrete rules defined over atoms; for example, if $\mathcal{K} =$

$\{foo(term1), foo(term2)\}$, then $\mathcal{R}_d = \{foo(X) => bar(Y)\}$ is shorthand for $\mathcal{R}_d = \{foo(term1) => bar(term1), foo(term2) => bar(term2)\}$. The same notation is also used in defining contraries and preferences.

Definition 5 *A knowledge base in an argumentation system* $AS = \langle \mathcal{L}, cf, \mathcal{R}, \leq \rangle$ *is a pair* $\langle \mathcal{K}, \leq' \rangle$, *where* $\mathcal{K} \subseteq \mathcal{L}$ *and* $\leq'$ *is a partial preorder on* $\mathcal{K} \setminus \mathcal{K}_n$.

Definition 6 *An argumentation theory is a triple* $AT = \langle AS, KB, \preccurlyeq \rangle$ *where AS is an argumentation system, KB is a knowledge base in AS and* $\preccurlyeq$ *is an argument ordering on the set of all arguments that can be constructed from KB in AS.*

Argument orderings in an argumentation theory are used determine preferences, and subsequently defeat. In the present work, the weakest link principle is used to determine this ordering because it takes into account preferences over premises in arguments.

Definition 7 *Let A and B be two arguments. Then* $A \prec B$ *iff either:* ***(1)*** *B is firm and strict and A is defeasible or plausable; or* ***(2)*** $LastDefRules(A) \preccurlyeq_S LastDefRules(B)$; *or* ***(3)*** $LastDefRules(A)$ *and* $LastDefRules(B)$ *are empty and* $Prem(A) \preccurlyeq_S Prem(B)$.

In Definition 7, *LastDefRules* and *Prem* are functions that return, respectively, the last defeasible rules and premises of the given argument, and $\preccurlyeq_S$ denotes a set ordering.

3. Building the framework

Underpinning the framework is an Argumentation Theory, $AT = \langle AS, KB, \preccurlyeq \rangle$. The resultant argumentation framework from AT is evaluated under some semantics that is left open to specific applications. A sceptical semantics, such as grounded, will lead to an outcome only if the move type and content preferences resolve all conflicts. A credulous semantics, such as preferred, will reveal all mutually-exclusive available outcomes.

3.1. Knowledge base and preferences

The knowledge base is constructed on the basis of the available move types, content and their respective properties. In the remainder of this paper, the following abbreviated terms are used: *thp* means “type has property” *chp* means “content has property”; *icf* means “is content for”:

- $\forall c \in C_{Valid}$, $content(c) \in \mathcal{K}$; and $\forall p \in Prop_C(c)$, $chp(c,p) \in \mathcal{K}$, and if $p \in Prop_D$, $property(p) \in \mathcal{K}$, else $\neg property(p) \in \mathcal{K}$
- $\forall m \in M$, $type(m) \in \mathcal{K}$; and $\forall c \in Content(m)$, $icf(c,m) \in \mathcal{K}$
- $\forall m \in M$, $\forall p \in Prop_M(m)$: $thp(m,p) \in \mathcal{K}$; and if $p \in Prop_D$, $property(p) \in \mathcal{K}$, else $\neg property(p) \in \mathcal{K}$

If for some X, $\{t,c\}hp(X,p) \in \mathcal{K}$ and $property(p) \notin \mathcal{K}$, then $\neg property(P) \in \mathcal{K}$. This imposes a partial closure property on $\mathcal{K}$ in terms of properties: if properties we know can exist (from the $\{t,c\}hp$ terms) are not explicitly true, they are assumed false.

Knowledge base preferences are determined from the preferences over the values they promote. As well as considering preferences over move types and content separately, preferences between move types and content are also permitted, i.e. a piece of content can be preferred to a certain move type. This allows the framework to resolve conflict between possible sub-optimal outcomes by considering whether or not a preferred move type is further preferred to a preferred piece of content, or vice versa.

Generating the knowledge base preferences requires a determination of preference over sets of values rather than individual values themselves. To do this, the *democratic determination* [19] is used:

$\forall c_1, c_2 \in C_{Available}$ s.t. $c_1 \neq c_2$: **(1)** if $Val_C(c_1) = \emptyset$ then $content(c_1) \not\leq content(c_2)$; else **(2)** if $Val_C(c_2) = \emptyset$ and $Val_C(c_1) \neq \emptyset$ then $content(c_1) \leq content(c_2)$; else **(3)** $content(c_1) \leq content(c_2)$ if $\forall X \in Val_C(c_1)$, $\exists Y \in Val_C(c_2)$ s.t. $X \leq Y$

$\forall m_1, m_2 \in M$ s.t. $m_1 \neq m_2$: **(1)** if $Val_M(m_1) = \emptyset$ then $type(m_1) \not\leq type(m_2)$; else **(2)** if $Val_M(m_2) = \emptyset$ and $Val_M(m_1) \neq \emptyset$ then $type(m_1) \leq type(m_2)$; else **(3)** $type(m_1) \leq type(m_2)$ if $\forall X \in Val_M(m_1)$, $\exists Y \in Val_M(m_2)$ s.t. $X \leq Y$

$\forall m \in M$, $\forall c \in C$: **(1)** if $Val_M(m) = \emptyset$ then $type(m) \not\leq content(c)$; else **(2)** if $Val_C(c) = \emptyset$ and $Val_M(m) \neq \emptyset$ then $type(m) \leq content(c)$; else **(3)** $type(m) \leq content(c)$ if $\forall X \in Val_M(m)$, $\exists Y \in Val_C(c)$ s.t. $X \leq Y$

3.2. Contrariness

Identifying the most preferred move types and content is a binary problem, insofar as if one move type (resp. piece of content) is most preferred, no others can be. To model this, we declare that *pt* ("preferred type") and *pc* ("preferred content") terms are contraries of themselves, except where specific instantiations would self-attack. In terms of outcomes, an optimal outcome (*opt*) should always attack a sub-optimal (*sub_opt*) outcome, while all sub-optimal outcomes should be in conflict with each other, again except where their arguments are assigned to the same atoms. Formally:

- $Cf(pt(Type1)) = \{pt(Type2)\}$, where $Type1 \neq Type2$;
- $Cf(pc(Content1)) = \{pc(Content2)\}$, where $Content1 \neq Content2$;
- $Cf(opt(\_,\_)) = \{sub\_opt(\_,\_)\}$, where _ represents any atom;
- $Cf(sub\_opt(Type1, Content1) = \{sub\_opt(Type2, Content2)\}$, where if $Type1 = Type2$, $Content1 \neq Content2$ and if $Content1 = Content2$, $Type1 \neq Type2$

3.3. Rules

One of the strengths of the framework is that move types and content are considered separately, before being brought together to determine outcomes. This achieved through the rules in AS, which are $\mathcal{R}_d = \{r_1, r_2, r_4, r_5, r_7, r_8, r_9\}$ and $\mathcal{R}_s = \{r_3, r_6\}$, where:

r_1 : $property(Property), thp(MoveType, Property) \Rightarrow at(MoveType)$
r_2 : $at(MoveType), type(Type) \Rightarrow pt(MoveType)$
r_3 : $\neg property(Property)\ thp(MoveType, Property) \rightarrow \neg at(MoveType)$
r_4 : $property(Property), chp(Content, Property) \Rightarrow ac(Content)$
r_5 : $ac(Content)\ content(Content) \Rightarrow pc(Content)$
r_6 : $\neg property(Property), chp(Content, Property) \rightarrow \neg ac(Content)$
r_7 : $pt(MoveType), pc(Content), icf(Content, MoveType) \Rightarrow opt(MoveType, Content)$

r_8 : $at(MoveType), pc(Content), icf(Content, MoveType) \Rightarrow sub\_opt(MoveType, Content)$
r_9 : $pt(MoveType), ac(Content), icf(Content, MoveType) \Rightarrow sub\_opt(MoveType, Content)$

Rules r_1 and r_3, and r_5 and r_6 determine whether or not move types (resp. content) are available based on their properties. Rules r_2 and r_5 create arguments for move types (resp. content) being preferred. Rules r_7 through r_9 determine outcomes, either optimal (r_7) or two types of sub-optimal: incorporating a preferred move type (r_8), or preferred content (r_9). Also, two preferences over rules are defined: $r_8 < r_7$ and $r_9 < r_7$, ensuring that when there is an optimal outcome, all sub-optimal outcomes are defeated.

4. Examples

Here, three concrete examples are presented that illustrate applications of the framework using different preference orderings over values. All examples use the following knowledge base $\mathcal{K}$, whose construction is left implicit, and chp, thp and icf have the same meaning as in Section 3:

$$\left\{\begin{array}{l} property(p),\ \neg property(q),\ content(\phi),\ chp(\phi,p), chp(\psi,p), \\ content(\theta),\ chp(\theta,q),\ type(assert),\ thp(assert,p),\ type(question), \\ thp(question,p),\ icf(\psi,assert),\ icf(\phi,question),\ icf(\theta,assert) \end{array}\right\}$$

Notice that $\neg property(q), chp(\theta,q) \in K$. This means that, as a result of the strict rule $r6$ (defined in section 3), any argument for $ac(\theta)$ is strictly defeated, as are any other arguments in which it is a sub-argument. Since all rules (and thus all arguments) for optimal and sub-optimal outcomes rely on content being available, θ is not considered in any of the examples, illustrating the impact properties have in the framework.

Values and properties are assigned to move types and content as follows:

$Val_M(assert) = \{v_1\}$
$Val_M(question) = \{v_2\}$
$Val_C(\phi) = \{v_3\}$
$Val_C(\psi) = \{v_4\}$
$Val_C(\theta) = \{v_5\}$
$Prop_M(assert) = \{p\}$
$Prop_M(question) = \{p\}$
$Prop_C(\phi) = \{p\}$
$Prop_C(\psi) = \{p\}$
$Prop_C(\theta) = \{q\}$

4.1. Example 1: optimal outcome

Assume that the value preferences are: $v_4 < v_3 < v_2 < v_1$. On the basis of these preferences, we obtain a preference ordering over $\mathcal{K}$: $content(\psi) < content(\phi) < type(question) < type(assert)$. From $\mathcal{K}$ and the preferences over $\mathcal{K}$, the resultant arguments and defeats yield an argumentation framework that is shown (for clarity only partially) on the left of Figure 1 after evaluation under grounded semantics, with a solid line indicating "acceptable" and a dashed line indicating "unacceptable". The argument labels correspond to the following conclusions:

$A12$: $pc(\phi)$
$A13$: $pt(question)$
$A16$: $pt(assert)$
$A18$: $sub\_opt(assert, \phi)$
$A19$: $opt(assert, \phi)$
$A21$: $pc(\psi)$
$A22$: $opt(question, \psi)$
$A23$: $sub\_opt(question, \psi)$
$A24$: $sub\_opt(question, \psi)$
$A25$: $sub\_opt(assert, \phi)$

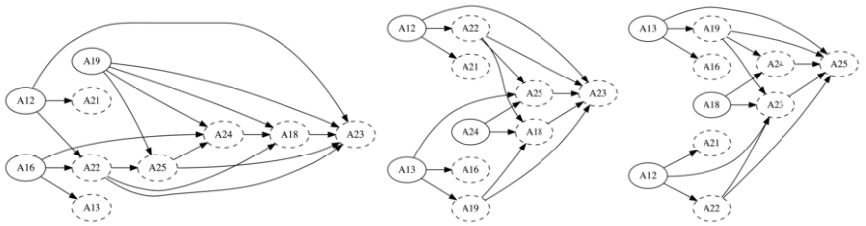


Figure 1. Partial argumentation frameworks for Examples 1 (L), 2 (C) and 3 (R)

Argument $A19$ is acceptable, representing an **optimal outcome**, of move type *assert* and content ϕ.

4.2. Example 2: sub-optimal outcome with preferred move type

Assume that the value preferences are changed to: $v_4 < v_3 < v_1 < v_2$. These lead to the knowledge base preferences: $content(\psi) < content(\phi) < type(assert) < type(question)$, which yield the argumentation framework partially shown in the centre of Figure 1, where the argument labels are the same as in Example 1. Argument $A24$ is acceptable, representing a **sub-optimal outcome** of move type *question* with content ψ. This is consistent with the preferences: *question* is more preferred to *assert*, but ψ is less preferred to ϕ, thus we could only have a sub-optimal outcome; the chosen sub-optimal outcome arises because *question* is more preferred to ϕ.

4.3. Example 3: sub-optimal outcome with preferred content

This final example uses value preferences: $v_1 < v_4 < v_2 < v_3$. These lead to the knowledge base preferences: $type(assert) < content(\psi) < type(question) < content(\phi)$, which yield the argumentation framework partially shown in the right of Figure 1, where again the argument labels are the same as in Example 1. Argument $A18$ is acceptable, representing a **sub-optimal outcome** of move type *assert* with content ϕ. This too is consistent with the preferences: ϕ is more preferred to ψ, but *assert* is less preferred to *question*, thus we could only have a sub-optimal outcome; the chosen sub-optimal outcome arises because ϕ is more preferred to *question*.

5. Summary and conclusions

This paper has presented an argument-based framework for selecting dialogue move types and content. The framework takes into account necessary properties, values promoted by move types and content, and a preference ordering over those values by the dialogue participant. By constructing arguments in favour of preferred move types and content, the framework can determine *optimal* and *sub-optimal* outcomes: an optimal outcome is where a preferred move type can be matched with preferred content; a sub-optimal outcome is where only the move type or the content is preferred.

Directions for future work include examining the properties of the framework to determine whether or not an outcome (whether optimal or sub-optimal) can always be reached. Additionally, the framework could be extended to take into account exactly how the set of possible content is arrived at for each move type. If for instance pieces of content are themselves the conclusions of arguments, those arguments may influence the values the content promotes, or the preference ordering over those values. Further extensions will also examine refinement and/or expansion of the argument model to either increase the possibility of yielding only a single outcome (optimal or sub-optimal), or providing an additional step to further choose between them.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553.

References

[1] Hamblin CL. Fallacies. Methuen & Company; 1970.

[2] Walton DN. Logical Dialogue-Games and Fallacies. Lanham: University Press of America; 1984.

[3] Walton DN, Krabbe ECW. Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, New York; 1995.

[4] McBurney P, Parsons S. Dialogue Games in Multi-Agent Systems. Informal Logic. 2002;22(3):257–274.

[5] McBurney P, Parsons S. Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. Journal of Logic, Language and Information. 2002;11:315–334.

[6] Prakken H. Formal systems for persuasion dialogue. The knowledge engineering review. 2006;21(2):163–188.

[7] Black E, Hunter A. An inquiry dialogue system. Autonomous Agents and Multi-Agent Systems. 2009;19:173–209.

[8] Kakas A, Maudet N, Moraitis P. Layered strategies and protocols for argumentation-based agent interaction. In: International Workshop on Argumentation in Multi-Agent Systems. Springer; 2004. p. 64–77.

[9] Oren N, Norman T, Preece A. Loose Lips Sink Ships: a Heuristic for Argumentation. In: Maudet N, Parsons S, Rahwan I, editors. Proceeings of the Third International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006); 2006. .

[10] Parsons S, McBurney P, Sklar E, Wooldridge M. On the relevance of utterances in formal inter-agent dialogues. In: International Workshop on Argumentation in Multi-Agent Systems. Springer; 2007. p. 47–62.

[11] Amgoud L, Prade H. Using arguments for making and explaining decisions. Artificial Intelligence. 2009;173(3-4):413–436.

[12] Amgoud L, Hameurlain N. An argumentation-based approach for dialog move selection. In: Maudet N, Parsons S, Rahwan I, editors. Proceeings of the Third International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006). Springer; 2006. p. 128–141.

[13] Beigi A, Mozayani N. A new dialogue strategy in multi-agent systems. Journal of Intelligent and Fuzzy Systems. 2014 01;27:641–653.

[14] Bench-Capon TJM. Value-based argumentation frameworks. In: Proceedings of Non Monotonic Reasoning; 2002. p. 444–453.

[15] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93–124.

[16] Pollock JL. Defeasible Reasoning. Cognitive Science. 1987;11:481–518.

[17] Vreeswijk GA. Abstract Argumentation Systems. Artificial Intelligence. 1997;90:225–279.

[18] Dung PM. On the acceptability of arguments and its fundemental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77:321–357.

[19] Caminada M, Modgil S, Oren N. Preferences and unrestricted rebut. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Proceddings of the Fifth International Conference on Computational Models of Argument (COMMA 2014). IOS Press; 2014. p. 209–220.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200520

On Computing the Set of Acceptable Arguments in Abstract Argumentation

Matthias THIMM [a,1], Federico CERUTTI [b] and Mauro VALLATI [c]
[a] *University of Koblenz-Landau, Germany*
[b] *University of Brescia, Italy*
[c] *University of Huddersfield, UK*

Abstract. We investigate the computational problem of determining the set of acceptable arguments in abstract argumentation wrt. credulous and skeptical reasoning under grounded, complete, stable, and preferred semantics. In particular, we investigate the computational complexity of that problem and its verification variant, and develop four SAT-based algorithms for the case of credulous reasoning under complete semantics, two baseline approaches based on iterative acceptability queries and extension enumeration and two optimised algorithms.

Keywords. abstract argumentation, computational complexity, algorithms

1. Introduction

In abstract argumentation [5], an argument a is skeptically (credulously) accepted wrt. some semantics σ, if it belongs to all (at least one) σ-extensions, respectively. Work on algorithms for solving reasoning problems in abstract argumentation—see e. g. the survey [2]—so far focused on deciding acceptability for a single query argument, or determining a single or all σ-extensions. However, the computational problem of directly computing the set of all acceptable arguments (wrt. either credulous or skeptical reasoning) has not been considered yet explicitly in the literature. Of course, this problem can be solved by reducing it to the aforementioned problems. For example, one can determine the set of all credulously accepted arguments by first computing all σ-extensions and then taking their union. In this paper, we ask the question whether this obvious approach is appropriate for the problem and whether other approaches provide superior performance.

Having efficient algorithms for computing the set of credulously or skeptically accepted arguments is of practical importance. Knowing whether specific arguments are not in any possible extensions—the dual problem of credulous acceptance—or knowing whether arguments are skeptically justified is of great service as also discussed in [11]. It allows human analysts to reduce their cognitive burden by consciously deciding whether or not to look more into a specific argument they made in their sense-making process.

[1] Corresponding Author: Matthias Thimm, University of Koblenz-Landau, Germany; E-mail: thimm@uni-koblenz.de.

In this paper, we first have a look at the theoretical complexity of the problem of verifying whether a given set of arguments is exactly the set of acceptable arguments wrt. both credulous and skeptical reasoning and the grounded, complete, stable, and preferred semantics. Our results mirror similar previous results [6] in that, for example, the verification problem for grounded semantics under both credulous and skeptical reasoning is in P, while the verification problem for skeptical reasoning for preferred semantics is DP2-complete (see Section 3 for definitions of the complexity classes). While the proofs of membership follow easily from existing results [6], the hardness proofs require some novel reduction techniques and insights. In addition to this theoretical analysis, we also present four SAT-based algorithms for addressing the question of determining the set of acceptable arguments wrt. credulous reasoning under complete semantics: two baseline approaches based on iterative acceptability queries and extension enumeration and two optimised algorithms. We provide an extensive experimental evaluation of the introduced algorithms considering all benchmarks from the past ICCMA competitions.

2. Preliminaries

An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ where A is a set of arguments and R is a relation $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$. For two arguments $a, b \in \mathsf{A}$ the relation $a\mathsf{R}b$ means that argument a attacks argument b. For $a \in \mathsf{A}$ define $a^- = \{b \mid b\mathsf{R}a\}$ and $a^+ = \{b \mid a\mathsf{R}b\}$. We say that a set $S \subseteq \mathsf{A}$ *defends* an argument $b \in \mathsf{A}$ if for all a with $a\mathsf{R}b$ then there is $c \in S$ with $c\mathsf{R}a$.

Semantics are given to abstract argumentation frameworks by means of extensions [5]. An extension E is a set of arguments $E \subseteq \mathsf{A}$ that is intended to represent a coherent point of view on the argumentation modelled by AF. Arguably, the most important property of a semantics is its admissibility. An extension E is called *admissible* if and only if (1) E is *conflict-free*, i. e., there are no arguments $a, b \in E$ with $a\mathsf{R}b$ and (2) E *defends* every $a \in E$, and it is called *complete* (CO) if, additionally, it satisfies (3) if E defends a then $a \in E$.

Different types of classical semantics can be phrased by imposing further constraints. In particular, a complete extension E: is *grounded* (GR) if and only if E is minimal; is *preferred* (PR) if and only if E is maximal; and is *stable* (ST) if and only if $\mathsf{A} = E \cup \{b \mid \exists a \in E : a\mathsf{R}b\}$.

All statements on minimality/maximality are meant to be with respect to set inclusion. Note that the grounded extension is uniquely determined and that stable extensions may not exist [5].

Let $\sigma \in \{\mathsf{CO}, \mathsf{GR}, \mathsf{ST}, \mathsf{PR}\}$ be some semantics and $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and abstract argumentation framework. Then, an argument $a \in \mathsf{A}$ is *skeptically accepted* in AF, denoted by $\mathsf{AF} \models^{s}_{\sigma} a$, if a is contained in *every* σ-extension. An argument $a \in \mathsf{A}$ is *credulously accepted* in AF, denoted by $\mathsf{AF} \models^{c}_{\sigma} a$, if a is contained in *some* σ-extension. Define $\mathsf{Acc}^{s}_{\sigma}(\mathsf{AF}) = \{a \in \mathsf{A} \mid \mathsf{AF} \models^{s}_{\sigma} a\}$ and $\mathsf{Acc}^{c}_{\sigma}(\mathsf{AF}) = \{a \in \mathsf{A} \mid \mathsf{AF} \models^{c}_{\sigma} a\}$ to be the sets of skeptically and credulously accepted arguments in AF, respectively. Observe that $\mathsf{Acc}^{s}_{\sigma}(\mathsf{AF}) \subseteq \mathsf{Acc}^{c}_{\sigma}(\mathsf{AF})$ for all semantics and abstract argumentation frameworks, except for $\sigma = \mathsf{ST}$ and an argumentation framework AF' that possesses no stable extension. In the latter case $\mathsf{Acc}^{s}_{\sigma}(\mathsf{AF}') = \mathsf{A}$ and $\mathsf{Acc}^{c}_{\sigma}(\mathsf{AF}') = \emptyset$ by definition.

In the remainder of the paper, we consider the computational problem of determining the sets $\mathsf{Acc}^{s}_{\sigma}(\mathsf{AF})$ and $\mathsf{Acc}^{c}_{\sigma}(\mathsf{AF})$, respectively. Note that, these exact problems have not been investigated before, to the best of our knowledge, in terms of computational complexity and algorithms. Previous studies and algorithms either focus on a single acceptability problem, such as deciding whether $\mathsf{AF} \models^{x}_{\sigma} a$ is true for $x \in \{s,c\}$ and some argument $a \in \mathsf{A}$, or computing one or all extensions (as done in the ICCMA series of argumentation competitions[2]).

3. Complexity of Computing the Set of Acceptable Arguments

We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P, NP and coNP, see [9] for an introduction. Recall that every decision problem can be represented as a language L that contains exactly those instances to the problem with answer "yes." A complexity class can then be represented by the languages of those problems it contains. We will make use of the complexity class DP, which is defined as $\mathsf{DP} = \{L_1 \cap L_2 \mid L_1 \in \mathsf{NP}, L_2 \in \mathsf{coNP}\}$. So DP contains those languages that are intersections of a language in NP and a language in coNP. We also need the following class $\mathsf{DP2} = \{L_1 \cap L_2 \mid L_1 \in \mathsf{NP}^{\mathsf{NP}}, L_2 \in \mathsf{coNP}^{\mathsf{NP}}\}$ where $\mathsf{NP}^{\mathsf{NP}}$ is the class of problems that can be solved by a non-deterministic Turing machine in polynomial time that has access to an NP oracle and $\mathsf{coNP}^{\mathsf{NP}}$ is the class of problems where the complement can be solved by a non-deterministic Turing machine in polynomial time that has access to an NP oracle. $\mathsf{NP}^{\mathsf{NP}}$ is also written as Σ^P_2 and $\mathsf{coNP}^{\mathsf{NP}}$ as Π^P_2. So $\mathsf{DP2}$ contains those languages that are intersections of a language in Σ^P_2 and a language in Π^P_2.

In this section we are interested in the computational complexity of the following decision problem:

$\mathsf{ACC}^{x}_{\sigma}$	**Input**:	$\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and $E \subseteq \mathsf{A}$
	Output:	TRUE iff $E = \mathsf{Acc}^{x}_{\sigma}(\mathsf{AF})$.

for a semantics σ and $x \in \{s,c\}$.

The proofs of the following results are omitted due to space restrictions but can be found in an online appendix[3]. We start with the tractable problems.

Proposition 1. *$\mathsf{ACC}^{s}_{\mathsf{GR}}$, $\mathsf{ACC}^{c}_{\mathsf{GR}}$, and $\mathsf{ACC}^{s}_{\mathsf{CO}}$ are in P.*

Many other problems are DP-complete.

Proposition 2. *$\mathsf{ACC}^{c}_{\mathsf{CO}}$, $\mathsf{ACC}^{c}_{\mathsf{PR}}$, and $\mathsf{ACC}^{c}_{\mathsf{ST}}$ are DP-complete.*

Proposition 3. *$\mathsf{ACC}^{s}_{\mathsf{ST}}$ is DP-complete.*

Skeptical inference with preferred semantics is (unsurprisingly) on the second level of the polynomial hierarchy.

Proposition 4. *$\mathsf{ACC}^{s}_{\mathsf{PR}}$ is DP2-complete.*

[2] `http://argumentationcompetition.org`

[3] `http://mthimm.de/misc/mtfcmv_accAF_proofs.pdf`

The results from above also allow us to easily provide an upper bound for the computational complexity of the functional problem of determining the set of acceptable arguments. For the following result, recall that $\mathsf{FNP}^{\mathsf{DP}[1]}$ is the complexity class of functional problems that can be solved by a non-deterministic Turing machine running in polynomial time that can call a DP-oracle for a constant number of times. The class $\mathsf{FNP}^{\mathsf{DP2}[1]}$ is defined analogously.

Corollary 5. *Let* AF *be an abstract argumentation framework.*

1. *The problems of computing $\mathit{ACC}^s_{\mathit{GR}}$, $\mathit{ACC}^c_{\mathit{GR}}$, $\mathit{ACC}^s_{\mathit{CO}}$ are in FP, respectively.*
2. *The problems of computing $\mathit{ACC}^c_{\mathit{CO}}$, $\mathit{ACC}^c_{\mathit{PR}}$, $\mathit{ACC}^c_{\mathit{ST}}$, $\mathit{ACC}^s_{\mathit{ST}}$ are in $\mathit{FNP}^{\mathit{DP}[1]}$, respectively.*
3. *The problem of computing $\mathit{ACC}^s_{\mathit{PR}}$ is in $\mathit{FNP}^{\mathit{DP2}[1]}$.*

4. SAT-based Algorithms for Credulous Reasoning

We will now investigate some algorithms that compute the set ACC^x_σ. Here, we will focus on the case of credulous reasoning under complete semantics (which is equivalent to credulous reasoning under preferred semantics).

We will develop reduction-based algorithms [4,2] and leverage SAT-solving technologies. Our encodings of acceptability problems into SAT are based on the encodings proposed initially in [1] and used in modern SAT-based argumentation solvers, see e. g. [4,3]. Let $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ be and abstract argumentation framework. For each argument $a \in \mathsf{A}$ we introduce three propositional variables $\mathtt{in}_a, \mathtt{out}_a, \mathtt{undec}_a$ which model the cases that a is in the extension, a is attacked by the extension, a is not in the extension nor attacked by it, respectively. Then define $\Phi_a = (\mathtt{out}_a \Leftrightarrow \bigvee_{b \in a^-} \mathtt{in}_b) \wedge (\mathtt{in}_a \Leftrightarrow \bigwedge_{b \in a^-} \mathtt{out}_b) \wedge (\mathtt{in}_a \vee \mathtt{out}_a \vee \mathtt{undec}_a)$ and $\Psi_{\mathsf{AF}} = \bigwedge_{a \in \mathsf{A}} \Phi_a$. For any propositional formula Φ, let $\mathsf{Mod}(\Phi)$ denote its set of models. For any model ω let $E(\omega) = \{a \mid \omega(\mathtt{in}_a) = \text{TRUE}\}$. Variants of the following observations have been proven in e. g. [1], so we state it without proof.

Proposition 6. *Let* $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ *be an abstract argumentation framework.*

1. *If $\omega \in \mathit{Mod}(\Psi_{\mathsf{AF}})$ then $E(\omega)$ is a complete extension of* AF.
2. *If E is a complete extension of* AF *then there is $\omega \in \mathit{Mod}(\Psi_{\mathsf{AF}})$ with $E(\omega) = E$.*
3. *$a \in \mathit{Acc}^c_{\mathit{CO}}(\mathsf{AF})$ if and only if $\Psi_{\mathsf{AF}} \wedge \mathtt{in}_a$ is satisfiable.*

The above observations enable us to use SAT solving technology by encoding abstract argumentation problems into one or a series of SAT problems.[4]

4.1. Iterative Acceptability Queries via SAT

A straightforward algorithm for determining $\mathsf{Acc}^c_{\mathsf{CO}}(\mathsf{AF})$ is to exploit observation 3.) of Proposition 6 and check for each $a \in \mathsf{A}$ whether $\Psi_{\mathsf{AF}} \wedge \mathtt{in}_a$ is satisfiable using some SAT solver. We denote this algorithm IAQ, it is depicted as Algorithm 1. We write $\text{SAT}(\phi)$ for a call to an external SAT solver that evaluates to TRUE if ϕ is satisfiable.

[4] Note that formulas such as Ψ_{AF} can be easily turned in conjunctive normal form, the standard input format for SAT solvers, with only polynomial overhead, so we do not explicitly discuss matters related to this aspect in the following.

Algorithm 1 Algorithm IAQ

```
Input:   $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$
Output:  $\mathsf{Acc}^c_{\mathsf{CO}}(\mathsf{AF})$
$S = \emptyset$
for $a \in \mathsf{A}$ do
    if $\textsc{Sat}(\Psi_{\mathsf{AF}} \wedge \mathtt{in}_a)$ then
        $S \leftarrow S \cup \{a\}$
return $S$
```

4.2. *Exhaustive extension enumeration via* SAT

Another straightforward approach is to leverage the fact that SAT solvers usually do not only report on the satisfiability of a given formula but also provide a model as witness. For a model ω let $C(\omega) = \bigvee_{\omega(\alpha)=\text{TRUE}} \neg\alpha \vee \bigvee_{\omega(\alpha)=\text{FALSE}} \alpha$. One can then enumerate all models of formula ϕ by first retrieving any one model ω, then retrieving a model ω' of $\phi \wedge C(\omega)$, then a model ω'' if $\phi \wedge C(\omega) \wedge C(\omega')$ and so on. It is clear that all models retrieved this way are models of ϕ and that by adding $C(\omega)$ we avoid retrieving the same model on future calls again. At some point, the formula becomes unsatisfiable and we retrieved all models. We can use this strategy to enumerate all complete extensions of an input abstract argumentation framework (using observations 2 and 3 of Proposition 6). The union of these is then the set $\mathsf{Acc}^c_{\mathsf{CO}}(\mathsf{AF})$. We denote this algorithm EEE, it is depicted as Algorithm 2. We write $\textsc{Witness}(\phi)$ for a call to an external SAT solver that evaluates to a model ω of ϕ if ϕ is satisfiable, or FALSE otherwise.

Algorithm 2 Algorithm EEE

```
Input:   $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$
Output:  $\mathsf{Acc}^c_{\mathsf{CO}}(\mathsf{AF})$
1: $S = \emptyset$
2: $\Psi \leftarrow \Psi_{\mathsf{AF}}$
3: while $\text{FALSE} \neq \omega = \textsc{Witness}(\Psi)$ do
4:     $S \leftarrow S \cup E(\omega)$
5:     $\Psi \leftarrow \Psi \wedge C(\omega)$
6: return $S$
```

4.3. *Selective extension enumeration via* SAT

We now turn to our proposal of the first non-trivial algorithm for computing $\mathsf{Acc}^c_{\mathsf{CO}}(\mathsf{AF})$. A major drawback of the algorithm EEE is that an abstract argumentation framework may feature an exponential number of complete extensions and many may overlap to a large degree. It may therefore be the case that in many iterations of the main loop in line 3 of Algorithm 2 no new arguments are added to S. In order to address this issue we propose a more selective extension enumeration SEE, implemented in Algorithm 3.

Differently from Algorithm 2, the algorithm SEE constrains the search for further models (line 3) by requiring that at least one argument that has not already been classified as accepted, needs to be included. Indeed, at the first iteration (line 3) the SAT solver will

Algorithm 3 Algorithm SEE

```
Input:   $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$
Output:  $\mathsf{Acc}^{c}_{\mathsf{CO}}(\mathsf{AF})$
1: $S = \emptyset$
2: $D \leftarrow \mathsf{A}$
3: while $\textsc{false} \neq \omega = \textsc{Witness}(\Psi_{\mathsf{AF}} \wedge \bigvee_{a \in D} \mathtt{in}_a)$ do
4:     $S \leftarrow S \cup E(\omega)$
5:     $D \leftarrow D \setminus E(\omega)$
6: return $S$
```

identify a complete extension with at least one in argument. The set of in arguments in the found extension will then be removed from the set D of *unvisited* arguments (line 5). From the second iteration, the SAT solver will then be forced to identify complete extensions that intersect with the unvisited arguments. It is straightforward to see that this algorithm is sound and complete.

4.4. *Selective extension enumeration via* MAXSAT

In (unweighted) MAXSAT [8] formulas can be either *hard* or *soft*. The solutions of a MAXSAT problem are determined among all assignments that satisfy all the hard formulas and are those that maximize the number of satisfied soft formulas. We write $\textsc{MaxSat}(S,H)$ (with a set of formulas S and a formula H) for a call to an external MAXSAT solver that evaluates to a model ω that satisfies H and a maximal number of formulas in S. If H is not satisfiable, $\textsc{MaxSat}(S,H)$ evaluates to FALSE. Algorithm 4 shows our final algorithm SEEM.

Algorithm 4 Algorithm SEEM

```
Input:   $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$
Output:  $\mathsf{Acc}^{c}_{\mathsf{CO}}(\mathsf{AF})$
1: $S = \emptyset$
2: $D \leftarrow \mathsf{A}$
3: while $\textsc{false} \neq \omega = \textsc{MaxSat}(\{\mathtt{in}_a \mid a \in D\}, \Psi_{\mathsf{AF}})$ do
4:     $S \leftarrow S \cup E(\omega)$
5:     $D \leftarrow D \setminus E(\omega)$
6: return $S$
```

The algorithm SEEM forces the MAXSAT solver to maximise the set of *unvisited* arguments at each iteration. Once again, it is straightforward to see how this algorithm is sound and complete.

5. Experimental Evaluation

We implemented the presented algorithms in the TWEETYPROJECT[5] and used the Open-WBO MAXSAT solver[6] for all calls of the form $\textsc{Sat}(\cdot)$, $\textsc{Witness}(\cdot)$, and $\textsc{MaxSat}(\cdot,\cdot)$.

[5] http://tweetyproject.org/r/?r=acc_reasoner

[6] http://sat.inesc-id.pt/open-wbo/

ICCMA'15					
No.	Algorithm	*N*	#TO	RT	PAR10
1	SEE	192	58	19947.35	3728.89
2	EEE	192	72	27061.65	4640.95
3	SEEM	192	79	21247.51	5048.16
4	IAQ	192	149	20285.96	9418.16
ICCMA'17					
No.	Algorithm	*N*	#TO	RT	PAR10
1	SEE	1050	558	60866.95	6435.11
2	SEEM	1050	579	55810.53	6670.29
3	IAQ	1050	742	54504.04	8531.91
4	EEE	1050	791	50607.98	9088.2
ICCMA'19					
No.	Algorithm	*N*	#TO	RT	PAR10
1	SEE	326	81	9775.1	3011.58
2	SEEM	326	82	14574.94	3063.11
3	EEE	326	109	11717.29	4048.21
4	IAQ	326	130	20257.52	4847.42

Table 1. Results of the ICCMA'15, ICCMA'17, and ICCMA'19 benchmark set; *N* is the total number of instances of the benchmark set; #TO gives the number of time-outs/errors of each solver on this benchmark set; RT gives the runtime in seconds on all correctly solved benchmarks; PAR10 gives the average runtime where time-outs count ten times the cutoff-time, i. e., 12,000 seconds.

We ran the experiments on a virtual machine running Ubuntu 18.04 with a 2.9 GHz CPU core and 8GB of RAM. We considered the following sets of benchmarks: ICCMA'15 consisting of 192 abstract argumentation frameworks [10]; ICCMA'17 consisting of 1050 abstract argumentation frameworks [7]; ICCMA'19 consisting of 326 abstract argumentation frameworks.[7]

Each algorithm was given 20 minutes to compute the set of acceptable arguments wrt. credulous reasoning with complete semantics. Algorithms are ranked by the number of unsolved instances (in increasing order). In case of ties, solvers are then ranked by runtime (in increasing order). We also considered the PAR10 (Penalised Average Runtime) score for comparing the performance of algorithms. PAR10 is a metric usually exploited in algorithm configuration techniques, where average runtime is calculated by considering runs that did not solve the problem as ten times the cutoff time. Intuitively,

Table 1 shows the performance of the considered algorithms on benchmarks from ICCMA'15, ICCMA'17, and ICCMA'19. The SEE algorithm is consistently delivering the best performance. Notably, on benchmarks from ICCMA'15 and ICCMA'19, the cumulative runtime of SEE is much lower than those of the other algorithms, despite the largest number of problems solved: SEE solved a larger number of problems in a much shorter amount of CPU-time. On the other end of the spectrum, the IAQ algorithm is generally delivering the worst performance. This comes as no surprise, considering that the IAQ algorithm is the less optimized among the implemented approaches.

The performance of EEE and SEEM algorithms, instead, are more remarkable. On the ICCMA'15 benchmarks the EEE algorithm is outperforming SEEM while on the ICCMA'17 benchmarks it is delivering the worst performance among the considered ap-

[7] The interested reader is referred to `https://www.iccma2019.dmi.unipg.it` for details.

proaches. In particular, the EEE algorithm shows a very limited coverage on instances generated on the basis of the Barabási–Albert model. Since frameworks derived according to the Barabási–Albert model usually have an extremely large number of complete extensions, the number of SAT calls made by the EEE algorithm is even larger than those made by the IAQ where one call per argument is made.

Finally, we observe that the selective extension enumeration implemented by the SEEM does not improve performance; instead it has a detrimental impact on performance, particularly when compared with the SEE approach. This may be due to the fact that the use of MAXSAT results in a more complex problem to be solved.

6. Summary and Conclusion

In this paper, we considered the computational task of computing the set of acceptable arguments in abstract argumentation wrt. credulous and skeptical reasoning and grounded, complete, stable, and preferred semantics. Our study on computational complexity showed that the corresponding decision variants are complete for the DP family of complexity classes, mirroring results for classical problems. We presented four different SAT-based algorithms for computing the set of credulously accepted arguments wrt. complete semantics and our evaluation showed that the two optimised algorithms significantly outperform the baseline algorithms. Future work will focus on extending the experimental evaluation to credulous reasoning with the other investigated semantics, and then investigating the case of skeptical reasoning.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant KE 1686/3-1).

References

[1] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In *Proceedinfs of NMR*, 2004.

[2] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. In *Handbook of Formal Argumentation*. 2018.

[3] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. How we designed winning algorithms for abstract argumentation and which insight we attained. *Artificial Intelligence*, 276:1 – 40, 2019.

[4] Günther Charwat, Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *Artificial Intelligence*, 220:28–63, 2015.

[5] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[6] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*. 2018.

[7] Sarah A. Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. *Artificial Intelligence*, 278, 2020.

[8] Chu Min Li and Felip Manya. Maxsat, hard and soft constraints. In *Handbook of Satisfiability*. 2009.

[9] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[10] Matthias Thimm and Serena Villata. The first international competition on computational models of argumentation: Results and analysis. *Artificial Intelligence*, 252:267–294, August 2017.

[11] Mauro Vallati, Federico Cerutti, and Massimiliano Giacomin. Predictive models and abstract argumentation: the case of high-complexity semantics. *The Knowledge Engineering Review*, 34:e6, 2019.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200521

Consistency in Assumption-Based Argumentation

Toshiko WAKAKI [1]
Shibaura Institute of Technology, Japan

Abstract. There exist counterexamples to Schulz and Toni's theorems which are the basis of their approach for justifying answer sets using assumption-based argumentation (ABA) whose language contains explicit negation. Against their claims, we present theorems showing the correspondence between answer sets of a consistent extended logic program and consistent stable extensions of the ABA instantiated with it. We show such ABA is not ensured to satisfy the consistency postulates. We also propose the novel notion of consistency for admissible dispute trees to avoid consistency problems in ABA applications containing explicit negation. We show the condition under which ABA consistency is ensured.

Keywords. consistency postulates, ABA consistency, consistent extensions, consistent sets of assumptions, consistent admissible dispute trees

1. Introduction

Consistency in assumption-based argumentation (ABA, for short) [1] is crucial to avoid anomalies in ABA applications whose languages contain explicit negation. In ASPIC$^+$ [9], for example, as the ways in which arguments can be in conflict, it allows the rebutting attack between two arguments having the mutually contradictory conclusions w.r.t. explicit negation along with undercutting and undermining attacks, while some conditions (e.g. ensuring closure under *transposition* or *contraposition*) under which ASPIC$^+$ satisfies *rationality postulates* [2] have been proposed to avoid anomalous results. In contrast, ABA allows only attacks against the support of arguments as defined in terms of a notion of *contrary*, while the *ab-self-contradiction axiom* [7] was proposed as the condition to guarantee the *consistency property* in an ABA framework containing explicit negation.

Recently as one of ABA applications containing explicit negation, extended abduction in ABA [14] was presented on the basis of the newly proposed definition of *consistency* in a flat ABA framework, which is slightly different from the notion of satisfying the *consistency property* [7] or the *direct consistency postulate* [2].

On the other hand, as another ABA application, Schulz and Toni proposed the approach of justifying answer sets using argumentation [11], where they used flat ABA frameworks instantiated with consistent extended logic programs (ELPs, for short) containing classical negation [8], i.e. explicit negation. However they took account of neither rationality postulates in such ABAs nor inconsistent extensions. As a result, it reveals the

[1] Corresponding Author: Toshiko Wakaki; E-mail: twakaki@shibaura-it.ac.jp

serious problem that there exist counterexamples to their theorems [11, Theorems 1, 2] such that answer sets of a *consistent* ELP are captured by stable extensions of the ABA framework instantiated with the ELP, though they are the basis of their approach for justifying why a literal belongs to an answer set of an ELP based on ABA. Besides regarding their computational machinery, we found that according to their lemma [11, Lemma 11], there may arise admissible abstract dispute trees whose defence sets are *inconsistent* though they are *admissible*. To the best of our knowledge, the notion of consistency for the defence set of an admissible dispute tree [5] has not been taken into account so far.

In this paper, first we discuss consistent extensions and ABA consistency. Second we show counterexamples to Schulz and Toni's theorems [11, Theorems 1, 2]. Then against their claims, we present the theorems showing that there is a one-to-one correspondence between answer sets of a *consistent* ELP and (*not* stable extensions but) *consistent* stable extensions of the ABA instantiated with the ELP. Besides we show that such ABA instantiated with a consistent ELP is not ensured to satisfy the consistency property, which implies that their theorems are incorrect. Third as another serious problem, we show the admissible abstract dispute tree whose defence set is *inconsistent* though it is *admissible* as derived due to their lemma [11, Lemma 11]. Then to detect and avoid such anomaly, we propose the novel notion of *consistency* for admissible dispute trees. So far a simplified assumption-based framework [5] (a simplified ABA, for short) having the restricted form w.r.t. explicit negation has often been used to illustrate admissible dispute trees without addressing consistency. Instead, thanks to our notion of consistency, we can show that the serious consistency problems addressed above never occurs in a simplified ABA since any defence set of its admissible dispute tree is consistent. Finally we present the condition to ensure ABA consistency in comparison with the *ab-self-contradiction axiom* to guarantee consistency-property in ABA.

The rest of this paper is as follows. Section 2 shows preliminaries. Section 3 discusses consistency in ABA. Section 4 shows counter examples to their theorems, presents the corrected theorems, proposes (in)consistent admissible dispute trees and shows the condition to ensure ABA consistency. Section 5 discusses related work and concludes.

2. Preliminaries

Definition 1 An ABA framework (or ABA) [6,1] is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, consisting of a language $\mathcal{L}$ (a set of sentences) and a set $\mathcal{R}$ of inference rules of the form: $b_0 \leftarrow b_1, \ldots, b_m$ ($b_i \in \mathcal{L}$ for $0 \leq i \leq m$), $\mathcal{A} \subseteq \mathcal{L}$ is a set of *assumptions*, and $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$. $\overline{\alpha}$ is referred to as the *contrary* of $\alpha \in \mathcal{A}$. For a rule $r \in \mathcal{R}$ of the form $b_0 \leftarrow b_1, \ldots, b_m$, let the head be $head(r) = b_0$ (resp. the body $body(r) = \{b_1, \ldots, b_m\}$).

We enforce that ABA frameworks are *flat*, namely assumptions do not occur in the head of rules. In ABA, *arguments* and $attacks$ are defined as follows [6]:

- an *argument for* $c \in \mathcal{L}$ (*the conclusion* or *claim*) *supported by* $K \subseteq \mathcal{A}$ ($K \vdash c$ in short) is a (finite) tree with nodes labelled by sentences in $\mathcal{L}$ or by $\tau \notin \mathcal{L}$ denoting "true", such that the root is labelled by c, leaves are labelled either by τ or by assumptions in K, and each non-leaf node N is labelled by $b_0 = head(r)$ for some rule $r \in \mathcal{R}$, where N has a child labelled by τ if $body(r) = \emptyset$; otherwise N has m children, each of which is labelled by $b_j \in body(r) = \{b_1, \ldots, b_m\} (1 \leq j \leq m)$.

- *an argument* $K_1 \vdash c_1$ *attacks an argument* $K_2 \vdash c_2$ iff $c_1 = \overline{\alpha}$ for $\alpha \in K_2$ (1)

Let $AF_{\mathcal{F}} = (AR, attacks)$ be the abstract argumentation framework generated from a flat ABA framework $\mathcal{F}$. For $Args \subseteq AR$, let $Args^+ = \{B \in AR \mid A \ attacks\ B$ for $A \in Args\}$. $Args$ is *conflict-free* iff $Args \cap Args^+ = \emptyset$. $Args$ *defends* an argument A iff each argument that attacks A is attacked by an argument in $Args$.

Definition 2 ***(ABA semantics)*** [1,4] Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ } \rangle$ be a flat ABA framework, and AR the associated set of arguments. Then $Args \subseteq AR$ is: admissible iff $Args$ is conflict-free and *defends* all its elements; a complete argument extension iff $Args$ is admissible and contains all arguments it *defends*; a preferred (resp. grounded) argument extension iff it is a (subset-)maximal (resp. (subset-)minimal) complete argument extension; a stable argument extension iff it is conflict-free and $Args \cup Args^+ = AR$; an ideal argument extension iff it is a (subset-)maximal admissible set contained in every preferred argument extension.

Hereafter let $\sigma \in \{$complete, preferred, grounded, stable, ideal$\}$. The various ABA semantics is originally given by sets of assumptions called assumption extensions. There is a one-to-one correspondence between σ assumption extensions and σ argument extensions such that for a σ assumption extension $Asms$, $\texttt{Asms2Args}(Asms) = \{K \vdash c \in AR \mid K \subseteq Asms\}$ is a σ argument extension, and for a σ argument extension $Args$, $\texttt{Args2Asms}(Args) = \{\alpha \in K \mid K \vdash c \in Args, K \subseteq \mathcal{A}\}$ is a σ assumption extension [3]. Let *claim*(Ag) be the claim (or conclusion) of an argument Ag. Then the *conclusion* of a set of arguments $\mathcal{E}$ is $\texttt{Concs}(\mathcal{E}) = \{c \in \mathcal{L} \mid c = claim(Ag)$ for $Ag \in \mathcal{E}\}$, while the *consequences* of a set of assumptions $A \subseteq \mathcal{A}$ is $Cn(A) = \{s \in \mathcal{L} \mid \exists A' \vdash s$ for $A' \subseteq A\}$.

Definition 3 ***(Dispute trees)*** [5] Given a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ } \rangle$, an abstract dispute tree for an initial argument a is defined as a (possibly infinite) tree $\mathcal{T}$ such that **1.** Every node of $\mathcal{T}$ is labelled by an argument and is assigned the status of *proponent* node or *opponent* node, but not both. **2.** The root is a proponent node labelled by a. **3.** For every proponent node N labelled by an argument b, and for every argument c that attacks b, there exists a child of N, which is an opponent node labelled by c. **4.** For every opponent node N labelled by an argument b, there exists exactly one child of N which is a proponent node labelled by an argument which attacks some assumption α in the set supporting b. α is said to be the *culprit* in b. **5.** There are no other nodes in $\mathcal{T}$ except those given by **1-4** above.

The set of all assumptions belonging to the proponent nodes in $\mathcal{T}$ is called the defence set of $\mathcal{T}$, denoted by $\mathcal{D}(\mathcal{T})$. An abstract dispute tree $\mathcal{T}$ is admissible if and only if no culprit in the argument of an opponent node belongs to $\mathcal{D}(\mathcal{T})$. If $\mathcal{T}$ is an admissible abstract dispute tree for an argument a, then $\mathcal{D}(\mathcal{T})$ is an admissible set of assumptions. If a is an argument supported by a set of assumptions $A_0 \subseteq E$ where E is admissible, then there exists an admissible dispute tree $\mathcal{T}$ for a with defence set $\mathcal{D}(\mathcal{T})$ and $A_0 \subseteq \mathcal{D}(\mathcal{T}) \subseteq E$ and $\mathcal{D}(\mathcal{T})$ is admissible [5, Theorem 5.1].

Satisfying Caminada and Amgoud's rationality postulates [2] or the *closure* and *consistency* properties in ABA [7] is stated as follows.

Definition 4 ***(Rationality postulates)*** [7,2] Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ } \rangle$ be a flat ABA framework. A set $X \subseteq \mathcal{L}$ is said to be contradictory iff X is contradictory w.r.t. $\overline{\ }$, i.e. there exists an assumption $\alpha \in \mathcal{A}$ such that $\{\alpha, \overline{\alpha}\} \subseteq X$; or X is contradictory w.r.t. $\neg$, i.e. there exists $s \in \mathcal{L}$ such that $\{s, \neg s\} \subseteq X$ if $\mathcal{L}$ contains an explicit negation operator $\neg$. Let $CN_{\mathcal{R}} : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ be a consequence operator. For a set $X \subseteq \mathcal{L}$, $CN_{\mathcal{R}}(X)$ is the smallest set such that $X \subseteq CN_{\mathcal{R}}(X)$, and for each rule $r \in \mathcal{R}$, if $body(r) \subseteq CN_{\mathcal{R}}(X)$ then $head(r) \in CN_{\mathcal{R}}(X)$. X is closed iff $X = CN_{\mathcal{R}}(X)$. A set $X \subseteq \mathcal{L}$ is said to be inconsistent iff its closure $CN_{\mathcal{R}}(X)$ is contradictory. X is said to be consistent iff it is not inconsistent. A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is said to satisfy the *consistency-property* (resp. the *closure-property*) if for each complete extension $\mathcal{E}$ of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$, $\texttt{Concs}(\mathcal{E})$ is consistent (resp. $\texttt{Concs}(\mathcal{E})$ is closed) [7].

Definition 5 An extended logic program (ELP, for short) [8] is a set of rules of the form:

$$L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n \quad (n \geq m \geq 0), \tag{2}$$

where each L_i is a literal, i.e. either an atom A or $\neg A$ preceded by classical negation $\neg$. not represents *negation as failure* (NAF). A literal preceded by not is called a NAF-literal. Let Lit_P be the set of all ground literals in the language of an ELP P. The semantics of an ELP is given by *answer sets* [8] (resp. *paraconsistent stable models* or *p-stable models*, for short [10]) defined as follows.

First, let P be a *not*-free ELP (i.e., for each rule $m = n$). Then, $S \subseteq Lit_P$ is an *answer set* of P if S is a minimal set satisfying the following two conditions **(i)**,**(ii)**: **(i)** For each ground instance of a rule $L_0 \leftarrow L_1, \ldots, L_m$ in P, if $\{L_1, \ldots, L_m\} \subseteq S$, then $L_0 \in S$. **(ii)** If S contains a pair of complementary literals L and $\neg L$, then $S = Lit_P$. Second, let P be any ELP and $S \subseteq Lit_P$. The *reduct* of P by S is a *not*-free ELP P^S which contains $L_0 \leftarrow L_1, \ldots, L_m$ iff there is a ground rule of the form (2) in P such that $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then S is an answer set of P if S is an answer set of P^S.

In contrast, p-stable models are regarded as answer sets defined without the condition **(ii)**. An answer set is *consistent* if it is not Lit_P; otherwise it is *inconsistent*. An ELP P is *consistent* if it has a consistent answer set; otherwise P is *inconsistent* under answer set semantics. On the other hand, a p-stable model is *inconsistent* if it contains a pair of complementary literals; otherwise it is *consistent*. For an ELP P, P is *consistent* if it has a consistent p-stable model; otherwise it is *inconsistent* under paraconsistent stable model semantics.

3. Consistency in Assumption-Based Argumentation

We discuss *ABA consistency* and *consistent* extensions in an ABA framework whose language contains explicit negation. The following theorem holds in ABA.

Theorem 1 [14] *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework and $\mathcal{E}$ be a complete argument extension of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$.*

1. *$\mathcal{F}$ satisfies* the closure-property.
2. *$\mathcal{F}$ satisfies* the consistency-property *iff for every $\mathcal{E}$,* $\texttt{Concs}(\mathcal{E})$ *is consistent iff for every $\mathcal{E}$,* $\texttt{Concs}(\mathcal{E})$ *is not contradictory w.r.t. explicit negation $\neg$.*

Item 2 in Theorem 1 denotes that an ABA $\mathcal{F}$ satisfies the consistency-property iff $AF_{\mathcal{F}}$ satisfies the *direct consistency* postulate [2] under complete semantics. In contrast, ABA consistency is differently defined as follows.

Definition 6 ***(Consistent argument extensions)***[14] Given a flat ABA framework $\mathcal{F}$, let $\mathcal{E}$ be a complete argument extension of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$. Then the extension $\mathcal{E}$ is said to be consistent if $\texttt{Concs}(\mathcal{E})$ is not contradictory w.r.t. $\neg$; otherwise it is inconsistent.

Definition 7 ***(ABA consistency)*** [14] A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is said to be *consistent* under σ semantics if $AF_{\mathcal{F}}$ generated from $\mathcal{F}$ has a consistent σ argument extension; otherwise it is inconsistent.

Note that if a flat ABA framework $\mathcal{F}$ satisfies the consistency-property, $\mathcal{F}$ is consistent under complete semantics, but not vice versa. ABA consistency can be also stated in terms of consistent assumption extensions based on the theorem shown below.

Proposition 1 ***(Consistent conflict-free sets of assumptions)*** *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework and $A \subseteq \mathcal{A}$ be a conflict-free set of assumptions. Then A is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.*

Proof. *(i) Since $A \subseteq \mathcal{A}$ is conflict-free, it holds that $\nexists \alpha \in A$ such that $A' \vdash \bar{\alpha}$ for $A' \subseteq A$. (ii) Since $\mathcal{F}$ is flat, it holds that $Cn(A) \cap \mathcal{A} = A$ (iii) It holds that for a set $A \subseteq \mathcal{A}$, $CN_{\mathcal{R}}(A) = Cn(A)$. Then due to (i),(ii),(iii), there exists no assumption $\alpha \in \mathcal{A}$ such that $\{\alpha, \overline{\alpha}\} \subseteq Cn(A) = CN_{\mathcal{R}}(A)$, which means that $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\overline{}$. Hence a conflict-free set $A \subseteq \mathcal{A}$ is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.* □

The following is derived based on Proposition 1.

Corollary 1 ***(Consistent admissible set of assumptions/ consistent assumption extensions)*** *Let A be any one of an admissible set of assumptions and an assumption extension. Then $A \subseteq \mathcal{A}$ is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.*

Theorem 2 ***(ABA consistency)*** *A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is* consistent *under σ semantics iff $\mathcal{F}$ has a consistent σ assumption extension; otherwise it is inconsistent.*

Proof. *($\Leftarrow$) Let $A \subseteq \mathcal{A}$ be a consistent σ assumption extension in $\mathcal{F}$. For A, there is a σ argument extension $\mathcal{E}$ in $\mathcal{F}$ such that $\mathcal{E} = \texttt{Asms2Args}(A)$. Then it holds that $CN_{\mathcal{R}}(A) = Cn(A) = \{c \in \mathcal{L} \mid \exists K \vdash c, K \subseteq A\} = \{c \in \mathcal{L} \mid \exists K \vdash c \in \mathcal{E}\} = \texttt{Concs}(\mathcal{E})$. Besides since the assumption extension $A \subseteq \mathcal{A} \subseteq \mathcal{L}$ is consistent, $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$ due to Corollary 1. Therefore $\mathcal{F}$ has the consistent σ argument extension $\mathcal{E}$ since $\texttt{Concs}(\mathcal{E}) = CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$. Hence $\mathcal{F}$ is consistent.*
($\Rightarrow$) The converse is also proved similarly. □

The following example illustrates the difference between satisfaction of the consistency-property and ABA consistency.

Example 1 The following ELP P_1 [2] expresses "*Married John*" [2] extended with the rule, $\neg wr \leftarrow not\ hw$:

$$P_1 = \{wr \leftarrow,\ \ go \leftarrow,\ \ m \leftarrow wr, not\ \neg m,\ \ b \leftarrow go, not\ \neg b,$$
$$hw \leftarrow m,\ \ \neg hw \leftarrow b,\ \ \neg b \leftarrow hw,\ \ \neg m \leftarrow \neg hw,\ \ \neg wr \leftarrow not\ hw\}.$$

P_1 has the unique consistent answer set $M_1 = \{wr, go, m, \neg b, hw\}$,
while it has two p-stable models, M_1 and $M_2 = \{wr, go, \neg m, b, \neg hw, \neg wr\}$,

[2] This ELP P_1 was inspired in a personal communication with Dr. Martin Caminada.

where M_1 is consistent but M_2 is inconsistent. Hence P_1 is consistent under answer set semantics as well as under paraconsistent stable model semantics.

In contrast, in the ABA framework $\mathcal{F}_{P_1} = \langle \mathcal{L}, \overline{P_1, \mathcal{A},}\overline{\ \ }\rangle$ instantiated with P_1 where $\mathcal{A} = \{not\ \neg m, not\ \neg b, not\ hw\}$, $\overline{not\ \neg m} = \neg m$, $\overline{not\ \neg b} = \neg b$, $\overline{not\ hw} = hw$, there are arguments and *attacks* as follows:

$A_1 : \{\} \vdash wr$, $A_2 : \{\} \vdash go$, $A_3 : \{not\ \neg m\} \vdash m$, $A_4 : \{not\ \neg b\} \vdash b$,
$A_5 : \{not\ \neg m\} \vdash hw$, $A_6 : \{not\ \neg b\} \vdash \neg hw$, $A_7 : \{not\ \neg m\} \vdash \neg b$,
$A_8 : \{not\ \neg b\} \vdash \neg m$, $A_9 : \{not\ hw\} \vdash \neg wr$, $A_{10} : \{not\ \neg m\} \vdash not\ \neg m$,
$A_{11} : \{not\ \neg b\} \vdash not\ \neg b$, $A_{12} : \{not\ hw\} \vdash not\ hw$,
$attacks = \{(A_7, A_4), (A_7, A_6), (A_7, A_8), (A_7, A_{11}), (A_8, A_3), (A_8, A_5), (A_8, A_7),$
$(A_8, A_{10}), (A_5, A_9), (A_5, A_{12})\}$.

Then it has three complete argument extensions $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ as follows:

$$\mathcal{E}_1 = \{A_1, A_2, A_3, A_5, A_7, A_{10}\},$$
$$\mathcal{E}_2 = \{A_1, A_2, A_4, A_6, A_8, A_9, A_{11}, A_{12}\},$$
$$\mathcal{E}_3 = \{A_1, A_2\},$$

where $\mathtt{Concs}(\mathcal{E}_1) = \{wr, go, m, hw, \neg\, b, not\ \neg m\}$,
$\mathtt{Concs}(\mathcal{E}_2) = \{wr, go, b, \neg\ hw, \neg\ m, \neg wr, not\ \neg b, not\ hw\}$, $\mathtt{Concs}(\mathcal{E}_3) = \{wr, go\}$.
Note that $\mathcal{E}_1, \mathcal{E}_2$ are stable extensions such that $\mathtt{Concs}(\mathcal{E}_i) \cap Lit_{P_1} = M_i$ $(i = 1, 2)$.

Regarding classical negation $\neg$ contained in P_1 as explicit negation in $\mathcal{F}_{P_1}$, both $\mathcal{E}_1$ and $\mathcal{E}_3$ are consistent, while $\mathcal{E}_2$ is inconsistent. Therefore the ABA $\mathcal{F}_{P_1}$ is consistent under stable (resp. complete) semantics since it has the consistent stable extension $\mathcal{E}_1$, while $\mathcal{F}_{P_1}$ does not satisfy the consistency-property (i.e. violates the direct consistency postulate) since it has the inconsistent $\mathcal{E}_2$.

4. Consistency Required in ABA Applications

4.1. Counterexamples to Schulz and Toni's Theorems

In [11], an argument $K \vdash c$ in a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ }\rangle$ is denoted by $(K, F) \vdash c$, where $F = \{\ell_N |\ \ell_N \leftarrow \in R\}$ is the set of heads of rules with an empty body singled out from the set $R \subseteq \mathcal{R}$ of inference rules used in the construction of the argument $K \vdash c$. Then in their approach, the relation *attacks* is defined by using arguments whose form is $(K, F) \vdash c$ instead of $K \vdash c$ in (1) as follows:

- an argument $(K_1, F_1) \vdash c_1$ *attacks* an argument $(K_2, F_2) \vdash c_2$ iff $c_1 = \overline{\alpha}$ for $\alpha \in K_2$.

Given an ELP P, they defined *the translated ABA framework* $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P$ $\mathcal{A}_P, \overline{\ \ }\rangle$, i.e. the ABA instantiated with P, where $NAF_P = \{not\ \ell | \ell \in Lit_P\}$, $\mathcal{L}_P = Lit_P \cup NAF_P$, $\mathcal{R}_P = P$, $\mathcal{A}_P = NAF_P$, and $\overline{not\ \ell} = \ell$ for every $not\ \ell \in \mathcal{A}_P$. For $S \subseteq Lit_P$, $\Delta_S = \{not\ \ell \in NAF_P | \ell \notin S\}$ is the set of NAF-literals. If S is an answer set of P, $S_{NAF} = S \cup \Delta_S$ is called an *answer set with NAF literals* of P. According to [11, Notation 2], $\vdash_{MP}$ denotes derivability using *modus ponens* on $\leftarrow$ as the only inference rule. When used on $P \cup \Delta_S$, $\vdash_{MP}$ treats NAF literals syntactically like facts. Then to justify why a literal belongs to an answer set of an ELP based on ABA, Schulz and Toni claimed their theorems and lemma [11, Theorems 1, 2 and Lemma 1] for a logic program P [3], i.e. a consistent ELP as follows.

[3] In [11, Section 2.1], it is described that "if not stated otherwise, we assume that logic programs are consistent".

(c1) **[11, Theorem 1].** Let P be a logic program and let $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, \overline{} \rangle$. Let X be a set of arguments in ABA_P and let $T = \{k \mid \exists (AP, FP) \vdash k \in X\}$ be the set of all conclusions of arguments in X. X is a stable extension of ABA_P if and only if T is an answer set with NAF literals of P.

(c2) **[11, Theorem 2].** Let P be a logic program and let $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, \overline{} \rangle$. Let $T \subseteq Lit_P$ be a set of classical literals and let $X = \{(AP, FP) \vdash k \mid AP \subseteq \Delta_T\}$ be the set of arguments in ABA_P whose assumptions are in Δ_T. T is an answer set of P if and only if X is a stable extension of ABA_P.

(c3) **[11, Lemma 1].** Let P be a consistent logic program and let $S \subseteq Lit_P$.

(i) S is an answer set of P if and only if $S = \{\ell \in Lit_P \mid P \cup \Delta_S \vdash_{MP} \ell\}$.

(ii) $S_{NAF} = S \cup \Delta_S$ is an answer set with NAF literals of P if and only if $S_{NAF} = \{k \mid P \cup \Delta_S \vdash_{MP} k\}$.

There exist counterexamples to their claims (c1), (c2) as follows.

Example 2 ***(Counterexamples to (c1), (c2))*** Consider the following ELP P_2 [14],

$$P_2 = \{\neg p \leftarrow not\ a, \quad a \leftarrow p, not\ b, \quad p \leftarrow, \quad b \leftarrow not\ a\},$$

where $Lit_{P_2} = \{a, b, p, \neg a, \neg b, \neg p\}$. P_2 has the unique consistent answer set $S_1 = \{a, p\}$, where $S_{1_{NAF}} = S_1 \cup \Delta_{S_1} = \{a, p,\ not\ b, not\ \neg a, not\ \neg b, not\ \neg p\}$ is the answer set with NAF literals of P_2, while P_2 has two p-stable models, $S_1 = \{a, p\}$ and $S_2 = \{\neg p, p, b\}$, where $\Delta_{S_2} = \{not\ a, not\ \neg a, not\ \neg b\}$. Thus S_1 is consistent but S_2 is inconsistent. Hence P_2 is consistent under answer set semantics as well as under paraconsistent stable model semantics.

In contrast, $ABA_{P_2} = \langle \mathcal{L}_{P_2}, P_2, \mathcal{A}_{P_2}, \overline{} \rangle$ is constructed from P_2, which has arguments and *attacks* as follows:

$A_1 : (\{not\ a\}, \emptyset) \vdash \neg p,\ \ A_2 : (\{not\ b\}, \{p\}) \vdash a,\ \ A_3 : (\emptyset, \{p\}) \vdash p,$
$A_4 : (\{not\ a\}, \emptyset) \vdash b,\ \ A_5 : (\{not\ a\}, \emptyset) \vdash not\ a,\ \ A_6 : (\{not\ b\}, \emptyset) \vdash not\ b$
$A_7 : (\{not\ p\}, \emptyset) \vdash not\ p,\quad A_8 : (\{not\ \neg a\}, \emptyset) \vdash not\ \neg a,$
$A_9 : (\{not\ \neg b\}, \emptyset) \vdash not\ \neg b,\ \ A_{10} : (\{not\ \neg p\}, \emptyset) \vdash not\ \neg p,$
$attacks = \{(A_1, A_{10}), (A_2, A_1), (A_2, A_4), (A_2, A_5), (A_3, A_7), (A_4, A_2), (A_4, A_6)\}.$

Then ABA_{P_2} has two stable extensions $\mathcal{E}_1, \mathcal{E}_2$ as follows.

$$\mathcal{E}_1 = \{A_2, A_3, A_6, A_8, A_9, A_{10}\}, \quad \mathcal{E}_2 = \{A_1, A_3, A_4, A_5, A_8, A_9\},$$

where $\texttt{Concs}(\mathcal{E}_1) = \{a, p, not\ b, not\ \neg a, not\ \neg b, not\ \neg p\} = S_{1_{NAF}}$,
$\texttt{Concs}(\mathcal{E}_2) = \{\neg p, p, b, not\ a, not\ \neg a, not\ \neg b\} = S_2 \cup \Delta_{S_2}$.

Hence $\mathcal{E}_1$ is consistent but $\mathcal{E}_2$ is not. Besides $\texttt{Concs}(\mathcal{E}_i) = S_i \cup \Delta_{S_i}$ (or $\texttt{Concs}(\mathcal{E}_i) \cap Lit_{P_2} = S_i$) holds for $\mathcal{E}_i$ and S_i $(i = 1, 2)$. In the following, it is shown that claims (c1), (c2) do not hold for this consistent ELP P_2.

(1) Suppose that (c1) holds. Let X be the stable extension $\mathcal{E}_2$ of ABA_{P_2}. Then due to (c1), $T = \{k | \exists (AP, FP) \vdash k \in \mathcal{E}_2\} = \texttt{Concs}(\mathcal{E}_2)$ should be the answer set with NAF literals of P_2. However $\texttt{Concs}(\mathcal{E}_2) = S_2 \cup \Delta_{S_2}$ is not the answer set with NAF literals of P_2 since S_2 is not the answer set of P_2. Contradiction. Thus (c1) does not hold for P_2. □

(2) Suppose that (c2) holds. For $T = \{\neg p, p, b\}$ and $\Delta_T = \{not\ a, not\ \neg a, not\ \neg b\}$, $X = \{(AP, FP) \vdash k \mid AP \subseteq \Delta_T\} = \{A_1, A_3, A_4, A_5, A_8, A_9\} = \mathcal{E}_2$ is obtained, where $\mathcal{E}_2$ is the stable extension of ABA_{P_2}. Then due to (c2), $T = S_2$ should be the answer set of

P_2. However S_2 is not the answer set. Contradiction. Hence (c2) does not hold for P_2. □

Remark: The ELP P_1 in Example 1 is also a counterexample to (c1), (c2) [11, Theorems 1,2]. (Details are omitted due to space limitations.)

The reason why their theorems [11, Theorems 1, 2] do not hold is that they proved them based on the claim (c3), to which there exists also a counterexample.

Example 3 ***(Counterexample to (c3) [11, Lemma 1])*** Consider the consistent ELP P_2 and $S_2 = \{\neg p, p, b\}$ in Example 2.

- Suppose that (c3) (i) holds. For P_2 and Δ_{S_2}, $\{\ell \in Lit_{P_2} \mid P_2 \cup \Delta_{S_2} \vdash_{MP} \ell\} = \{\neg p, p, b\} = S_2$ is derived. Then due to (c3) (i), S_2 should be the answer set of P_2. However S_2 is not the answer set of P_2 but the p-stable model. Contradiction. Thus (c3) (i) does not hold.
- Similarly we can easily show that (c3) (ii) does not hold. □

4.2. Correspondence between Consistent Answer Sets and Consistent Stable Extensions

Two theorems [13, Theorems 3, 4] for an ELP were presented as *Extended Logic Programming as Argumentation*, whereas Schulz and Toni claimed (c1), (c2) [11, Theorems 1, 2] for a *consistent* ELP, i.e. the subclass of an ELP.

In [13, Theorems 3, 4], the following notations are used. Given an ELP P,

$$\mathcal{F}(P) = \langle \mathcal{L}_P, P, Lit_{not}, \overline{} \rangle$$

is the ABA framework instantiated with P, where $Lit_{not} = \{not\ L \mid L \in Lit_P\}$, $\mathcal{L}_P = Lit_P \cup Lit_{not}$ and $\overline{not\ L} = L$ for $not\ L \in Lit_{not}$. $AF_{\mathcal{F}}(P)$ denotes the abstract argumentation framework generated from the ABA $\mathcal{F}(P)$. For an ELP P, let

$$P_{tr} \stackrel{\text{def}}{=} P \cup \{L \leftarrow p, \neg p \mid p \in Lit_P,\ L \in Lit_P\},$$

be the ELP obtained from P by incorporating the *trivialization rules* [10]. Then $\mathcal{F}(P_{tr}) = \langle \mathcal{L}_P, P_{tr}, Lit_{not}, \overline{} \rangle$ is the ABA framework instantiated with P_{tr}, where $Lit_{P_{tr}} = Lit_P$ and $\mathcal{L}_{P_{tr}} = \mathcal{L}_P$. Besides for $M \subseteq Lit_P$, $\neg.CM = \{notL \mid L \in Lit_P \setminus M\}$ is the set of NAF literals.

Hence for an ELP P, $\mathcal{F}(P)$ (resp. Lit_{not}) corresponds to ABA_P (resp. $\mathcal{A}_P$) in [11], while for $M \subseteq Lit_P$, $M \cup \neg.CM$ (resp. $\neg.CM$) coincides with $M \cup \Delta_M$ (resp. Δ_M). Thus for an answer set M, $M \cup \neg.CM$ denotes $M_{NAF} = M \cup \Delta_M$ called *the answer set with NAF literals*. Theorems for an ELP are shown as follows.

Theorem 3 ***[13, Theorem 3].*** *Let P be an ELP. Then M is a p-stable model of P iff there is a stable extension $\mathcal{E}$ of $AF_{\mathcal{F}}(P)$ such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E})$ (in other words, $M = \mathtt{Concs}(\mathcal{E}) \cap Lit_P$).*

Theorem 4 ***[13, Theorem 4].*** *Let P be an ELP. Then M is an answer set of P iff there is a stable extension $\mathcal{E}_{tr}$ of the ABA $\mathcal{F}(P_{tr})$ (or $AF_{\mathcal{F}}(P_{tr})$ such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E}_{tr})$ (in other words, $M = \mathtt{Concs}(\mathcal{E}_{tr}) \cap Lit_P$).*

Example 2 illustrates that Theorem 3 holds for the p-stable model S_i of P_2 since $\mathtt{Concs}(\mathcal{E}_i) = S_i \cup \Delta_{S_i} = S_i \cup \neg.CS_i$ holds w.r.t. the stable extension $\mathcal{E}_i$ ($i = 1, 2$), while the following illustrates that Theorem 4 holds for answer sets of P_2.

Example 4 ***(Cont. Ex. 2)*** For P_2, P_{tr} is obtained as follows.
$P_{tr} = P_2 \cup \{L \leftarrow a, \neg a \,|\, L \in Lit_{P_2}\} \cup \{L \leftarrow b, \neg b \,|\, L \in Lit_{P_2}\} \cup \{L \leftarrow p, \neg p \,|\, L \in Lit_{P_2}\}$. Then the ABA $\mathcal{F}(P_{tr})$ (i.e. $ABA_{P_{tr}}$) has the unique stable extension $\mathcal{E}_{tr} = \mathcal{E}_1$ such that $\texttt{Concs}(\mathcal{E}_{tr}) = S_1 \cup \neg.CS_1 = S_1 \cup \Delta_{S_1} = S_{1_{NAF}}$ for the answer set with NAF literals $S_{1_{NAF}}$, where S_1 is the unique consistent answer set of P_2.

In what follows, we prove and present the correct theorems against their claims. First of all, we provide the following lemmas regarding a consistent ELP.

Lemma 1 *Let P be an ELP. M is a* consistent *answer set of P iff there is a* consistent *p-stable model M of P.*

Proof. *($\Leftarrow$) Let M be a consistent p-stable model of P. Then M does not contain a pair of complementary literals. Since M is also a p-stable model of the* reduct P^M *according to Def. 5, M is a minimal set satisfying the condition (i) for P^M which is the not-free ELP. Then since M does not contain a pair of complementary literals, M is also a minimal set satisfying both conditions (i) and (ii) for P^M. This denotes that M is the answer set of P^M which does not contain a pair of complementary literals. Thus M is the answer set of P^M and it is not Lit_P. Hence since the answer set M of P^M which is not Lit_P is the answer set of P, M is the consistent answer set of P.*
($\Rightarrow$) The converse is proved in a similar way. □

The following corollary is the direct result of Lemma 1.

Corollary 2 *An ELP P is* consistent *under answer set semantics iff P is* consistent *under paraconsistent stable model semantics.*

Lemma 2 *Let P be a* consistent *ELP. If M is an answer set of P, M is a p-stable model of P, but not vice versa.*

Proof. *($\Rightarrow$) Since P is consistent, its answer set M is consistent. Thus due to Lemma 1, M is a p-stable model of P.*
($\Leftarrow$) A consistent ELP P has a consistent p-stable model which is the answer set of P. Moreover it may have an inconsistent p-stable model M containing a pair of complementary literals L and $\neg L$. Then suppose that such inconsistent p-stable model M is also the answer set of P. Since M is the answer set of P, M is a minimal set satisfying the condition (i),(ii) in Def. 5 for the reduct P^M. Thus M is Lit_P due to (ii) since M contains a pair of complementary literals. However P has a consistent answer set $S \subset Lit_P$ because P is consistent. Thus M which is Lit_P is not minimal. Hence M is not the answer set of P. Contradiction. □

Hereby given a consistent ELP, we can obtain the following theorems.

Theorem 5 *Let P be a* consistent *ELP. Then M is an answer set of P iff there is a* consistent *stable extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ (or $AF_{\mathcal{F}}(P)$) such that $M \cup \neg.CM = \texttt{Concs}(\mathcal{E})$.*

Proof. *($\Leftarrow$) Let $\mathcal{E}$ be a consistent stable extension of the ABA $\mathcal{F}(P)$. Then $\texttt{Concs}(\mathcal{E})$ is consistent, i.e. not contradictory w.r.t. $\neg$. Due to Theorem 3, for the stable extension $\mathcal{E}$, there is the p-stable model M of P such that $M \cup \neg.CM = \texttt{Concs}(\mathcal{E})$. Since $\texttt{Concs}(\mathcal{E})$ does not contain a pair of complementary literals, $M \cup \neg.CM$ as well as the p-stable model M are consistent. Hence due to Lemma 1, M is the consistent answer set of P.*
($\Rightarrow$) The converse is also proved in a similar way. □

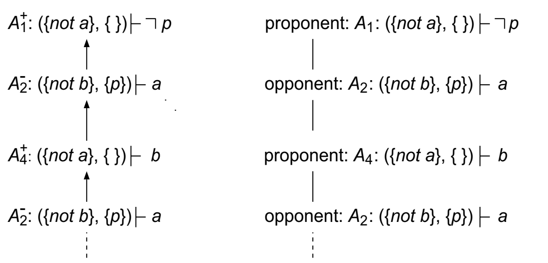

A1+: ({not a}, { })⊢¬p
A2-: ({not b}, {p})⊢ a
A4+: ({not a}, { })⊢ b
A2-: ({not b}, {p})⊢ a
proponent: A1: ({not a}, { })⊢¬p
opponent: A2: ({not b}, {p}) ⊢ a
proponent: A4: ({not a}, { }) ⊢ b
opponent: A2: ({not b}, {p}) ⊢ a

Figure 1. The admissible dispute tree $\mathcal{T}_{\mathcal{E}_2}(A_1)$ (right) vs. the positive attack tree $attTree^{+}_{\mathcal{E}_2}(A_1)$ (left) in Ex. 5

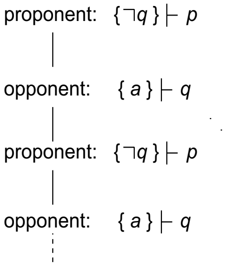


Figure 2. The admissible dispute tree $\mathcal{T}$ for $\{\neg q\} \vdash p$ in Ex. 6

Theorem 6 *Let P be a* consistent *ELP. If M is an answer set of P, there is a stable extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ such that $M \cup \neg.CM = \texttt{Concs}(\mathcal{E})$, but not vice versa.*

Proof. *This is proved based on Lemma 2 and Theorem 3.* □

Corollary 3 *Let P be a* consistent *ELP. $\mathcal{E}$ is a* consistent *stable extension of the ABA framework $\mathcal{F}(P)$ iff $\mathcal{E}$ is a stable extension of the ABA framework $\mathcal{F}(P_{pr})$.*

Theorem 5 and Theorem 6 state that there is a one-to-one correspondence between answer sets of a *consistent* ELP P and (*not* stable extensions but) *consistent* stable extensions of the ABA $\mathcal{F}(P)$ contrary to their claims (c1), (c2).

As for rationality postulates, the following theorem generally holds as illustrated in Example 1, which implies that Schulz and Toni's theorems are incorrect.

Theorem 7 *Let P be a* consistent *ELP. Then the ABA framework $\mathcal{F}(P)$ instantiated with P is consistent under complete (resp. stable) semantics, while it is not guaranteed to satisfy the consistent property or the direct consistency postulate.*

Proof. *There is an answer set of P. Then there is a consistent stable extension of $\mathcal{F}(P)$ based on Theorem 5. Hence $\mathcal{F}(P)$ is consistent under those semantics. Similarly there may be an inconsistent p-stable model of P. Then $\mathcal{F}(P)$ may have an inconsistent stable extension based on Theorem 3. Thus the latter is proved.* □

4.3. Consistency for Admissible Dispute Trees

Admissibility is defined for abstract (resp. concrete) dispute trees [5]. However *consistency* has not been taken into account for admissible dispute trees so far even though the following serious consistency problem may arise in ABA containing explicit negation.

Example 5 ***(Cont. Ex. 2)*** Consider ABA_{P_2} where classical negation $\neg$ in P_2 is regarded as explicit negation. In Figure 1, the left is the *positive Attack tree* $attTree^{+}_{\mathcal{E}_2}(A_1)$ of the argument $A_1 : (\{not\ a\}, \emptyset) \vdash \neg p$ with respect to the stable extension $\mathcal{E}_2 = \{A_1, A_3, A_4, A_5, A_8, A_9\}$ of ABA_{P_2}, while the right is the admissible abstract dispute tree $\mathcal{T}_{\mathcal{E}_2}(A_1)$ for A_1 translated from $attTree^{+}_{\mathcal{E}_2}(A_1)$ according to [11, Lemma 11]. Though there exists a fact p, i.e. $p \leftarrow\ \in P_2$ in ABA_{P_2}, the belief $\neg p$ is concluded to be admissible since the root of $\mathcal{T}_{\mathcal{E}_2}(A_1)$ is labelled with A_1 whose claim is $\neg p$, that implies contradiction. In fact, its defence set $\mathcal{D}(\mathcal{T}_{\mathcal{E}_2}(A_1)) = \{not\ a\}$ is inconsistent since $CN_{P_2}(\{not\ a\}) = \{\neg p, p, b, not\ a\}$ is contradictory w.r.t. $\neg$.

To detect and avoid such anomaly in ABA whose language contains explicit negation, we introduce the notion of *consistency* into admissible dispute trees.

Definition 8 ***(Consistent admissible dispute trees)*** Given a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, an admissible abstract (resp. concrete) dispute tree $\mathcal{T}$ is *consistent* if its defence set $\mathcal{D}(\mathcal{T})$ is *consistent*; otherwise it is *inconsistent*.

Proposition 2 ***(Consistent defence sets)*** *The defence set $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{A}$ of an admissible dispute tree $\mathcal{T}$ is consistent iff $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is not contradictory w.r.t. $\neg$.*

Proof. *This is proved due to Corollary 1 since $\mathcal{D}(\mathcal{T})$ is admissible.* □

A simplified assumption-based framework [5] is often used to illustrate an admissible dispute tree without stating consistency. The following ensures its consistency.

Proposition 3 *A simplified assumption-based framework (a simplified ABA, for short) [5] is an ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $\mathcal{F}$ is flat, all sentences in $\mathcal{L}$ are atoms $p, q, \ldots$ or negations of atoms $\neg p, \neg q, \ldots$ and $\overline{p} = \neg p$ for $p \in \mathcal{A}$ (resp. $\overline{\neg p} = p$ for $\neg p \in \mathcal{A}$). Then any admissible abstract (resp. concrete) dispute tree $\mathcal{T}$ in $\mathcal{F}$ is consistent.*

Proof. *Let $\alpha = p$ for $p \in \mathcal{A}$ (resp. $\alpha = \neg p$ for $\neg p \in \mathcal{A}$). Then $\{\alpha, \overline{\alpha}\} = \{p, \neg p\}$ for $p \in \mathcal{A}$ (resp. $\neg p \in \mathcal{A}$) is derived, which means that contradictoriness w.r.t. $\neg$ in $\mathcal{F}$ becomes contradictoriness w.r.t. $\overline{}$ in $\mathcal{F}$. Now let $\mathcal{T}$ be an admissible dispute tree in $\mathcal{F}$. Since $\mathcal{D}(\mathcal{T})$ is admissible, it is conflict-free. Besides $\mathcal{F}$ is flat. Then due to the proof of Proposition 1, $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is not contradictory w.r.t. $\overline{}$ in $\mathcal{F}$. Hence $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is also not contradictory w.r.t. $\neg$ in $\mathcal{F}$. Therefore any admissible dispute tree $\mathcal{T}$ in $\mathcal{F}$ is consistent since any $\mathcal{D}(\mathcal{T})$ is consistent based on Proposition 2.* □

Proposition 3 denotes that the consistency problem shown in Example 5 never arises in a simplified ABA. However the ABA $\mathcal{F}(P)$ (i.e. ABA$_P$) instantiated with an ELP P is not a simplified ABA.

Example 6 Consider the ABA $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $\mathcal{R} = \{p \leftarrow \neg q,\ q \leftarrow a, \neg p \leftarrow\}$, $\mathcal{A} = \{\neg q, a\}$, $\overline{\neg q} = q$ and $\overline{a} = p$. $\mathcal{F}$ is not a simplified ABA. It has three complete extensions $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ such that $\mathtt{Concs}(\mathcal{E}_1) = \{q, \neg p, a\}$, $\mathtt{Concs}(\mathcal{E}_2) = \{p, \neg p, \neg q\}$, $\mathtt{Concs}(\mathcal{E}_3) = \{\neg p\}$, where $\mathcal{E}_1$, $\mathcal{E}_2$ are stable extensions. $\mathcal{E}_1$, $\mathcal{E}_3$ are consistent but $\mathcal{E}_2$ is not. Then $\mathcal{F}$ is consistent under stable (resp. complete) semantics, while it does not satisfy the consistency property. Figure 2 shows the admissible abstract dispute tree $\mathcal{T}$ for the argument $\{\neg q\} \vdash p$. Its defence set $\mathcal{D}(\mathcal{T}) = \{\neg q\}$ is inconsistent since $CN_{\mathcal{R}}(\{\neg q\}) = \{p, \neg p, \neg q\}$ is contradictory w.r.t. $\neg$. Hence $\mathcal{T}$ is inconsistent though it is admissible. In contrast, the admissible abstract dispute tree $\mathcal{T}'$ for $\{a\} \vdash q$ is consistent since $\mathcal{D}(\mathcal{T}') = \{a\}$ is consistent due to $CN_{\mathcal{R}}(\{a\}) = \{q, \neg p, a\}$.

4.4. The Necessary and Sufficient Condition to Guarantee ABA Consistency

We show the condition to guarantee ABA consistency. Given an ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ whose set of arguments is finite, let Π be the ELP translated from $\mathcal{F}$ with no hypotheses (i.e. $\mathcal{H} = \emptyset$) defined in [14, Definition 13].

Then based on [14, Lemma 1], the *requirement* that the ELP Π (resp. $\Pi \cup \{\leftarrow undec(X)\}$ where $\mathcal{H} = \emptyset$ [14] should be consistent under answer set semantics is the *necessary* and *sufficient* condition that guarantees ABA consistency such that the ABA framework $\mathcal{F}$ is consistent under complete (resp. stable) semantics.

5. Related Work and Conclusion

Dung and Thang presented the *sufficient* condition referred to as the *ab-self-contradiction axiom* that guarantees closure- and consistency-properties in a flat ABA framework [7].

On the other hand, in [12], though it is shown that not the standard ABA but the generalized ABA mapped from a defeasible framework under some assumptions satisfies the closure and consistency postulates, Toni presented no results about satisfaction of those postulates in a standard flat ABA framework.

In this paper, we showed counterexamples to Schulz and Toni's theorems [11, Theorems 1, 2]. Then against their claims, we presented Theorems 5 and 6 showing that answer sets of a consistent ELP are captured by *not* stable extensions but *consistent* stable extensions of the ABA instantiated with the ELP. Theorem 7 shows such ABA instantiated with a consistent ELP is not ensured to satisfy the consistency postulate, that implies incorrectness of their theorems. We proposed the novel notion of consistency for admissible dispute trees to avoid anomalies in ABAs containing explicit negation. Finally we showed the condition to ensure ABA consistency. Our future work is to implement the method to compute consistent reasoning over ABA in answer set programming [8] (e.g. by using the ELP Π with $\mathcal{H} = \emptyset$ based on [14, Lemma 1]).

Acknowledgments This work was supported by KAKENHI (Grant-in-Aid for Scientific Research (S) 17H06103).

References

[1] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[2] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171 (5-6):286–310, 2007.

[3] M. Caminada, S. Sá, J.F.L. Alcântara, and W. Dvořá. On the difference between assumption-based argumentation and abstract argumentation. In *Proc. BNAIC'2013*, pages 25–32, 2013.

[4] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[5] P.M. Dung, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170(2):114-159, 2006.

[6] P.M. Dung, R.A. Kowalski, and F. Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218, Springer, 2009.

[7] P.M. Dung and P.M. Thang. Closure and consistency in logic-associated argumentation. *Journal of Artificial Intelligence Research*, 49:79-109, 2014.

[8] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–385, 1991.

[9] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[10] C. Sakama and K. Inoue. Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation*, 5(3):265–285, 1995.

[11] C. Schulz and F. Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59–110, 2016.

[12] F. Toni. Assumption-Based Argumentation for Closed and Consistent Defeasible Reasoning. In *Proc. JSAI'2007*, volume 4914 of LNCS, pages 390–402, Springer, 2007.

[13] T. Wakaki. Assumption-based argumentation equipped with preferences and its application to decision-making, practical reasoning, and epistemic reasoning. *Computational Intelligence*, 33(4):706–736, 2017.

[14] T. Wakaki. Extended Abduction in Assumption-Based Argumentation. In *Proc. IEA/AIE'2019*, volume 11606 of LNAI, pages 593–607, Springer, 2019.

doi:10.3233/FAIA200522

Deductive and Abductive Reasoning with Causal and Evidential Information

Remi WIETEN [a,1], Floris BEX [a,b], Henry PRAKKEN [a,c] and Silja RENOOIJ [a]
[a] *Information and Computing Sciences, Utrecht University, The Netherlands*
[b] *Institute for Law, Technology and Society, Tilburg University, The Netherlands*
[c] *Faculty of Law, University of Groningen, The Netherlands*

Abstract. In this paper, we propose the information graph (IG) formalism, which provides a precise account of the interplay between deductive and abductive inference and causal and evidential information. IGs formalise analyses performed by domain experts using the informal reasoning tools they are familiar with, such as mind maps used in crime analysis. Based on principles for reasoning with causal and evidential information given the evidence, we impose constraints on the inferences that may be performed with IGs. Moreover, we propose an argumentation formalism based on IGs that allows arguments to be formally evaluated.

Keywords. Deduction, abduction, causal and evidential information, argumentation

1. Introduction

In the legal and forensic domain, reasoning about evidence plays a central role in the rational process of proof [1]. To aid in this process, various graph-based tools exist that allow domain experts to make sense of a mass of evidence in a case, including mind maps, argument diagrams and Wigmore charts [2]. Because of their informal nature, these tools typically do not directly allow for formal evaluation using AI techniques. Hence, we wish to formalise analyses performed with such tools in a manner that allows for formal evaluation and that adheres to principles from the literature on reasoning about evidence [1,3,4] while allowing inference to be performed in a manner closely related to the way in which inference is performed using such tools.

In reasoning about evidence, inference is often performed using domain-specific *generalisations* [1], also called defaults [4], which capture knowledge about the world in conditional form. We distinguish between *causal* generalisations (e.g. fire typically causes smoke) and *evidential* generalisations (e.g. smoke is typically evidence for fire) [1,4]. Inference can be performed in a *deductive*, or forward, fashion, where from a generalisation (e.g. fire typically causes smoke) and its antecedent (fire), the consequent (smoke) is defeasibly inferred; *abduction* [3] can also be performed with causal generalisations, where by affirming the consequent (smoke) the antecedent (fire) is defeasibly inferred. Pearl [4, p. 264] argued that people generally consider it difficult to express knowledge using only causal generalisations, and in an empirical study, van den Braak and colleagues [5] found that while there are situations in which subjects consistently choose either causal or evidential modelling techniques, there are also many examples in which people use both types of generalisations in their reasoning. For instance, subjects often considered testimonies to be evidential, whereas a motive for committing an act

[1] Corresponding author: g.m.wieten@uu.nl.

is considered a cause for committing that act. This illustrates that in formal accounts of reasoning about evidence, it is important to allow for both types of generalisations [1].

When performing analyses using aforementioned tools such as mind maps, domain experts naturally mix both causal and evidential generalisations and perform both deductive and abductive inferences, where the used generalisations and the inference type (deduction, abduction) are typically left implicit. Hence, in formalising such analyses we need a precise account of the interplay between the different types of inferences and generalisations and the constraints on performing inference we need to impose. In this paper we propose the *information graph* (IG) formalism, which provides such an account. IGs are knowledge representations that formalise analyses performed by domain experts using the informal reasoning tools they are familiar with in a manner that makes the causal and evidential generalisations used in performing inference explicit. Based on principles for reasoning with causal and evidential generalisations, we then define how deduction and abduction can be performed with IGs given a set of propositions labelled evidence. Most existing formalisms that allow both inference types with causal and evidential information are logic-based (e.g. [1,6]); instead, we propose a graph-based formalism to remain closely related to analyses performed using aforementioned graph-based tools.

Our argumentation formalism generates an abstract argumentation framework as in Dung [7], that is, a set of arguments with a binary attack relation, which thus allows arguments to be formally evaluated according to Dung's classical semantics. Moreover, our argumentation formalism adheres to the constraints imposed by Pearl's C-E system [4], which say that, in performing inference, care should be taken that no cause for an effect is inferred in case an alternative cause for this effect was already previously inferred.

The paper is structured as follows. In Section 2, we provide principles for reasoning about evidence. In Section 3, we present an example of an analysis performed using a mind mapping tool, which illustrates that both deduction and abduction is performed by domain experts, using both causal and evidential generalisations. Based on this example, in Section 4 we motivate and define our IG-formalism. In Section 5, we then propose an argumentation formalism based on our IG-formalism. In Section 6, we discuss related work. In Section 7, we discuss future work and conclude.

2. Reasoning about Evidence

In this section, we provide principles for and review the terminology used to describe reasoning about evidence. Inference is the process of inferring claims from the observed evidence using *generalisations* [1]. We distinguish between causal and evidential generalisations [1,4]. Causal generalisations are of the form '$c_1, \ldots, c_n$ *usually/normally/typically causes e*', whereas evidential generalisations are of the form '$e_1, \ldots, e_n$ *is usually/normally/typically evidence for c*'. We denote generalisations as *fire* → *smoke*, where *fire* is the generalisation's *antecedent* and *smoke* its *consequent*. A generalisation may have multiple antecedents, in which case the generalisation expresses that only the antecedents together allow us to infer the consequent. The notation $\rightarrow_c$ and $\rightarrow_e$ is used for causal and evidential generalisations, respectively.

Deductive Inference Inference can be performed in a deductive fashion, where from a causal or evidential generalisation and by affirming the antecedents, the consequent is inferred by modus ponens on the generalisation. Note that while deduction is typically equated with strict inference (cf. [8]) in which the consequent universally holds given the antecedents, we use the term 'deduction' for defeasible 'forward' inference in which the consequent tentatively holds given the antecedents (cf. [9]). Hence, deduction is not necessarily a stronger or more reliable form of inference than abduction.

Abductive Inference Abduction [3] can also be performed: from a causal generalisation and by affirming the consequent, the antecedents are inferred, since if the antecedents are true it would allow us to deductively infer the consequent modus-ponens-style. In case causes $c_1,\dots,c_n$ and $c'_1,\dots,c'_m$ are abductively inferred from common effect e using causal generalisations $c_1,\dots,c_n \rightarrow_C e$ and $c'_1,\dots,c'_m \rightarrow_C e$, then c_i and c'_j for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,m\}$ are considered to be *competing alternative explanations* for e. We assume that causes c_i (and c'_j) are not in competition among themselves. For instance, consider the causal generalisations *fire* $\rightarrow_C$ *smoke* and *smoke_machine* $\rightarrow_C$ *smoke*. By affirming the common consequent (*smoke*), *fire* and *smoke_machine* are inferred, which are then competing causes for *smoke*.

Mixed and Ambiguous Inference Deduction and abduction can be iteratively performed, where *mixed* abductive-deductive inference is also possible. Suppose that from the causal generalisation *fire* $\rightarrow_C$ *smoke* and by affirming the consequent (*smoke*), the antecedent (*fire*) is inferred. Now, if the additional causal generalisation *fire* $\rightarrow_C$ *heat* is provided, then consequent *heat* can be deductively inferred (or predicted [9]) as antecedent *fire* has been previously abductively inferred.

Mixed deduction, using both causal and evidential generalisations, can also be performed [6], but as noted by Pearl [4] deductively chaining a causal and an evidential generalisation can lead to undesirable results. Consider the example in which a causal generalisation *smoke_machine* $\rightarrow_C$ *smoke* and an evidential generalisation *smoke* $\rightarrow_E$ *fire* are provided. Deductively chaining these generalisations would make us infer there is a fire when seeing a smoke machine, which is clearly undesirable. Similarly, in performing mixed deductive-abductive inference, care should be taken that no cause for an effect is inferred if an alternative cause for this effect was already previously inferred. Consider the above example, where instead of an evidential generalisation *smoke* $\rightarrow_E$ *fire* a causal generalisation *fire* $\rightarrow_C$ *smoke* is provided. Upon seeing a smoke machine, this would make us infer there is a fire in case deduction and abduction are iteratively performed, which is again undesirable. Accordingly, we wish to prohibit these types of inference patterns, and refer to the constraint that no cause for an effect should be inferred if an alternative cause for this effect was already previously inferred as *Pearl's constraint* [4].

Finally, situations may arise in practice in which both deduction and abduction can be performed with the same causal generalisation. For instance, consider the causal generalisation *fire* $\rightarrow_C$ *smoke* and assume that both *fire* and *smoke* are affirmed but not observed, then both deduction and abduction can be performed to either infer *smoke* from *fire* or *fire* from *smoke*, respectively. The inference type is, therefore, *ambiguous*.

3. Example of an Analysis Performed Using a Mind Mapping Tool

In this section, we present an example of an analysis performed using a mind mapping tool [2], which is an example of a tool typically used by domain experts, for instance in crime analysis. Based on this example, we motivate and illustrate the design choices for our IG-formalism in Section 4. A mind map usually takes the shape of a diagram in which hypotheses and claims are represented by boxes and underlined text, and undirected edges symbolise relations between these hypotheses and claims. The mind map represents various scenario-elements and the crime analyst uses evidence to support or oppose these elements, indicated by plus and minus symbols, respectively.

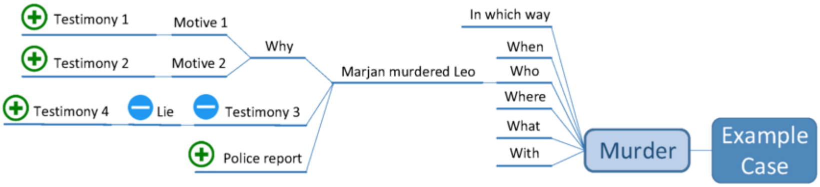


Figure 1. Example of a partially filled out mind map.

Example 1 *An example of a partially filled out mind map is depicted in Figure 1, which also serves as our running example. In this example, adapted from [1], the high-level hypothesis 'Murder' is considered. The case concerns the murder of Leo de Jager. Leo's body was found on the property of Marjan van der E.; we are interested in her involvement in the murder. As a police report (*Police report*) indicates that Leo's body was found on Marjan's property, claim* Marjan murdered Leo *is added as an answer to the 'Who' question. By means of a plus symbol and an undirected edge connecting the evidence to the claim, it is indicated that the police report supports the claim that Marjan murdered Leo. Possible motives (*Motive 1 *and* Motive 2*) are provided as to why Marjan may have wanted to murder Leo, which are connected to the 'Why' question via undirected edges.* Testimony 1 *and* Testimony 2 *support these two motives, indicated by the plus symbols connected to these claims. In her testimony (*Testimony 3*), Marjan denied any involvement in the murder of Leo, which is indicated by a minus symbol. This opposes the claim that Marjan murdered Leo. Further testimony (*Testimony 4*) indicates that Marjan had reason to lie when giving her testimony (*Lie*). By means of a minus symbol and an undirected edge connecting* Lie *to* Testimony 3*, it is indicated that this claim weakens the inference from her testimony to the claim that she did not murder Leo.* □

As the edges in a mind map are undirected, it is unclear from this graphical representation alone which types of generalisations and inferences were used in constructing this map. Establishing this with certainty would require directly consulting the domain experts involved in constructing the chart. We note, however, that the reasoning performed in constructing this mind map can be interpreted in at least two possible ways. One interpretation is that the domain expert first (preliminarily) inferred that Marjan murdered Leo from the police report via deduction using the evidential generalisation *Police report* $\to_e$ *Marjan murdered Leo*, and then abductively inferred the two possible motives using the causal generalisations g_i: *Motive i* $\to_c$ *Marjan murdered Leo*; $i = 1,2$. These two causes are then competing alternative explanations as to why Marjan murdered Leo and are subsequently grounded in evidence, namely via deduction from the testimonial evidence using evidential generalisations g'_j: *Testimony j* $\to_e$ *Motive j*; $j = 1,2$. An alternative interpretation is that the mind map was constructed iteratively from the observed evidence, where from testimonial evidence the motives are inferred via deduction using evidential generalisations g'_1 and g'_2. The claim that Marjan murdered Leo is then inferred modus-ponens style: from causal generalisations g_1 and g_2 and the previously inferred antecedents, the consequent is deductively inferred. In this way, the two motives are not in competition for the common effect that Marjan murdered Leo.

Lastly, note that in mind maps the exact manner in which claims and links conflict is not precisely specified: a minus symbol can either indicate support for the opposing claim (e.g. *Testimony 3* supports the negation of *Marjan murdered Leo*) or indicate an exception to the performed inference step (e.g. *Lie* opposes the inference step from *Testimony 3* to the negation of *Marjan murdered Leo*).

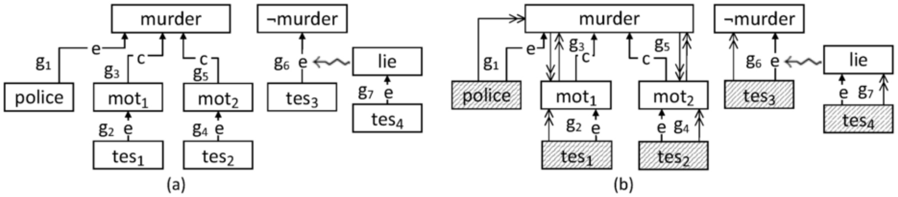


Figure 2. An IG corresponding to a possible interpretation of the mind map of Figure 1 (a); the same IG, where evidence set **E** (shaded) and resulting inference steps (↠) are also indicated (b).

4. The Information Graph Formalism

The example from Section 3 makes it plausible that both deduction and abduction is performed by domain experts when performing analyses using reasoning tools they are familiar with. In performing such analyses, the used generalisation, as well as the inference type (deduction, abduction), are left implicit. Furthermore, the assumptions of domain experts underlying their analyses are typically not explicitly stated, making these analyses ambiguous to interpret. For our current purposes of providing a precise account of the interplay between the different types of inferences and generalisations, we wish to formalise and disambiguate these analyses in a manner that makes the used generalisations explicit. *Information graphs* (IGs), which we define in Section 4.1, are knowledge representations that explicitly describe causal and evidential generalisations in the graph. In Section 4.2, we define how deductive and abductive inferences can be read from IGs, based on the principles for reasoning about evidence discussed in Section 2.

4.1. Information Graphs

IGs are defined as follows.

Definition 1 (Information graph) *An* information graph *(IG) is a directed graph $G = (\mathbf{P}, \mathbf{A})$, where $\mathbf{P}$ is a set of nodes representing propositions from a propositional literal language with ordinary negation symbol $\neg$. $\mathbf{A} = \mathbf{G} \cup \mathbf{X}$ is a set of directed (hyper)arcs with $\mathbf{G} \cap \mathbf{X} = \emptyset$, where $\mathbf{G}$ and $\mathbf{X}$ are sets of generalisation arcs and exception arcs, defined in Definitions 2 and 3, respectively.*

We write $p = -q$ in case $p = \neg q$ or $q = \neg p$. Note that an IG G does not have to be a connected graph (see Figure 2a). In the remainder of this paper, let $G = (\mathbf{P}, \mathbf{A})$ be an IG.

Definition 2 (Generalisation arc) *A* generalisation arc *$g \in \mathbf{G} \subseteq \mathbf{A}$ is a directed (hyper)arc $g\colon \{p_1, \dots, p_n\} \to p$, indicating a generalisation with antecedents $\mathbf{P_1} = \{p_1, \dots, p_n\} \subseteq \mathbf{P}$ and consequent $p \in \mathbf{P} \setminus \mathbf{P_1}$. Here, propositions in $\mathbf{P_1}$ are called the* tails *of g, denoted by $\mathbf{Tails}(g)$, and p is called the* head *of g, denoted by $Head(g)$. $\mathbf{G}$ divides into two disjoint subsets $\mathbf{G}^{\mathrm{c}}$ and $\mathbf{G}^{\mathrm{e}}$ of causal and evidential generalisation arcs, respectively.*

Curly brackets are omitted in case $|\mathbf{Tails}(g)| = 1$. In figures in this paper, generalisation arcs are denoted by solid (hyper)arcs, which are labelled ‘c’ for $g \in \mathbf{G}^{\mathrm{c}}$ and ‘e’ for $g \in \mathbf{G}^{\mathrm{e}}$.

Example 2 *In Figure 2a, an IG is depicted for a possible interpretation of the running example. First, we consider the undirected edges connected to the testimonies and the police report in the mind map of Figure 1. As noted earlier, testimonies are often considered to be evidential [5], where generalisations are of the form* ‘Testimony to fact x is normally evidence for x’. *Police reports can similarly be considered evidential. The IG therefore includes generalisation arcs $g_1, g_2, g_4, g_7 \in \mathbf{G}^{\mathrm{e}}$ to denote these generalisations.*

As tes_3 *concerns Marjan's testimony to denying any involvement in the murder,* $\neg\mathrm{murder}$ *is included in* $\mathbf{P}$ *and* $g_6\colon\ \mathrm{tes}_3 \to \neg\mathrm{murder}$ *in* $\mathbf{G}^{\mathrm{e}}$*. A motive for committing an act can be considered a cause for committing that act [5]. The IG therefore includes arcs* $g_3\colon$ $\mathrm{mot}_1 \to \mathrm{murder}$ *and* $g_5\colon\ \mathrm{mot}_2 \to \mathrm{murder}$ *in* $\mathbf{G}^{\mathrm{c}}$ *to denote these generalisations.* □

As generalisations hardly ever hold universally, exceptional circumstances can be provided under which a generalisation may not hold; hence, we allow exceptions to generalisations to be specified in IGs by means of exception arcs.

Definition 3 (Exception arc) *An* exception arc $x \in \mathbf{X} \subseteq \mathbf{A}$ *is a hyperarc* $x\colon\ p \rightsquigarrow g$*, where* $p \in \mathbf{P}$ *is called an* exception *to generalisation* $g \in \mathbf{G}$.

An exception arc directed from p to g indicates that p provides exceptional circumstances under which g may not hold.

Example 3 *Proposition* lie*, which states that Marjan had reason to lie when giving her testimony, provides an exception to evidential generalisation* $g_6\colon\ \mathrm{tes}_3 \to \neg\mathrm{murder}$ *in* $\mathbf{G}^{\mathrm{e}}$*. In Figure 2a, this is indicated by a curved hyperarc* $x\colon\ \mathrm{lie} \rightsquigarrow g_6$ *in* $\mathbf{X}$. □

4.2. Reading Inferences from Information Graphs

We now define how deductive and abductive inferences can be read from IGs. By itself, a generalisation arc only expresses that the tails together allow us to infer the head in case this generalisation is used in deductive inference, or that the tails together can be inferred from the head in case of abductive inference. Only when considering the available evidence can directionality of inference actually be read from the graph.

Definition 4 (Evidence set) *An* evidence set *is a subset* $\mathbf{E} \subseteq \mathbf{P}$ *for which it holds that for every* $p \in \mathbf{E}$*,* $\neg p \notin \mathbf{E}$*.*

In the remainder of this paper, let $\mathbf{E}$ be an evidence set. The restriction that for every $p \in \mathbf{E}$ it holds that $\neg p \notin \mathbf{E}$ ensures that not both a proposition and its negation are observed. In figures in this paper, nodes in G corresponding to elements of $\mathbf{E}$ are shaded and all shaded nodes correspond to elements of $\mathbf{E}$. We emphasise that various sets $\mathbf{E}$ can be used to establish inferences from the same IG.

Example 4 *In the running example, the evidence consists of the testimonies and the police report. In Figure 2b, the IG of Figure 2a is again depicted, with nodes in* $\mathbf{E} = \{\mathrm{tes}_1,$ tes_2, tes_3, tes_4, $\mathrm{police}\}$ *shaded.* □

4.2.1. Deductive Inference

First, we specify under which conditions we consider a configuration of generalisation arcs and evidence to express deductive inference.

Definition 5 (Deductive inference) *Let* $p_1,\ldots,p_n,q \in \mathbf{P}$*, with* $q \notin \mathbf{E}$*. Then given* $\mathbf{E}$*,* q *is* deductively inferred from propositions $p_1,\ldots,p_n$ using a generalisation $g\colon\ \{p_1,\ldots,p_n\} \to$ q in $\mathbf{G}$*, denoted* $p_1,\ldots,p_n \twoheadrightarrow_g q$*, iff* $\forall p_i$*,* $i = 1,\ldots,n$*:*

1. $p_i \in \mathbf{E}$*, or;*
2. p_i *is deductively inferred from propositions* $r_1,\ldots,r_m \in \mathbf{P}$ *using a generalisation* $g'\colon\ \{r_1,\ldots,r_m\} \to p_i$*, where* $g' \in \mathbf{G}^{\mathrm{e}}$ *if* $g \in \mathbf{G}^{\mathrm{e}}$*, or;*
3. p_i *is abductively inferred from a proposition* $r \in \mathbf{P}$ *using a generalisation* $g'\colon\ \{p_i, r_1,\ldots,$ $r_m\} \to r$ *in* $\mathbf{G}^{\mathrm{c}}$*,* $g \neq g'$*,* $r_1,\ldots,r_m \in \mathbf{P}$ *(see Definition 6).*

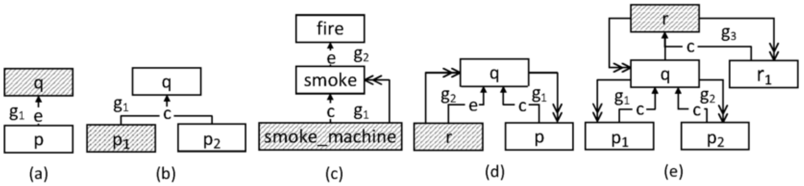


Figure 3. Examples of IGs illustrating the restrictions put on performing deduction within our IG-formalism (a-c); examples of IGs illustrating abductive inference (d-e).

In accordance with our assumptions stated in Section 2, deduction can be performed using generalisations in both $\mathbf{G}^{\mathrm{c}}$ and $\mathbf{G}^{\mathrm{e}}$. The condition $q \notin \mathbf{E}$ ensures that deduction cannot be performed to infer propositions that are already observed. Deduction can only be performed using a $g \in \mathbf{G}$ to infer $Head(g)$ from $\mathbf{Tails}(g)$ in case every tail $p_i \in \mathbf{Tails}(g)$ has been affirmed in that either $p_i \in \mathbf{E}$, p_i is itself deductively inferred, or p_i is abductively inferred. In correspondence with Pearl's constraint (see Section 2), we assume in condition 2 that a proposition $q \in \mathbf{P}$ cannot be deductively inferred from $p_1,\dots,p_n \in \mathbf{P}$ using a $g \in \mathbf{G}^{\mathrm{e}}$ if at least one of $p_1,\dots,p_n$ is deductively inferred using a $g' \in \mathbf{G}^{\mathrm{c}}$. Condition 3 of Definition 5 is further explained in Section 4.2.3, after abduction is defined.

Example 5 *Consider the running example. In Figure 2b,* mot_1 *and* mot_2 *are deductively inferred from* tes_1 *and* tes_2 *using generalisations* g_2 *and* g_4*, respectively, as* $\mathrm{tes}_1, \mathrm{tes}_2 \in \mathbf{E}$ *(condition* 1 *of Definition 5). Similarly,* murder, $\neg$murder *and* lie *are deductively inferred from* police, tes_3 *and* tes_4 *using generalisations* g_1*,* g_6 *and* g_7*, respectively, as* police, $\mathrm{tes}_3, \mathrm{tes}_4 \in \mathbf{E}$*. Proposition* murder *is also deductively inferred from* mot_1 *and* mot_2 *using causal generalisations* g_3 *and* g_5*, as* mot_1 *and* mot_2 *are deductively inferred (condition* 2 *of Definition 5). This illustrates mixed deduction using both types of generalisations.*□

We now illustrate the restrictions put on performing deduction within our IG-formalism.

Example 6 *Figure 3a depicts an example of an IG in which q cannot be deductively inferred from p using g_1, as $Head(g_1) = q \in \mathbf{E}$. In Figure 3b, q cannot be deductively inferred from p_1 and p_2 using g_1, as $p_2 \notin \mathbf{E}$ and p_2 is neither deductively nor abductively inferred. In Figure 3c, the example of Section 2 illustrating Pearl's constraint for deduction is modelled. As* smoke_machine $\in \mathbf{E}$*,* smoke *is deductively inferred from* smoke_machine *using* g_1 *by condition* 1 *of Definition 5.* fire *cannot in turn be inferred from* smoke *using* g_2*, as* $g_2 \in \mathbf{G}^{\mathrm{e}}$ *and* smoke *is deductively inferred using* $g_1 \in \mathbf{G}^{\mathrm{c}}$*.* □

4.2.2. Abductive Inference

Next, we specify under which conditions we consider a configuration of generalisation arcs and evidence to express abductive inference.

Definition 6 (Abductive inference) *Let $p_1,\dots,p_n,q \in \mathbf{P}$, with $\{p_1,\dots,p_n\} \cap \mathbf{E} = \emptyset$. Then given* $\mathbf{E}$, propositions $p_1,\dots,p_n$ are abductively inferred from q using a generalisation $g\colon \{p_1,\dots,p_n\} \to q$ in $\mathbf{G}^{\mathrm{c}}$, *denoted $q \twoheadrightarrow_g p_1;\dots;q \twoheadrightarrow_g p_n$, iff:*

1. *$q \in \mathbf{E}$, or;*
2. *q is deductively inferred from propositions $r_1,\dots,r_m \in \mathbf{P}$ using a generalisation $g'\colon r_1,\dots,r_m \to q$ in $\mathbf{G}$, $g \neq g'$ (see Definition 5), where $g' \in \mathbf{G} \setminus \mathbf{G}^{\mathrm{c}}$, or;*
3. *q is abductively inferred from a proposition $r \in \mathbf{P}$ using a generalisation $g'\colon \{q, r_1,\dots, r_m\} \to r$ in $\mathbf{G}^{\mathrm{c}}$, $r_1,\dots,r_m \in \mathbf{P}$.*

In accordance with our assumptions stated in Section 2, abduction is modelled using only causal generalisations. The condition $\{p_1,\ldots,p_n\} \cap \mathbf{E} = \emptyset$ ensures that abduction cannot be performed to infer propositions that are already observed. Furthermore, abduction can only be performed using a $g \in \mathbf{G}^{\mathrm{c}}$ to infer $\mathbf{Tails}(g)$ from $Head(g)$ in case $Head(g)$ has been affirmed in that either $Head(g) \in \mathbf{E}$, $Head(g)$ is deductively inferred, or $Head(g)$ is itself abductively inferred. In correspondence with Pearl's constraint (see Section 2), we assume in condition 2 that propositions $p_1,\ldots,p_n \in \mathbf{P}$ cannot be abductively inferred from a proposition $q \in \mathbf{P}$ using a $g \in \mathbf{G}^{\mathrm{c}}$ if q is deductively inferred using a $g' \neq g \in \mathbf{G}^{\mathrm{c}}$.

Example 7 *In Figure 3d, p is abductively inferred from q using generalisation $g_1 \in \mathbf{G}^{\mathrm{c}}$ by condition 2 of Definition 6, as q has been deductively inferred from r using generalisation $g_2 \in \mathbf{G}^{\mathrm{e}}$. In Figure 3e, q and r_1 are abductively inferred from r using generalisation $g_3\colon \{q,r_1\} \rightarrow r$ in $\mathbf{G}^{\mathrm{c}}$ by condition 1 of Definition 6, as $r \in \mathbf{E}$. Then by condition 3 of Definition 6, p_1 and p_2 are abductively inferred from q using generalisations g_1 and g_2, respectively. Consider Figure 4b, which illustrates that Pearl's constraint for mixed deductive-abductive inference is adhered to (see Section 2). As* smoke_machine *$\in \mathbf{E}$,* smoke *is deductively inferred from* smoke_machine *using $g_1 \in \mathbf{G}^{\mathrm{c}}$.* fire *cannot be inferred from* smoke*, as $g_2 \in \mathbf{G}^{\mathrm{c}}$ (condition 2 of Definition 6).* □

4.2.3. Mixed Abductive-Deductive and Ambiguous Inference

As apparent from Definitions 5 and 6, mixed abductive-deductive inference can be performed within our IG-formalism.

Example 8 *In Figure 4a, the example of Section 2 illustrating mixed abduction-deduction is modelled. From* smoke *$\in \mathbf{E}$,* fire *is abductively inferred using g_1. Then* heat *is deductively inferred (or predicted) from* fire *using g_2 (Definition 5, condition 3).* □

The conditions under which we consider a configuration of generalisation arcs and evidence to express deduction and abduction according to Definitions 5 and 6 are not mutually exclusive. Under specific conditions, both inference types can be established from the same $g \in \mathbf{G}^{\mathrm{c}}$ in an IG given the provided evidence; the inference type is, therefore, ambiguous (see Section 2). Examples of such inferences are provided in Figure 2b.

5. An Argumentation Formalism Based on Information Graphs

Based on our IG-formalism, we now propose an argumentation formalism that allows for both deductive and abductive argumentation. Our approach generates an abstract argumentation framework as in Dung [7], that is, a set of arguments with a binary attack relation, which thus allows arguments to be formally evaluated according to Dung's classical semantics. In Section 5.1, we define arguments on the basis of a provided G and $\mathbf{E}$, which capture sequences of inference steps as defined in Definitions 5 and 6 starting with elements from $\mathbf{E}$. We then formally prove that arguments constructed on the basis of IGs conform to Pearl's constraint. In Section 5.2, we define several types of attacks between arguments on the basis of IGs and instantiate Dung's abstract approach.

5.1. Arguments

In defining arguments on the basis of a G and $\mathbf{E}$, we take inspiration from the definition of an argument as given in [8]. In what follows, for a given argument A, the function CONC returns its conclusion, SUB returns its sub-arguments (including itself), IMMSUB returns its immediate sub-arguments, GEN returns all the generalisations used in constructing A, and TOPGEN returns the last generalisation used in constructing A.

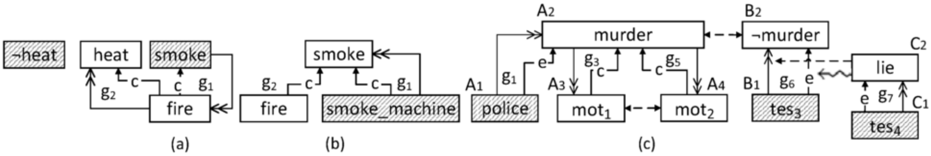


Figure 4. IGs illustrating mixed abduction-deduction (a) and Pearl's constraint for mixed deduction-abduction (b); adjustment to the IG of Figure 2b, where arguments and attacks (--→) are also indicated (c).

Definition 7 (Argument) *An* argument A on the basis of G and $\mathbf{E}$ *is any structure obtainable by applying one or more of the following steps finitely many times:*

1. *p if $p \in \mathbf{E}$, where:* $\textsc{Conc}(A) = p$*;* $\textsc{Sub}(A) = \{A\}$*;* $\textsc{ImmSub}(A) = \emptyset$*;* $\textsc{Gen}(A) = \emptyset$*;* $\textsc{TopGen}(A) =$ *undefined.*
2. *$A_1, \ldots, A_n \twoheadrightarrow_g p$ if $A_1, \ldots, A_n$ are arguments such that p is deductively inferred from* $\textsc{Conc}(A_1), \ldots, \textsc{Conc}(A_n)$ *using a generalisation* $g \in \mathbf{G} \setminus (\textsc{Gen}(A_1) \cup \ldots \cup \textsc{Gen}(A_n)), g\colon \{\textsc{Conc}(A_1), \ldots, \textsc{Conc}(A_n)\} \rightarrow p$ *according to Definition 5, where:* $\textsc{Conc}(A) = p$*;* $\textsc{Sub}(A) = \textsc{Sub}(A_1) \cup \ldots \cup \textsc{Sub}(A_n) \cup \{A\}$*;* $\textsc{ImmSub}(A) = \{A_1, \ldots, A_n\}$*;* $\textsc{Gen}(A) = \textsc{Gen}(A_1) \cup \ldots \cup \textsc{Gen}(A_n) \cup \{g\}$*;* $\textsc{TopGen}(A) = g$*.*
3. *$A' \twoheadrightarrow_g p$ if A' is an argument such that p is abductively inferred from* $\textsc{Conc}(A')$ *using a generalisation $g \in \mathbf{G} \setminus \textsc{Gen}(A'), g\colon \{p, p_1, \ldots, p_n\} \rightarrow \textsc{Conc}(A')$ for some propositions $p_1, \ldots, p_n \in \mathbf{P}$ according to Definition 6, where:* $\textsc{Conc}(A) = p$*;* $\textsc{Sub}(A) = \textsc{Sub}(A') \cup \{A\}$*;* $\textsc{ImmSub}(A) = \{A'\}$*;* $\textsc{Gen}(A) = \textsc{Gen}(A') \cup \{g\}$*;* $\textsc{TopGen}(A) = g$*.*

In the remainder of this paper, let the set of all arguments on the basis of G and $\mathbf{E}$ be denoted by $\mathcal{A}$. An argument $A \in \mathcal{A}$ is called a *premise argument* if only step 1 of Definition 7 is applied, *deductive* if only steps 1 and 2 are applied, *abductive* if only steps 1 and 3 are applied, and *mixed* otherwise. The restrictions in steps 2 and 3 that $g \notin (\textsc{Gen}(A_1) \cup \ldots \cup \textsc{Gen}(A_n))$ and $g \notin \textsc{Gen}(A')$, respectively, ensure that cycles in which two propositions are iteratively deductively and abductively inferred from each other using the same g are avoided in argument construction.

Example 9 *Consider the adjustment to the IG of Figure 2b depicted in Figure 4c, in which arguments on the basis of this IG and* $\mathbf{E} = \{$police, tes$_3$, tes$_4\}$ *are also indicated. According to step* 1 *of Definition 7, A_1:* police *is a premise argument. Based on A_1, deductive argument A_2: $A_1 \twoheadrightarrow_{g_1}$* murder *is constructed by step* 2 *of Definition 7, as* murder *is deductively inferred from* police *using g_1:* police → murder. *Then A_3: $A_2 \twoheadrightarrow_{g_3}$* mot$_1$ *is a mixed argument by step* 3 *of Definition 7, as* mot$_1$ *is abductively inferred from* murder *using g_3:* mot$_1$ → murder. *Consider Figure 3e, which illustrates step* 3 *in more detail. On the basis of this IG and $\mathbf{E} = \{r\}$, A'_1: r is a premise argument. From A'_1, arguments A'_2: $A'_1 \twoheadrightarrow_{g_3} r_1$ and A'_3: $A'_1 \twoheadrightarrow_{g_3} q$ are constructed by step* 3 *of Definition 7, as q and r_1 are abductively inferred from* $\textsc{Conc}(A'_1)$ *using g_3: $\{q, r_1\} \rightarrow r$. Again by step* 3*, A'_4: $A'_3 \twoheadrightarrow_{g_1} p_1$ and A'_5: $A'_3 \twoheadrightarrow_{g_2} p_2$ are constructed using g_1 and g_2, respectively.* □

In performing inference, care should be taken that no cause for an effect is inferred in case an alternative cause for this effect was already previously inferred (Pearl's constraint, see Section 2). In the context of IGs, for $g \in \mathbf{G}^{\mathrm{c}}$, propositions in $\mathbf{Tails}(g)$ express causes for the common effect expressed by $Head(g)$, and for $g \in \mathbf{G}^{\mathrm{e}}$, $Head(g)$ expresses a cause for propositions in $\mathbf{Tails}(g)$. Hence, in defining how inferences can be read from IGs, restrictions are put in Definitions 5 and 6 such that Pearl's constraint is adhered to. We now formally prove that Pearl's constraint is indeed never violated when constructing arguments on the basis of an IG G and an evidence set $\mathbf{E}$.

Proposition 1 (Adherence to Pearl's constraint) *Let $c_1, c_2 \in \mathbf{P}$ be alternative causes of $e \in \mathbf{P}$ in that either:*

1. $\exists g \in \mathbf{G}^{\mathrm{e}}$, $e \in \mathbf{Tails}(g)$, $Head(g) = c_1$, and either:
1a) $\exists g' \neq g \in \mathbf{G}^{\mathrm{e}}$, $e \in \mathbf{Tails}(g')$, $Head(g') = c_2$, or;
1b) $\exists g' \in \mathbf{G}^{\mathrm{c}}$, $c_2 \in \mathbf{Tails}(g')$, $Head(g') = e$.
2. $\exists g \in \mathbf{G}^{\mathrm{c}}$, $c_1 \in \mathbf{Tails}(g)$, $Head(g) = e$, and either:
2a) $\exists g' \neq g \in \mathbf{G}^{\mathrm{c}}$, $c_2 \in \mathbf{Tails}(g')$, $Head(g') = e$, or;
2b) $\exists g' \in \mathbf{G}^{\mathrm{e}}$, $e \in \mathbf{Tails}(g')$, $Head(g') = c_2$.

Assume arguments A and B exist in $\mathcal{A}$ with $\textsc{Conc}(B) = e$, $A \in \textsc{ImmSub}(B)$, and $\textsc{Conc}(A) = c_1$. Then no argument C exists in $\mathcal{A}$ with $B \in \textsc{ImmSub}(C)$, $\textsc{Conc}(C) = c_2$.

Proof. In constructing B from A, a generalisation $g \in \mathbf{G}^{\mathrm{e}}$, $e \in \mathbf{Tails}(g)$, $Head(g) = c_1$ could not have been used (case 1), as this would be an instance of abduction while per the restrictions of Definition 6 abduction can only be performed using generalisations $g \in \mathbf{G}^{\mathrm{c}}$. Thus, we only need to consider case 2, which is a deductive inference. First, consider case $2a$. Then by Definition 6 (condition 2), no argument C with $\textsc{Conc}(C) = c_2$ can be constructed from B using g'. Next, consider case $2b$. Then by Definition 5 (condition 2), no argument C with $\textsc{Conc}(C) = c_2$ can be constructed from B using g'. ■

5.2. Attack

In this section, several types of attacks between arguments on the basis of IGs are defined. In argumentation, two types of attacks are typically distinguished, namely rebuttal and undercutting attack [8]. We also distinguish a third type of attack, namely alternative attack, inspired by [6]. In our argumentation formalism, these three types of attacks directly follow from the constructed arguments and the specified exception arcs in an IG.

Definition 8 (Attack) *Let $A, B \in \mathcal{A}$. Then A* attacks *B iff A rebuts B, A undercuts B, or A alternative attacks B, as defined in Definitions 9, 10 and 11, respectively.*

First, rebuttal attack is considered, which informally is an attack on a $p \notin \mathbf{E}$.

Definition 9 (Rebuttal attack) *Let $A, B, B' \in \mathcal{A}$ with $B' \in \textsc{Sub}(B)$. Then A rebuts B (on B') iff $\textsc{Conc}(B') \notin \mathbf{E}$ and $\textsc{Conc}(A) = -\textsc{Conc}(B')$.*

Example 10 *Consider the IG of Figure 4c. Let A_1, A_2 be the arguments introduced in Example 9. Let B_1: tes_3 and let B_2: $B_1 \twoheadrightarrow_{g_6} \neg\text{murder}$. Then A_2 rebuts B_2 (on B_2) and B_2 rebuts A_2 (on A_2), as $\textsc{Conc}(A_2) = \text{murder}$, $\textsc{Conc}(B_2) = \neg\text{murder}$, and both* murder, $\neg$murder $\notin \mathbf{E}$. *This symmetric rebuttal is indicated in Figure 4c by means of a bidirectional dashed arc between these propositions. Consider again Example 8 and Figure 4a, in which* heat *is predicted from* fire. *Assume that contrary to this prediction we observe that there is no heat ($\neg$heat $\in \mathbf{E}$). Let A''_1: smoke; A''_2: $A''_1 \twoheadrightarrow_{g_1}$ fire; A''_3: $A''_2 \twoheadrightarrow_{g_2}$ heat; B''_1: $\neg$heat. Then B''_1 rebuts A''_2 (on A''_2), but A''_2 does not rebut B''_1 as $\textsc{Conc}(B''_1) \in \mathbf{E}$.* □

Next, undercutting attack is considered. Informally, an undercutter attacks an inference by providing exceptional circumstances under which the inference may not be applicable. Undercutting attacks between arguments follow from the specified exception arcs in G. Specifically, as an exception arc directed from $p \in \mathbf{P}$ to $g \in \mathbf{G}$ specifies an exception to g, an argument $A \in \mathcal{A}$ with $\textsc{Conc}(A) = p$ undercuts an argument $B \in \mathcal{A}$ with $g \in \textsc{Gen}(B)$.

Definition 10 (Undercutting attack) *Let $A, B, B' \in \mathcal{A}$ with $B' \in \textsc{Sub}(B)$. Then A* undercuts *B (on B') iff there exists an $x \in \mathbf{X}$ such that x: $\textsc{Conc}(A) \rightsquigarrow g$ and $\textsc{TopGen}(B') = g$.*

Example 11 *Consider the IG of Figure 4c. Let* B_1, B_2 *be the arguments introduced in Example 10. Let* $C_1\colon$ tes$_4$*;* $C_2\colon C_1 \twoheadrightarrow_{g_7}$ lie*. Then* C_2 *undercuts* B_2 *(on* B_2*), as* $x\colon$ lie $\rightsquigarrow g_6$ *in* $\mathbf{X}$ *and* $\textsc{TopGen}(B_2) = g_6$*. This attack is indicated in Figure 4c by means of a dashed arc directed from* lie *to inference* tes$_3 \twoheadrightarrow_{g_6} \neg$murder. □

Lastly, alternative attack is defined. Arguments are involved in alternative attack iff their abductively inferred conclusions are in competition for a common effect (see Section 2).

Definition 11 (Alternative attack) *Let* $A, B, B' \in \mathcal{A}$ *with* $B' \in \textsc{Sub}(B)$*. Then A alternative attacks B (on* B'*) iff there exists an argument* $C \in \textsc{ImmSub}(A) \cap \textsc{ImmSub}(B')$ *such that* $\textsc{Conc}(A)$ *and* $\textsc{Conc}(B')$ *are abductively inferred from* $\textsc{Conc}(\mathrm{C})$ *using generalisations* g *and* g' *in* $\mathbf{G}^{\mathrm{C}}$*,* $g \neq g'$*, respectively.*

Under the conditions set out in Definition 11, arguments $A_i\colon C \twoheadrightarrow_g p_i$ for $p_i \in \mathbf{Tails}(g)$ constructed from C via abduction are involved in alternative attack with $A'_j\colon C \twoheadrightarrow_{g'} p'_j$ for $p'_j \in \mathbf{Tails}(g')$ constructed from C via abduction. Arguments A_i (and A'_j) are not involved in alternative attack *among themselves*, in accordance with our assumption that the antecedents of causal generalisations are not in competition (see Section 2).

Example 12 *Consider the IG of Figure 4c. Let* A_1, A_2, A_3 *be the arguments introduced in Example 9, and let* $A_4\colon A_2 \twoheadrightarrow_{g_5}$ mot$_2$*, where* mot$_2$ *is abductively inferred from* murder*. Then* A_3 *and* A_4 *are involved in alternative attack, as indicated in Figure 4c by means of a bidirectional dashed arc between their conclusions.* □

Finally, we instantiate [7]'s abstract approach with arguments and attacks based on IGs.

Definition 12 (Argumentation framework) *An* argumentation framework defined by G and $\mathbf{E}$ *is a pair* $(\mathcal{A}, \mathcal{C})$*, where* $(A, B) \in \mathcal{C}$ *iff* $A \in \mathcal{A}$ *attacks* $B \in \mathcal{A}$ *(see Definition 8).*

Given an argumentation framework, we can use any semantics for argumentation frameworks as defined by [7] for determining the acceptability status of arguments (cf. [8]).

6. Related Work

In this paper, we have introduced the graph-based IG-formalism for deductive and abductive inference with causal and evidential information. Most related formalisms for inference with this type of information are logic-based. In the hybrid theory proposed by Bex [1], deduction and abduction are used in constructing evidential arguments and causal stories, which are completely separate entities with their own definitions related to conflict and evaluation. In comparison, our argumentation formalism based on IGs allows for the construction of both deductive and abductive arguments. Building on the hybrid theory, Bex proposed the integrated theory of causal and evidential arguments [6]. In the integrated theory, the roles of generalisation and inference are not separated; instead, causal and evidential inferences are defined and arguments are constructed by chaining such inferences. Actual abduction is thus not performed by constructing arguments.

Graph-based formalisms for reasoning with causality information have also been proposed, notably Pearl's causal diagrams [10]. Compared to IGs, causal diagrams do not allow for capturing asymmetric conflicts such as exceptions in the graph.

7. Conclusion and Future Work

In this paper, we have introduced the IG-formalism, which provides a principled way for representing and reasoning with causal and evidential information. Based on our IG-formalism, we have proposed an argumentation formalism that generates an abstract argumentation framework as in Dung [7], that is, a set of arguments with a binary attack relation, which thus allows arguments to be formally evaluated according to Dung's classical semantics. Moreover, our argumentation formalism adheres to the constraints imposed by Pearl's C-E system [4]. The added value of our argumentation formalism is that it allows both deductive and abductive argumentation, the latter of which has received relatively little attention in argumentation. In defining our argumentation formalism, we were inspired by the ASPIC$^+$ argumentation framework [8]. Our argumentation formalism can be regarded as an adaptation of a special case of ASPIC$^+$, which would among other things require introducing a new form of attack, namely alternative attack, and restricting the manner in which arguments are constructed within this framework. In future work, we intend to investigate the relations between our argumentation formalism and ASPIC$^+$ and whether Caminada and Amgoud's rationality postulates [11] are satisfied.

IGs formalise analyses performed by domain experts using the informal reasoning tools they are familiar with, such as mind maps. In interpreting a performed analysis as an IG, an additional knowledge elicitation step may be required, as the generalisations used in performing inference are typically left implicit in tools domain experts use. IGs may also be directly constructed by domain experts in case work. In our future work, we intend to investigate possible applications of our IG-formalism as intermediate formalism between informal tools and formalisms that allow for formal evaluation other than those for argumentation, for instance by extending on our previous work on facilitating Bayesian network (BN) construction from a preliminary form of IGs [12].

In our future work, we also intend to increase the expressivity of our IG-formalism by allowing generalisations that are neither causal nor evidential. For instance, definitions, or abstractions [13], allow for reasoning at different levels of abstraction, such as stating that guns can generally be considered deadly weapons.

References

[1] F. Bex. *Arguments, Stories and Criminal Evidence: A Formal Hybrid Theory*. Springer, 2011.

[2] A. Okada, S.J. Buckingham Shum, and T. Sherborne, eds., *Knowledge Cartography: Software Tools and Mapping Techniques*. Springer, 2nd ed., 2014.

[3] J.R. Josephson and S.G. Josephson. *Abductive Inference: Computation, Philosophy, Technology*. Cambridge University Press, 1994.

[4] J. Pearl. Embracing causality in default reasoning. *Artificial Intelligence*, 35(2): 259–271, 1988.

[5] S.W. van den Braak, H. van Oostendorp, H. Prakken, and G.A.W. Vreeswijk. Representing narrative and testimonial knowledge in sense-making software for crime analysis. In E. Francesconi, G. Sartor, and D. Tiscornia, eds., *Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference*, pp. 160–169. IOS Press, 2008.

[6] F. Bex. An integrated theory of causal stories and evidential arguments. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Law*, pp. 13–22. ACM Press, 2015.

[7] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2): 321–357, 1995.

[8] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2): 93–124, 2010.

[9] M. Shanahan. Prediction is deduction but explanation is abduction. In N.S. Sridharan, ed., *Proceedings of the International Joint Conference on Artificial Intelligence 89*, 1055–1060. Morgan Kaufmann, 1989.

[10] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd ed., 2009.

[11] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5–6): 286–310, 2007.

[12] R. Wieten, F. Bex, H. Prakken, and S. Renooij. Exploiting causality in constructing Bayesian networks from legal arguments. In M. Palmirani, ed., *Legal Knowledge and Information Systems. JURIX 2018: The Thirty-First Annual Conference*, pages 151–160. IOS Press, 2018.

[13] L. Console and D.T. Dupré. Abductive reasoning with abstraction axioms. In G. Lakemeyer and B. Nebel, eds., *Foundations of Knowledge Representation and Reasoning*, pp. 98–112. Springer, 1994.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200523

Enthymemes in Dialogues

Andreas XYDIS [1], Christopher HAMPSON, Sanjay MODGIL and Elizabeth BLACK
Department of Informatics, King's College London, London, United Kingdom

Abstract. Dialogical generalisations of formal logic-based argumentation are typically restricted to a limited set of locutions e.g., assert, why, claim or prefer. However, the use of enthymemes (i.e., arguments with incomplete logical structure) warrant extending this set of locutions. This paper formalises the use of additional novel locutions that account for the use of enthymemes and are typical of real world dialogues. We thus close the gap between formal logic-based models of dialogue and the kinds of dialogue studied by the informal logic community, which focus on more human-oriented models of dialogue. This is important if formal models of dialogues are to provide normative support for human-human debate, as well as for enabling computational and human agents to jointly reason via dialogue.

Keywords. enthymemes, locutions, dialogue, framework, argumentation

1. Introduction

In approaches to structured argumentation, arguments typically consist of a conclusion deductively and/or defeasibly inferred from some premises [1]. However, in practice, human agents typically assert 'incomplete' arguments known as enthymemes. Often, the intended 'complete' argument is obvious to the recipient of an enthymeme from the context and the shared common knowledge; otherwise, one may need to ask for clarification as to what is intended. Consider for example the following dialogue, which is annotated with the relevant locutions from the dialogue system proposed in this paper.

Example 1
1. Bob: *You can't afford to eat at a restaurant today.* (assert $\neg a$)
2. Alice: *Why not?* (why $\neg a$)
3. Bob: *Because you owe money and if you owe money then you probably can't afford to eat at a restaurant.* (because $c; c \Rightarrow \neg a; \neg a$)
4. Alice: *I made a deal with my creditors.* (assert f)
5. Bob: *So what?* (and-so)
6. Alice: *So I don't need to pay the bills today.* (hence $f; f \rightarrow \neg e; \neg e$)
7. Bob: *Why is that relevant?* (what-did-you-think-I-meant-by $c; c \Rightarrow \neg a; \neg a$)
8. Alice: *I thought that the reason you thought I owe money is because I have bills to pay today.* (assumed $e \rightarrow c; c$)
9. Bob: *No! I meant that you owe money because you need to pay Kate back today.* (meant $p \rightarrow c; c$)

[1] Corresponding Author E-mail: andreas.xydis@kcl.ac.uk.

Bob first asserts a claim without any supporting premises (1). The reasons for believing the claim are not clear to Alice, so she asks for clarification (2), which Bob provides (3). Notice that, when combined, (1) and (3) form a ‘complete argument’, hence they can both be considered enthymemes for this complete argument. Alice then presents an enthymeme (4) for an argument that she believes counters the argument Bob is making. Note that the enthymeme Alice presents does not explicitly contradict anything that Bob has said, and so Bob asks for clarification (5) on what he is meant to infer from this enthymeme, which Alice provides (6). However, Alice’s clarification still does not explicitly contradict anything Bob has said. Since Bob does not understand why Alice’s enthymeme is relevant to what he said, he asks Alice to explain what she thought he meant (7). Alice explains the assumption she had made (8), which Bob then corrects (9). This simple example illustrates the need for locutions that allow agents to both backward expand enthymemes (where missing premises are provided in 3 above) and forward expand enthymemes (where missing inferences are given, as in 6), and to request such expansions (2 and 5). It also shows the need for locutions that allow agents to ask what another agent has assumed was intended by an enthymeme (7), to answer such a question (8), and to correct any erroneous assumptions (step 9).

The primary contribution of this paper is to formalise a set of locutions, together with a protocol defined as constraints on when they may be made. We therefore support the use of enthymemes as seen in the example dialogue above, allowing agents to deal with any misunderstandings regarding what they revealed and what their counterpart thought was intended. Most works that formalise the use of enthymemes focus on how agents may construct enthymemes from an intended argument and reconstruct intended arguments from received enthymemes, based on assumptions about shared knowledge and context, e.g. [2,3,4,5,6]. Few works account for how enthymemes are handled during dialogues between human and/or computational agents. Notable exceptions include the work of Black and Hunter [7], Hosseini [8] and Dupin de Saint-Cyr [9], who formalise dialogue systems that accommodate enthymemes. However, although [7,8] employ locutions that capture the backward expansion of enthymemes and [9] addresses both backward and forward expansion of enthymemes, none of these address the misunderstandings that may occur due to the use of enthyemems in dialogue. This work therefore helps bridge the gap between formal logic-based models of dialogue and communication as witnessed in real-world dialogues. We thus contribute to theoretical foundations for dialogical models that enable communicative interactions between computational and/or human agents.

The paper is structured as follows. In Section 2 we review background for our work. In Sections 3 and 4 we present our work on enthymemes and their use in a dialogue framework. Section 5 then concludes and includes pointers to future work.

2. Preliminaries

In this paper, a *directed graph* G is a tuple $\langle N, E \rangle$ where $N \neq \emptyset$ is a set of nodes and $E \subseteq N \times N$ is a set of directed edges. A *directed tree* T is a special instance of G which has no cycles and a unique root node (denoted $\mathsf{Root}(T)$) such that there is a unique path from the root to each node in the graph. A *forest* is a disjoint union of directed trees.

Arguments and enthymemes are formalised within the *ASPIC*$^+$ framework for structured argumentation, which adopts a level of generality so as to subsume other approaches to structured argumentation, as well as providing argumentative formalisations of well known non-monotonic logics [10]. *ASPIC*$^+$ arguments are defined by an *argumentation theory* $AT = \langle AS, K\rangle$ where the *argumentation system* AS is a tuple $\langle L, (\bar{\cdot}), R, n\rangle$. L is a logical language, $(\bar{\cdot}) : L \longmapsto (2^L - \{\emptyset\})$ is a function that generalises the notion of negation, so as to declare that two formulae are in conflict. $R = R_s \cup R_d$ is a set of strict (R_s) and defeasible (R_d) inference rules and $n : R_d \rightarrow L$ is a naming function which assigns names to defeasible rules. A *knowledge base* $K = K_n \cup K_p$, where $K \subseteq L$, consists of disjoint sets of axiom (infallible) premises K_n and ordinary (fallible) premises K_p. Then an argument A is a tree, with *undirected* edges, whose leaves (denoted $\mathsf{Leaves}(A)$) belong to K, yielding the argument's claim (the root node) via application of strict and/or defeasible rules. Figurative representations of arguments depict application of strict, respectively defeasible rules, by solid, respectively dotted lines (see Fig.1.i). Each node of the tree represents an element $\alpha \in L$. The sub-arguments of A (denoted $\mathsf{Sub}(A)$) are sub-trees of A, which are themselves arguments whose root nodes are nodes in A (wff in L). Now note that enthymemes may be constructed by removal of a sub-argument whose conclusion is the antecedent of a strict/defeasible rule, while retaining the rule in the enthymeme, or indeed by removal of the conclusion of a sub-argument while retaining the inference rule. Hence, figurative representations of arguments in this paper will augment the standard representation of *ASPIC*$^+$ arguments to include the strict/defeasible inference rules applied (see Fig.1.ii). Finally, X *attacks* Y (where this attack may succeed as a *defeat*, contingent on preferences defined over the arguments X and the targeted sub-argument of Y) if X's claim conflicts with an ordinary premise or the consequent or name of a defeasible rule in Y (for details see [10]).

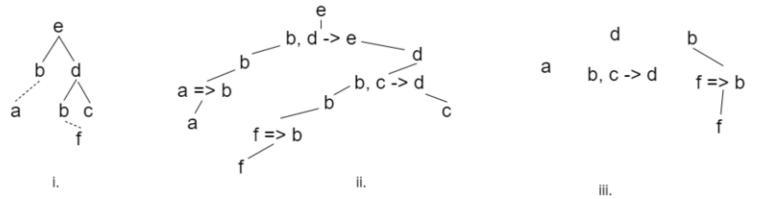


Figure 1. i. An *ASPIC*$^+$ argument. ii. An argument as represented in this paper. iii. An enthymeme constructed from the argument in ii.

3. Enthymemes

Enthymemes are incomplete arguments. Contrary to other approaches that handle enthymemes [7,8], we allow omission of an argument's claim, as well as its premises, and so may obtain a disjointed graph (as the claim is the root of the tree that is the intended argument). Hence we represent enthymemes as a forest of trees (see Fig.1.iii).

Since enthymemes are constructed from arguments, any node that is labelled with a proposition α (from L) may have at most one child, which must be labelled with an inference rule (from R) whose consequent is α (as *ASPIC*$^+$ ensures that an argument, and consequently all sub-arguments, can have at most one top rule from which the claim is inferred). The children of any node that is labelled with an inference rule (from R) must be labelled with an antecedent of that rule, and each child must have a different label so as to preclude multiple occurrences of the same proposition. Note that if an enthymeme includes the nodes n_i and n_j where either n_i is labelled with proposition p and n_j is labelled with an inference rule r whose antecedent includes p, or where n_i is labelled with rule r whose consequent is p and the node n_j is labelled with proposition p, this does not necessarily imply that n_i is a child of n_j. This allows us to handle cases where more than one sub-argument is used to support p within different branches of the same overall intended argument (see Fig.1.ii and 1.iii). Additionally, if an enthymeme E consists of a single tree, and its root and leaves are each labelled with an element of L (i.e., none are labelled with a rule) then E has the same structure as an $ASPIC^+$ argument [1]. We therefore consider arguments to be a special case of enthymemes (see Fig.1.ii).

Definition 1. Let $AS = \langle L, (\bar{\cdot}), R, n\rangle$. An **enthymeme** is $E = \langle \mathsf{Nodes}(E), \mathsf{Edges}(E), \mathsf{lab}_E\rangle$ such that:

- $\langle \mathsf{Nodes}(E), \mathsf{Edges}(E)\rangle$ is a forest;
- $\mathsf{lab}_E : \mathsf{Nodes}(E) \to L \cup R$;
- $\mathsf{Edges}(E) \subseteq \mathsf{Nodes}(E) \times \mathsf{Nodes}(E)$ such that if $(n_i, n_j) \in \mathsf{Edges}(E)$ then either:
  (a) $\mathsf{lab}_E(n_i) \in L$, $\mathsf{lab}_E(n_j) \in R$, $\mathsf{lab}_E(n_i)$ is the consequent of $\mathsf{lab}_E(n_j)$ and n_j is the only child of n_i, or;
  (b) $\mathsf{lab}_E(n_i) \in R$, $\mathsf{lab}_E(n_j) \in L$, $\mathsf{lab}_E(n_j)$ is an antecedent of $\mathsf{lab}_E(n_i)$ and there does not exist n_k, $k \neq j$, such that $\mathsf{lab}_E(n_j) = \mathsf{lab}_E(n_k)$ and $(n_i, n_k) \in \mathsf{Edges}(E)$;

$\mathsf{Rules}(E) = \{n_i \in \mathsf{Nodes}(E) \mid \mathsf{lab}_E(n_i) \in R\}$. $\mathsf{Leaves}(E) = \{n_i \in \mathsf{Nodes}(E) \mid \nexists (n_i, n_j) \in \mathsf{Edges}(E)\}$. $\mathsf{Top}(E) = \{n_i \in \mathsf{Nodes}(E) \mid \nexists (n_j, n_i) \in \mathsf{Edges}(E)\}$. The set of all enthymemes that can be constructed from an argumentation system AS is denoted E_{AS}.

If an enthymeme E includes a leaf node n labelled with a proposition $\phi \in L$, or a node labelled with a rule $r \in R$ whose antecedent is ϕ, but there is no child of r labelled with ϕ (see Fig.2.i), then there is no support for ϕ. We say that an enthymeme E' is the *backward expansion* of E on ϕ if and only if E' is a tree whose root is labelled with ϕ (see Fig.2.ii and 2.iii). Backward expansions thus expand the enthymeme 'downwards', beyond some leaf node.

Definition 2. Let $E = \langle \mathsf{Nodes}(E), \mathsf{Edges}(E), \mathsf{lab}_E\rangle$ and $E' = \langle \mathsf{Nodes}(E'), \mathsf{Edges}(E'), \mathsf{lab}_{E'}\rangle$ be enthymemes. Let $n_i \in \mathsf{Nodes}(E)$ such that either $n_i \in \mathsf{Leaves}(E)$ and $\mathsf{lab}_E(n_i) = \phi$ where $\phi \in L$; or $n_i \in \mathsf{Rules}(E)$ and there exists an antecedent ϕ of $\mathsf{lab}_E(n_i)$ such that there is no $n_j \in \mathsf{Nodes}(E)$ such that $(n_i, n_j) \in \mathsf{Edges}(E)$ and $\mathsf{lab}_E(n_j) = \phi$. We say that E' is a **backward expansion** of E on ϕ iff $\langle \mathsf{Nodes}(E'), \mathsf{Edges}(E')\rangle$ is a tree T' such that $\mathsf{lab}_{E'}(\mathsf{Root}(T')) = \phi$.

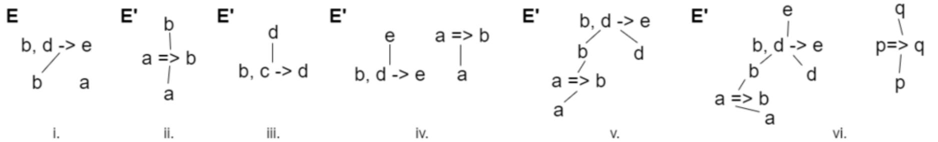


Figure 2. i. An enthymeme E. ii. The enthymeme E' is the backward expansion of E on b. iii. The enthymeme E' is the backward expansion of E on d. iv. The enthymeme E' is a forward expansion of E. v. and vi. do not represent forward expansions of E.

Since we allow omission of an argument's claim, an enthymeme E may entail some missing information. The missing information is all of the elements between the top nodes (nodes without any incoming edges) of E and the claim of the intended argument. So, we say that an enthymeme E' which consists of all (or some) of these information, including (or excluding) the claim, is the *forward expansion* of E. Forward expansions thus expand the enthymeme 'upwards', beyond one or more top nodes. For example, E' in Fig.2.iv is a forward expansion of E, but Fig.2.v and Fig.2.vi are not enthymemes that forward expand E, since a top node in E remains a top node in Fig.2.v and Fig.2.vi contains an arbitrary enthymeme.

Definition 3. Let $E = \langle \mathsf{Nodes}(E), \mathsf{Edges}(E), \mathsf{lab}_E \rangle$ and $E' = \langle \mathsf{Nodes}(E'), \mathsf{Edges}(E'), \mathsf{lab}_{E'} \rangle$ be enthymemes. We say that E' is a **forward expansion** of E iff:

- for every $n_i \in \mathsf{Top}(E)$ there exist $n_k, n_j \in \mathsf{Nodes}(E')$ such that $(n_k, n_j) \in \mathsf{Edges}(E')$, $\mathsf{lab}_{E'}(n_j) = \mathsf{lab}_E(n_i)$ and either $n_j \in \mathsf{Leaves}(E')$ or there exist $n_g \in \mathsf{Top}(E)$ and $n_h \in \mathsf{Leaves}(E')$ such that there is a path from n_j to n_h and $\mathsf{lab}_{E'}(n_h) = \mathsf{lab}_E(n_g)$;
- for every $n_i \in \mathsf{Top}(E')$ there exists $n_j \in \mathsf{Leaves}(E')$ such that there is a path from n_i to n_j and $\mathsf{lab}_{E'}(n_j) = \mathsf{lab}_E(n_k)$ where $n_k \in \mathsf{Top}(E)$.

4. Enthymeme Dialogue System

This section presents our novel two-party dialogue system for handling enthymemes. Our system permits the following locutions, described below in Table 1. From these locutions, we define a set of *moves* with which participants may move, query, and provide expansions of enthymemes. For the locutions hence, assumed, meant and agree, we employ (non-vocalised) variants, marked with either *bw*, *fw*, or *eq*, which dictate how the other participant is expected to respond (see Fig. 3 and Definition 4). Given a move's locution, Fig. 3 describes the reply structure between moves (i.e., if m replies to m' then m's locution must be a valid response to m''s locution). Note that if m is moved as a reply to m', this does not necessarily mean that m must immediately follow m' in the dialogue; agents are free to backtrack and reply to moves made previously, and it is possible that a single move may have multiple replies. Lastly, if a move m has a target m', this indicates that the content of m has been moved as a defeat against the content of m'.

Locution	Meaning
assert	Assert an enthymeme.
why	Question a particular element of a previous enthymeme, which is a request for the other participant to provide a backward expansion on that element.
because	Provide a backward expansion on a questioned element.
and-so	Request a forward expansion of a previous enthymeme.
hence^x	Provide a forward expansion of a previous enthymeme.
w.d.y.t.i.m.b.	Check the other participant's understanding of an enthymeme by asking *"what did you think I meant by … ".*
assumed^y	Provide their own interpretation of an enthymeme.
meant^y	Correct the other participant's interpretation of an enthymeme.
agree^y	Confirm the other participant's interpretation of an enthymeme.

Table 1. Table of possible locutions, with variants for $x \in \{eq, fw\}$ and $y \in \{eq, fw, bw\}$.

Definition 4. Let Loc denote the set of possible *locutions* provided in Table 1, let the reply structure be the binary relation $\rightarrow_{\mathrm{Loc}} \subseteq \mathrm{Loc}^2$ depicted in Fig.3 and let $\mathscr{M}$ denote the set of all possible moves. Given an $AS = \langle L, (\bar{\cdot}), R, n \rangle$, a set of enthymemes E_{AS} and participants $\mathscr{P} = \{Prop, Opp\}$, we define a **move** to be a tuple $m = \langle sender_m, location_m, content_m, reply_m, target_m \rangle$ where:

- $sender_m \in \mathscr{P}$ and $locution_m \in \mathrm{Loc}$;
- $reply_m \in \mathscr{M} \cup \{\emptyset\}$ is such that:
  - ◦ If $reply_m = \emptyset$ then $locution_m = \mathsf{assert}$,
  - ◦ If $reply_m = m' \in \mathscr{M}$ then $(locution_m, locution_{m'}) \in \rightarrow_{\mathrm{Loc}}$;
- $target_m \in \mathscr{M} \cup \{\emptyset\}$ is such that:
  - ◦ if $locution_m \in \{\mathsf{because}, \mathsf{assumed}^{bw}, \mathsf{meant}^{fw}, \mathsf{and\text{-}so}, \mathsf{stop}\}$ then $target_m = \emptyset$ (which is to say that these moves do not have a target);
  - ◦ if $locution_m \in \{\mathsf{assert}, \mathsf{why}\}$ then $target_m = reply_m$ (which is to say that these moves target the move that they reply to);
  - ◦ if $locution_m \in \{\mathsf{assumed}^{eq}, \mathsf{assumed}^{fw}, \mathsf{agree}^{eq}, \mathsf{agree}^{fw}, \mathsf{agree}^{bw}\}$, then $target_m = target_{m'}$ where $m' = reply_m$, (which is to say that these moves copy the target from the move they reply to);
  - ◦ if $locution_m = \mathsf{w.d.y.t.i.m.b.}$ then $target_m = target_{n''}$ where $n'' = target_{n'}$, $n' = target_{m'}$ and $m' = reply_m$ (which is to say that this move copies the target of the target of the move (m') it replies to);
  - ◦ if $locution_m \in \{\mathsf{hence}^{eq}, \mathsf{hence}^{fw}, \mathsf{meant}^{eq}, \mathsf{meant}^{bw}\}$, then $target_m = target_{m''}$ where $m'' = reply_{m'}$ and $m' = reply_m$ (which is to say that this move copies the target of the move (m'') which is replied to by the move m' that m replies to);
- $content_m \in E_{AS} \cup (E_{AS} \times L) \cup \{\emptyset\}$ is such that:
  - ◦ if $locution_m \in \{\mathsf{and\text{-}so}, \mathsf{stop}\}$, then $content_m = \emptyset$;
  - ◦ if $locution_m = \mathsf{assert}$, then $content_m \in E_{AS}$;
  - ◦ if $locution_m = \mathsf{why}$, then $content_m = (content_{m'}, \phi)$ where $\phi \in L$ and either $\phi = \mathsf{lab}_{content_{m'}}(n_i)$ for some leaf $n_i \in \mathsf{Leaves}(content_{m'})$ or ϕ is an antecedent of $\mathsf{lab}_{content_{m'}}(n_j)$ such that $n_j \in \mathsf{Rules}(content_{m'})$ and there does not exist $n_k \in \mathsf{Nodes}(content_{m'})$ such that $(n_j, n_k) \in \mathsf{Edges}(content_{m'})$ and $\phi = \mathsf{lab}_{content_{m'}}(n_k)$ and $m' = reply_m$;
  - ◦ if $locution_m = \mathsf{because}$, then $content_m$ is a backward expansion of A on ϕ where $content_{m'} = (A, \phi)$ and $m' = reply_m$;
  - ◦ if $locution_m = \mathsf{hence}^x$, then $content_m = content_{m''}$ or $content_m$ is a forward expansion of $content_{m''}$ where $m'' = reply_{m'}$ and $m' = reply_m$, for $x \in \{eq, fw\}$ respectively;
  - ◦ if $locution_m = \mathsf{w.d.y.t.i.m.b.}$, then $content_m = content_{n'}$ where $n' = target_{m'}$ and $m' = reply_m$;
  - ◦ if $locution_m = \{\mathsf{assumed}^{eq}, \mathsf{assumed}^{bw}, \mathsf{assumed}^{fw}\}$, then $content_m = content_{m'}$ or $content_m$ is a backward expansion or forward expansion of $content_{m'}$, respectively, where $m' = reply_m$;
  - ◦ If $locution_m = \{\mathsf{meant}^{eq}, \mathsf{meant}^{bw}, \mathsf{meant}^{fw}\}$, then $content_m \neq content_{m'}$ and either $content_m = content_{m''}$ or $content_m$ is a forward expansion or backward expansion of $content_{m''}$, respectively, where $m'' = reply_{m'}$ and $m' = reply_m$;
  - ◦ If $locution_m = \{\mathsf{agree}^{eq}, \mathsf{agree}^{bw}, \mathsf{agree}^{fw}\}$, then $content_m = content_{m'}$ and either $content_m = content_{m''}$ or $content_m$ is a backward expansion or forward expansion of $content_{m''}$, respectively, where $m'' = reply_{m'}$ and $m' = reply_m$;

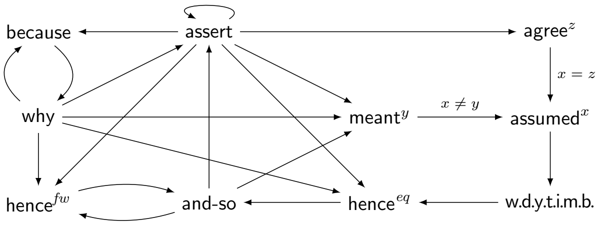


Figure 3. Illustration of the graph $\langle \mathrm{Loc}, \rightarrow_{\mathrm{Loc}} \rangle$, where $x \in \{fw, bw, eq\}$. For clarity, we have omitted the vertex stop $\in \mathrm{Loc}$ and edges $\{(\mathsf{stop}, L) : L \in \mathrm{Loc}\} \subseteq \rightarrow_{\mathrm{Loc}}$

.

We may then define an *enthymeme dialogue* (henceforth referred to as ‘dialogue’ for short) between two participants to be a (finite) sequence of moves such that each move replies to and targets some previous move or nothing, an assumed move is followed by a meant or agree move and the dialogue is concluded by two consecutive stop moves. We assume participants have the same logical language L, and the same functions $(\overline{\cdot})$ and (the naming function) n. Note that the first move of the dialogue is an assert move since it is the only move whose reply may be the emptyset. Table 2 shows how our system can capture the dialogue between Alice and Bob given in Example 1.

Definition 5. Let $AS_{\mathsf{Ag}} = \langle L_{\mathsf{Ag}}, (\overline{\cdot})_{\mathsf{Ag}}, R_{\mathsf{Ag}}, n_{\mathsf{Ag}} \rangle$ be an argumentation system for $\mathsf{Ag} \in \{Prop, Opp\}$, such that $L_{Prop} = L_{Opp}$, $(\overline{\cdot})_{Prop} = (\overline{\cdot})_{Opp}$ and $n_{Prop} = n_{Opp}$. An enthymeme dialogue between *Prop* and *Opp* is a sequence of moves $d = [m_0, \ldots, m_\ell]$ such that for all $i \leq \ell$:

– $sender_{m_i} = Prop$ if i is even, otherwise $sender_{m_i} = Opp$;
– $target_{m_i}, reply_{m_i} \in \{\emptyset, m_0, \ldots, m_{i-1}\}$;
– If $locution_{m_{i-1}} = \mathsf{assumed}^x$, for $x \in \{fw, bw, eq\}$, then $reply_{m_i} = m_{i-1}$.
– $locution_{m_{i-1}} = locution_{m_i} = \mathsf{stop}$ if and only if $i = \ell$.

5. Discussion

If logic-based models of argumentation based dialogue are to enable human-computer dialogue and provide normative support for human-human dialogue, they need to account for the ubiquitous use of enthymemes in real-world dialogues. To this end, our work complements and extends existing work [7,8,9] by broadening the set of locutions and protocol rules governing their use. To the best of our knowledge, our dialogue system is the first to provide locutions that allow recovery from misunderstanding that may arise due to the use of enthymemes. Indeed, it is instructive to note that commonly used locutions in real-world dialogues can effectively be understood as being motivated by the need to accommodate the use of enthymemes. Future work will show how an argument framework can be constructed on the basis of locutions moved during the dialogue such that, if participants play ‘logically perfectly’ (see [11]), the status of enthymemes in this framework corresponds to the status of these enthymemes in the Dung argument framework instantiated by the contents of all the locutions moved at that stage in the dialogue. Moreover, we will explore how enthymemes may be used strategically in persuasion dialogues to yield favourable outcomes for their users.

Step	Move	Enthymeme
1	$m_0 = (Prop, \mathsf{assert}, A_1, \emptyset, \emptyset)$	$A_1 = \langle \{n_1\}, \emptyset, \mathsf{lab}_{A_1} \rangle$, $\mathsf{lab}_{A_1}(n_1) = \neg a$
2	$m_1 = (Opp, \mathsf{why}, (A_1, \neg a), m_0, m_0)$	–
3	$m_2 = (Prop, \mathsf{because}, A_2, m_1, \emptyset)$	$A_2 = \langle \{n_1, n_2, n_3\}, \{(n_3, n_2), (n_2, n_1)\}, \mathsf{lab}_{A_2} \rangle$ $\mathsf{lab}_{A_2}(n_1) = c$, $\mathsf{lab}_{A_2}(n_2) = c \Rightarrow \neg a$, $\mathsf{lab}_{A_2}(n_3) = \neg a$
4	$m_3 = (Opp, \mathsf{assert}, B_1, m_2, m_2)$	$B_1 = \langle \{n_1\}, \emptyset, \mathsf{lab}_{B_1} \rangle$, $\mathsf{lab}_{B_1}(n_1) = f$
5	$m_4 = (Prop, \mathsf{and\text{-}so}, \emptyset, m_3, \emptyset)$	–
6	$m_5 = (Opp, \mathsf{hence}^{fw}, B_2, m_4, m_2)$	$B_2 = \langle \{n_1, n_2, n_3\}, \{(n_3, n_2), (n_2, n_1)\}, \mathsf{lab}_{B_2} \rangle$ $\mathsf{lab}_{B_2}(n_1) = f$, $\mathsf{lab}_{B_2}(n_2) = f \rightarrow \neg e$, $\mathsf{lab}_{B_2}(n_3) = \neg e$
–	$m_6 = (Prop, \mathsf{and\text{-}so}, \emptyset, m_5, \emptyset)$ $m_7 = (Opp, \mathsf{hence}^{eq}, B_2, m_6, m_2)$	– B_2 same as step 6
7	$m_8 = (Prop, \mathsf{w.d.y.t.i.m.b.}, A_2, m_7, \emptyset)$	A_2 same as step 3
8	$m_9 = (Opp, \mathsf{assumed}^{bw}, C, m_8, \emptyset)$	$C = \langle \{n_1, n_2\}, \{(n_2, n_1)\}, \mathsf{lab}_C \rangle$ $\mathsf{lab}_C(n_1) = e \rightarrow c$, $\mathsf{lab}_C(n_2) = c$
9	$m_{10} = (Prop, \mathsf{meant}^{fw}, A_3, m_9, \emptyset)$	$A_3 = \langle \{n_1, n_2\}, \{(n_2, n_1)\}, \mathsf{lab}_{A_3} \rangle$ $\mathsf{lab}_{A_3}(n_1) = p \rightarrow c$, $\mathsf{lab}_{A_3}(n_2) = c$
9'	$m'_{10} = (Prop, \mathsf{agree}^{bw}, C, m_9, \emptyset)$	C same as step 8

Table 2. Extended version of dialogue from Example 1 (*Prop* is Bob and *Opp* is Alice). The moves between steps 6 and 7 are excluded from Example 1 for simplicity, whereas step 9' is an alternative reply to m_9.

References

[1] Besnard P, Garcia A, Hunter A, Modgil S, Prakken H, Simari G, Toni F. Tutorials on Structured Argumentation. *Argument & Computation*. 2014; 5(1): 1-4.

[2] Black E, Hunter A. A relevance-theoretic framework for constructing and deconstructing enthymemes. *J. of Logic and Computation*. 2012; 22(1): 55-78.

[3] Hosseini S. A, Modgil S, Rodrigues O. Enthymeme Construction in Dialogues using Shared Knowledge. In: *Proc. of Int. Conf. on Computational Models of Argument*; 2014; p. 325-332.

[4] Hunter A. Real arguments are approximate arguments. In: *Proc. of AAAI Conf. on Artificial Intelligence*; 2007; p. 66-71.

[5] Panisson A. R, Bordini R. H. Uttering only what is needed: Enthymemes in multi-agent systems. In: *Proc. of Int. Conf. on Autonomous Agents and Multi-Agent Systems*; 2017; p.1670-1672.

[6] Walton D, Reed C. A. Argumentation schemes and enthymemes. *Synthese*. 2005; 145(3): 339-370.

[7] Black E, Hunter A. Using enthymemes in an inquiry dialogue system. In: *Proc. of Int. Conf. on Autonomous Agents and Multi-Agent Systems*; 2008; p. 437-444.

[8] Hosseini S. A. *Dialogues incorporating enthymemes and modelling of other agents' beliefs*. PhD Thesis. King's College London. 2017.

[9] Dupin de Saint-Cyr F. Handling enthymemes in time-limited persuasion dialogs. In: *Proc. of Int. Conf. on Scalable Uncertainty Management*; 2011; p.149–162.

[10] Modgil S, Prakken H. Abstract rule-based argumentation. Handbook of Formal Argumentation. 2018; 1: 286-361

[11] Prakken H. Coherence and flexibility in dialogue games for argumentation. Journal of logic and computation. 2005; 15(6): 1009-40.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200524

Continuum Argumentation Frameworks from Cooperative Game Theory

Anthony P. YOUNG [a,1], David KOHAN MARZAGÃO [a] and Josh MURPHY [a]
[a] *Department of Informatics, King's College London*

Abstract. We investigate the argumentation frameworks (AFs) that arise from multi-player transferable-utility cooperative games. These AFs have uncountably infinitely many arguments; arguments represent alternative payoff distributions to the players. We examine which of the various properties of AFs (from Dung's 1995 seminal paper) hold; we prove that these AFs are never finitary, never well-founded, always controversial and never limited controversial. We hope that this will encourage further exchange of ideas between argumentation and cooperative games.

Keywords. Abstract argumentation, cooperative game theory, dialogue

1. Introduction

Abstract argumentation theory is the branch of artificial intelligence (AI) concerned with resolving conflicts between disparate claims in a transparent and rational manner, while abstracting away from the contents of such claims by focussing on how they disagree (e.g. [6,12]). The resulting directed graph (digraph) representation of arguments (nodes) and the attacks between them (directed edges), called an *abstract argumentation framework* (AF), resolves conflicts by selecting suitable subsets of arguments, called *extensions*; this has been used to further understand and unify many areas within and outside of AI (see, e.g. [6,12]), where a situation can be represented by some AF such that the resulting extensions correspond to solutions for that situation; this gives a dialectical perspective to the situation that has been applicable to many practical domains (e.g. [11]).

Moreover, the "correctness" of argumentation theory has been shown by demonstrating that a correspondence exists between abstract argumentation theory and *cooperative game theory* (e.g. [5]), the branch of game theory (e.g. [20]) where agents that interact strategically can also work together under binding contracts [5, page 7] to earn more payoff than they can otherwise. This correspondence was first articulated in [6, Section 3.1], and then developed in [22], which further reinforces the applicability of concepts in abstract argumentation to problems of societal concern, but also allows for a cross-fertilisation of concepts between argumentation and cooperative games.

Abstract argumentation theory has mostly considered AFs that have a finite number of arguments (e.g. [1]). AFs that have an infinite number of arguments have not been considered as often, but they have been implicitly investigated in that all the results in [6] also hold for infinite AFs. Properties of sets of winning arguments in infinite AFs

[1] Corresponding Author: Department of Informatics, King's College London, Bush House, Strand Campus, 30 Aldwych, WC2B 4BG, London, United Kingdom. E-mail: peter.young@kcl.ac.uk.

were further investigated in [3], albeit in an abstract setting. It has been shown in [22] that uncountably infinite (continuum) AFs arise naturally from cooperative games. In this paper, we study these continuum AFs in their own right by asking whether various properties defined in [6, Section 2] hold for these AFs. We prove that these AFs fail to possess several desiderata due to the density of the continuum, specifically *finitariness* (all arguments have finitely many attackers), *well-foundedness* (there are no infinitely long "backwards" chains of attackers), *non-controversy* (no argument can simultaneously (indirectly) attack and defend other arguments), and *limited controversy* (there are no infinite chains of controversies); this makes precise the claim that these AFs are non-trivial, because one cannot invoke these properties to reduce the multiplicity of solutions.

The paper is structured as follows. In Section 2 we recap the relevant aspects of cooperative game theory and abstract argumentation theory. In Section 3, we investigate the properties of the continuum AFs arising from cooperative games and present our main results. We conclude with related and future work in Section 4.

2. Background

Notation: If X is a set, its *power set* is $\mathscr{P}(X)$ and its *cardinality* is $|X|$. $\mathbb{N}$ ($\mathbb{N}^+$) is the *set of (resp. positive) natural numbers*, with $|\mathbb{N}| =: \aleph_0$. $\mathbb{R}$ ($\mathbb{R}_0^+$ / $\mathbb{R}^+$) denotes the set of all (resp. non-negative / positive) real numbers, all with cardinality $2^{\aleph_0}$. For $a,b \in \mathbb{R}$, *the open interval from a to b* is the set $(a,b) := \{x \in \mathbb{R} | a < x < b\}$. For $n \in \mathbb{N}$, the *n-fold Cartesian power of* X is X^n, e.g. $X^2 = X \times X$. For sets Y and Z, and functions $f : X \to Y$ and $g : Y \to Z$, $g \circ f : X \to Z$ is *the composition of f then g*. $X \hookrightarrow Y$ denotes there is an injection from X to Y, including the case $X \subseteq Y$. For a function $f : X \to \mathbb{R}$, $f \geq 0$ abbreviates $(\forall x \in X)\, f(x) \geq 0$. An X-*sequence* is a function $\mathbb{N} \to X$, denoted as $\{x_k\}_{k \in \mathbb{N}}$.

2.1. Cooperative Game Theory

We review the basics of cooperative game theory (see, e.g. [5,22]). Given $m \in \mathbb{N}^+$, the **set of players** or **agents** is $N := \{1,2,3,\ldots,m\}$. Clearly, $|N| = m$. A **coalition** is any set $C \subseteq N$, where the **empty coalition** is $\varnothing$ and the **grand coalition** is N itself; each such C denotes that the players in C are cooperating under some contract. The **valuation function** $v : \mathscr{P}(N) \to \mathbb{R}$ such that $v(\varnothing) = 0$; $v(C)$ is C's payoff (in arbitrary units) as a result of the agents in C coordinating their strategies as agreed; this measures how "good" each C is. A **(cooperative) (m-player) game (in normal form)** is the pair $G := \langle N, v \rangle$.

The following properties are standard in the literature for v. We say v is **non-negative** iff $v \geq 0$. We say v is **monotonic** iff $(\forall C, C' \subseteq N)\,[C \subseteq C' \Rightarrow v(C) \leq v(C')]$. We say v is **constant-sum** iff $(\forall C \subseteq N)\, v(C) + v(N - C) = v(N)$. We say v is **super-additive** iff $(\forall C, C' \subseteq N)\,[C \cap C' = \varnothing \Rightarrow v(C \cup C') \geq v(C) + v(C')]$. We say v is **inessential** iff $\sum_{k=1}^{m} v(\{k\}) = v(N)$. For the rest of the games in this paper, we will assume that v is non-negative, super-additive and essential (i.e. not inessential). Intuitively, this means there is an incentive to cooperate such that agents working together will earn strictly more (as a coalition) than when working separately.

Given v, what coalitions will form? A **coalition structure**, CS, is a partition of N. As each coalition C earns a payoff $v(C) \geq 0$, we are interested in asking which ways of dividing $v(C)$ amongst the players $k \in C$ are "sensible". In this paper, we consider **transferable utility (TU) games**, which allows for *any* distribution of $v(C)$ to the players in C.[2] An **outcome** of a game is a pair $(CS, \mathbf{x})$, where CS is a coalition structure and $\mathbf{x} \in \mathbb{R}^m$ is a **payoff vector** that distributes the value of each $C \in CS$ to each $k \in C$. As usual in cooperative games (e.g. [5]), we focus on the case where $CS = \{N\}$, i.e. where all agents work together to form the grand coalition, and consider how the resulting payoff $v(N)$ can be distributed to each of the m players via the vector $\mathbf{x}$.

How should $v(N)$ be distributed amongst the m players? We say the payoff vector $\mathbf{x} := (x_1, \ldots, x_m) \in \mathbb{R}^m$ is **feasible** iff $\sum_{k \in N} x_k \leq v(N)$, **efficient** iff $\sum_{k \in N} x_k = v(N)$, **individually rational** iff $(\forall k \in N)\, v(\{k\}) \leq x_k$ and an **imputation** iff $\mathbf{x}$ is efficient and individually rational. We denote the **set of imputations** for a game G with $IMP(G)$, or just IMP if G is clear from context [6]. If G is inessential, then $IMP(G)$ is a singleton set by individual rationality, consisting of just $(v(\{1\}), v(\{2\}). \ldots, v(\{m\}))$, Otherwise, $IMP(G)$ is uncountably infinite; we focus on essential games to avoid this trivialisation.[3]

The solution concepts of cooperative games that we will consider are concerned with whether coalitions of agents are incentivised to defect from the grand coalition because they can earn strictly more payoff. Given a game $G = \langle N, v \rangle$, let $C \subseteq N$ and $\mathbf{x}, \mathbf{y} \in IMP$. We say **x dominates y via** C, denoted $\mathbf{x} \rightarrow_C \mathbf{y}$, iff (1) $(\forall k \in C)\, x_k > y_k$ and (2) $\sum_{k \in C} x_k \leq v(C)$, i.e. the agents are (1) strictly better off in C because (2) they will be earning enough as a coalition to be able to split the earnings among themselves. We call C the **defecting coalition**. It is easy to see that for any C, the binary relation $\rightarrow_C$ on IMP is irreflexive, acyclic, antisymmetric and transitive. Further, it can be shown that $\rightarrow_N = \varnothing$, $(\forall k \in N) \rightarrow_{\{k\}} = \varnothing$ and $\rightarrow_\varnothing = IMP^2$ (the total relation on IMP). It follows that if $m < 3$, $\rightarrow_C = \varnothing$ for any coalition C. The relation $\rightarrow$ is irreflexive, but not in general complete, transitive or acyclic (e.g. [19, Chapter 4]). Each cooperative game thus gives rise to an associated digraph, $\langle IMP, \rightarrow \rangle$, called an **abstract game**. The domination relation is empty for $m < 3$, so we will consider $m \geq 3$ to avoid this trivialisation.

We now review the solution concepts of cooperative games that are relevant to this paper.[4] Let $I \subseteq IMP$. Define the **forward set of** I to be $I^+ := \{\mathbf{y} \in IMP | (\exists \mathbf{x} \in I)\, \mathbf{x} \rightarrow \mathbf{y}\}$. If $I = \{\mathbf{x}\}$, then we write $\mathbf{x}^+ := \{\mathbf{x}\}^+$. Dually, we define the **backward set of** I, $I^- := \{\mathbf{y} \in IMP | (\exists \mathbf{x} \in I)\, \mathbf{y} \rightarrow \mathbf{x}\}$, and $\mathbf{x}^-$ is when $I = \{\mathbf{x}\}$. Define a function $U : \mathscr{P}(IMP) \rightarrow \mathscr{P}(IMP)$ to be $U(I) = IMP - I^+$. We say I is a **(von-Neumann-Morgenstern) stable set** iff $I = U(I)$ [20]. We say I is a **subsolution** iff $I \subseteq U(I)$ and $I = U^2(I) := U \circ U(I)$ [14]. We say I is the **supercore** iff I is the $\subseteq$-least subsolution [14]. We say I is the **core** iff $I = \{\mathbf{x} \in IMP | \mathbf{x}^- = \varnothing\}$, i.e. the set of all undominated imputations [7]. Lucas has shown that stable sets may not exist for cooperative games [9,10], although subsolutions, the supercore and the core always exist [13,14,15], but the core can be empty [4,18] exactly when the supercore is empty [14,21,22]. Each of these solution concepts offer alternative "socially acceptable" ways of distributing payoff to the players [20]. We now give two examples to illustrate some of these concepts.

Example 2.1. *[6, page 336] Let* $N = \{1,2,3\}$ *and* $v(C) = 0$ *if* $|C| \leq 1$, *and* $v(C) = 2$ *if* $|C| \geq 2$. *We show that* $I = \{(1,1,0), (1,0,1), (0,1,1)\}$ *is a stable set. Showing that* $I \subseteq U(I)$ *is equivalent to showing that no two elements in I dominate each other, which is true because we cannot have two components of* $\mathbf{x} \in I$ *being strictly greater than the two*

[2] The formalism of N and $v : \mathscr{P}(N) \rightarrow \mathbb{R}$ is different for non-TU games, see (e.g.) [5, Chapter 5].

[3] This corrects a minor error in [22, Corollary 1], where the assumption of $m \geq 2$ was omitted.

[4] These are *defection-based* solution concepts, whereas solution concepts based on *marginal contributions* (e.g. [17]) are currently outside the scope of this work.

corresponding components of $\mathbf{y} \in I$*, and considering two components suffices because* $\rightarrow_C = \varnothing$ *if* $|C| = 1$ *or* $C = N$*. To show that* $U(I) \subseteq I$*, it is equivalent to showing that every imputation* $\mathbf{x} = (x_1, x_2, x_3) \in IMP - I$ *is attacked by some imputation in I. By definition, we have* $0 \leq x_k \leq 2$ *for all* $k \in N$ *and* $x_1 + x_2 + x_3 = 2$*. Either (1)* $x_3 = 0$*, (2)* $x_3 > 1$ *or (3)* $x_3 \in (0,1)$*. (1) implies that* $x_1 + x_2 = 2$*, but as* $\mathbf{x} \notin I$*, WLOG assume* $x_1 < 1$ *and* $x_2 > 1$*, then* $(1,0,1) \rightarrow_{\{1,3\}} \mathbf{x}$*. Similarly, if* $x_1 > 1$ *and* $x_2 < 1$*, then* $(0,1,1) \rightarrow_{\{2,3\}} \mathbf{x}$*. (2) means* $x_1 + x_2 < 1$ *hence* $(1,1,0) \rightarrow_{\{1,2\}} \mathbf{x}$*. (3) means* $x_1 + x_2 < 2$*. If* $x_1 \geq 1$*, then* $x_2 < 1$ *hence* $(0,1,1) \rightarrow_{\{2,3\}} \mathbf{x}$*. If* $x_1 < 1$ *then* $x_2 \geq 1$ *so* $(1,0,1) \rightarrow_{\{1,3\}} \mathbf{x}$*. In all cases, some imputation in I dominates* $\mathbf{x}$*. Therefore, I is a stable set.*

Example 2.2. *[14, Example 5.1] Consider* $N = \{1,2,3\}$ *with* $v(\{1,2\}) = v(\{3,1\}) = v(N) = 1$*, and for all other S,* $v(S) = 0$*. We claim that* $I := \{(1,0,0)\}$ *is the core and that it is disconnected w.r.t.* $\rightarrow$*. Suppose* $(x_1, x_2, x_3) \rightarrow_C (1,0,0)$ *for some* $C \subseteq N$*, which is only possible for* $|C| = 2$*. As* $x_1 + x_2 + x_3 = 1$ *and* $x_k \geq 0$ *for* $k = 1,2,3$*, we cannot have* $x_1 > 1$ *and hence* $1 \notin C$*, so the only possible coalition is* $C = \{2,3\}$*, but then* $v(\{2,3\}) = 0$*, hence* $0 \leq x_2 + x_3 \leq 0$*, which means* $x_2 = x_3 = 0$*; this violates the domination condition* $x_2, x_3 > 0$*, hence for all* $\mathbf{x} \in IMP$*,* $\mathbf{x} \not\rightarrow (1,0,0)$*, therefore* $(1,0,0)$ *is amongst the undominated imputations. To show that* $(1,0,0)$ *is the only undominated imputation, consider* $(x_1, x_2, x_3) \notin I$*, hence for some* $\varepsilon > 0$*,* $x_1 = 1 - \varepsilon$ *and* $x_2 + x_3 = \varepsilon$*. Either (1) one of* x_2*,* x_3 *is zero or (2) neither are zero. In case (1), WLOG say* $x_2 = 0$*, then* $(x_1, x_2, x_3) = (1 - \varepsilon, 0, \varepsilon)$ *which is dominated by* $\left(1 - \frac{\varepsilon}{2}, \frac{\varepsilon}{2}, 0\right)$ *with defecting coalition* $\{1,2\}$*. In case (2),* $(x_1, x_2, x_3) = (1 - \varepsilon, x_2, \varepsilon - x_2)$ *for* $x_2 > 0$*, which is dominated by* $\left(1 - \frac{2\varepsilon}{3}, x_2 + \frac{\varepsilon}{3}, \frac{\varepsilon}{3} - x_2\right)$ *with defecting coalition* $\{1,2\}$*. Therefore, I is the core. Now suppose* $(1,0,0) \rightarrow_C (x_1, x_2, x_3)$*, but we know that* $x_2, x_3 \geq 0$ *so we cannot have* $0 > x_2, x_3$*, therefore* $2,3 \notin C$*, hence* $C = \{1\}$ *is the only possibility, but* $\rightarrow_{\{k\}} = \varnothing$ *for all* $k \in \mathbb{N}$*. Therefore,* $(1,0,0) \not\rightarrow \mathbf{x}$ *for all* $\mathbf{x} \in IMP$*. Hence* $(1,0,0)$ *is disconnected from all other imputations w.r.t.* $\rightarrow$*.*

2.2. Abstract Argumentation Theory

Recall that an **(abstract) argumentation framework** (AF) is a digraph $\langle A, R \rangle$, where A is **the set of arguments** and $R \subseteq A^2$ is **the attack relation** [6], where $(a,b) \in R$, alternatively denoted as $R(a,b)$, means argument a disagrees with argument b. Let $S \subseteq A$ for the remainder of this subsection. Define the **forward set of** S to be $S^+ := \{b \in A \,|\, (\exists a \in S) R(a,b)\}$. The **neutrality function** $n : \mathscr{P}(A) \rightarrow \mathscr{P}(A)$ is defined as $n(S) := A - S^+$. We say S is a **stable extension** iff $S = n(S)$. We say S is a **complete extension** iff $S \subseteq n(S)$ and $S = n^2(S) := n \circ n(S)$. We say S is a **preferred extension** iff it is a $\subseteq$-maximal complete extension. We say S is the **grounded extension** iff it is the $\subseteq$-least complete extension. We say S is the **set of all unattacked arguments** iff $S = \{a \in A | a^- = \varnothing\}$, where $S^- = \{a \in A \,|\, (\exists b \in S) R(b,a)\}$ and $a^- := \{a\}^-$. Stable extensions may not exist for AFs, although complete extensions always exist. Grounded, complete, preferred and stable extensions are collectively called the **Dung semantics**, and each defines a way of resolving the conflicts represented by R.

We say an AF $\langle A, R \rangle$ is **finitary** iff $(\forall a \in A) |a^-| < \aleph_0$. An AF is **well-founded** iff there is no A-sequence $\{a_k\}_{k \in \mathbb{N}}$ such that $(\forall k \in \mathbb{N}) R(a_{k+1}, a_k)$; if an AF is well-founded, then its grounded extension is stable [6, Theorem 30] and therefore there is only one subset of winning arguments. For $a, b \in A$, we say a is **indirectly attacking (defending)** b iff there is an odd (respectively, even)-length path from a to b. We say a is

controversial with respect to b iff a both indirectly attacks and indirectly defends b. We say a is controversial iff $(\exists b \in A)\, a$ is controversial w.r.t. b. An AF is **controversial** iff it has a controversial argument, else it is **uncontroversial**. An AF is **limited controversial** iff there is no A-sequence $\{a_k\}_{k\in\mathbb{N}}$ such that $(\forall k \in \mathbb{N})\, a_{k+1}$ is controversial w.r.t. a_k.

By interpreting $\langle IMP, \rightarrow\rangle$ as an AF, it has been shown that Dung's abstract argumentation semantics correspond to the solution concepts of cooperative games:

Abstract Argumentation	Cooperative Game	Reference
Argumentation Framework $\langle A, R\rangle$	Abstract Game $\langle IMP, \rightarrow\rangle$	[6, Section 3.1]
All unattacked arguments	The Core	[6, Theorem 38]
The Grounded Extension	The Supercore	[22, Theorem 5]
Complete Extensions	Subsolutions	[22, Theorem 3]
Preferred Extensions	$\subseteq$-maximal Subsolutions	[6, Section 3], [22, Theorem 3]
Stable Extensions	Stable Sets	[6, Theorem 37]

Table 2.1. Summarising the Correspondence Between Abstract Argumentation and Cooperative Game Theory

3. Some Properties of these Continuum Argumentation Frameworks

Having recapped how $\langle IMP, \rightarrow\rangle$ can be interpreted as an AF with uncountably infinitely many arguments, we now study these AFs in their own right, specifically whether these AFs satisfy or fail to satisfy the various properties defined by Dung in [6, Section 2], which we have recapped in Section 2.2. We prove that these AFs are not finitary, not well-founded, not limited controversial and not uncontroversial. This is due to the continuum nature of IMP arising from transferable utility, and shows that these AFs are not trivial in that we cannot appeal to these properties to conclude other properties that may reduce the multiplicity of the sets of winning arguments [6, Section 2].

Before we begin, let us recapitulate a simplification that does not lose generality. Let $\langle N, v\rangle$ be a game with abstract game $\langle IMP, \rightarrow\rangle$. We can convert it to its $(0,1)$**-normalised form**, which is the game $\left\langle N, v^{(0,1)}\right\rangle$, via the following affine transformation: $v^{(0,1)}(C) := Kv(C) + \sum_{k\in C} c_k$, where $\frac{1}{K} := v(N) - \sum_{k\in N} v(\{k\})$ and $(\forall k \in N)\, c_k := -Kv(\{k\})$. It follows that $(\forall k \in N)\, v^{(0,1)}(\{k\}) = 0$ and $v^{(0,1)}(N) = 1$. Further, the abstract game arising from the $(0,1)$-normalised form is digraph-isomorphic to $\langle IMP, \rightarrow\rangle$, and hence the solution concepts mentioned in Section 2.1 are preserved [2, Definition 2.7]. WLOG, we may assume that IMP is the **standard** $(m-1)$**-dimensional simplex**, $\{(x_1, \ldots, x_m) \in \mathbb{R}^m | (\forall 1 \leq k \leq m) x_k \geq 0, \sum_{k=1}^m x_k = 1\}$. Further, we will invoke the **Cantor-Schröder-Bernstein (CSB) theorem** (see, e.g. [8, Theorem 3.2]), which states that for (not necessarily finite) sets A and B, if $A \hookrightarrow B \hookrightarrow A$, then A and B have the same cardinality, in which case we write $A \cong B$. We assume standard results from set theory such as $(0,1) \cong \mathbb{R} \cong \mathbb{R}^m$ for every $m \in \mathbb{N}^+$.

First recall that the simplex is closed under affine combinations of two imputations $\mathbf{x}$ and $\mathbf{y}$, as imputations are vectors in $\mathbb{R}^m$ that can be added and scaled. Further, the imputations strictly in between $\mathbf{x}$ and $\mathbf{y}$ can be parameterised uniquely by $(0,1)$.

Lemma 3.1. *Let $t \in (0,1)$ and $\mathbf{x}, \mathbf{y} \in IMP$ be distinct. We have that $(1-t)\mathbf{x} + t\mathbf{y} \in IMP$ and $(0,1) \hookrightarrow IMP$ with rule $t \mapsto (1-t)\mathbf{x} + t\mathbf{y}$ is a well-defined injection.*

Proof. $t \in (0,1)$ implies $t, (1-t) > 0$. (Individual rationality) As each component is of the form $(1-t)x_k + ty_k$, we have $(1-t)x_k + ty_k \geq 0$ because $x_k, y_k \geq 0$, for all $k = 1, \ldots, m$. (Efficiency) $\sum_{k=1}^m [(1-t)x_k + ty_k] = (1-t)\sum_{k=1}^m x_k + t\sum_{k=1}^m y_k = 1 - t + t = 1$.

Assume for contradiction that $t \mapsto (1-t)\mathbf{x}+t\mathbf{y}$ is not injective. Therefore, there exists $t,t' \in (0,1)$ distinct such that $(1-t)\mathbf{x}+t\mathbf{y} = (1-t')\mathbf{x}+t'\mathbf{y}$. Basic algebra means we have $(t'-t)\mathbf{x} = (t'-t)\mathbf{y}$, but as $t'-t \neq 0$, it follows $\mathbf{x} = \mathbf{y}$, which is a contradiction. □

Clearly, this family of imputations between $\mathbf{x}$ and $\mathbf{y}$ contains uncountably infinitely many imputations, as the open line segment is a continuum.

Corollary 3.2. *The image set of the function defined in Lemma 3.1 is uncountable.*

Proof. By CSB, $\mathbb{R} \cong (0,1) \hookrightarrow \{(1-t)\mathbf{x}+t\mathbf{y} \in IMP | t \in (0,1)\} \subseteq IMP \subseteq \mathbb{R}^m \cong \mathbb{R}$. □

The continuum nature of the simplex allows us to "interpolate" a domination relation along the line segment joining an imputation and another imputation it dominates.

Theorem 3.3. *(Interpolation theorem) For* $\mathbf{x},\mathbf{y} \in IMP$ *and* $C \subseteq N$, *if* $\mathbf{x} \rightarrow_C \mathbf{y}$, *then* $(\forall t \in (0,1))\, \mathbf{x} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$.

Proof. Let $t \in (0,1)$ be arbitrary. We prove $\mathbf{x} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y}$ and $(1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$.

For the first domination, as $\mathbf{x} \rightarrow_C \mathbf{y}$, we know that $\sum_{k\in C} x_k \leq v(C)$. Further, $(\forall k \in C)\, x_k > y_k$. Let $k \in C$ be arbitrary, then we have $x_k > (1-t)x_k + ty_k \Leftrightarrow tx_k > ty_k \Leftrightarrow x_k > y_k$ (as $t > 0$), which is true. Therefore, $\mathbf{x} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y}$.

For the second domination, as $\mathbf{x} \rightarrow_C \mathbf{y}$, we know that $\sum_{k\in C} x_k \leq v(C)$. Further, $(\forall k \in C)\, x_k > y_k$. The second property means $\sum_{k\in C} x_k > \sum_{k\in C} y_k$. Therefore, $\sum_{k\in C} y_k \leq v(C)$. Now consider the quantity $\sum_{k\in C}[(1-t)x_k+ty_k]$. This is equal to $(1-t)\sum_{k\in C} x_k + t\sum_{k\in C} y_k \leq (1-t)v(C) + t\sum_{k\in C} y_k \leq (1-t)v(C)+tv(C) = v(C)$. Therefore, $\sum_{k\in C}[(1-t)x_k+ty_k] \leq v(C)$. Now for $k \in C$, $(1-t)x_k + ty_k > y_k \Leftrightarrow (1-t)x_k > (1-t)y_k$. As $t < 1$, we have $x_k > y_k$, which is true. Therefore, $(1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$. □

Theorem 3.3 also has the following consequences for whether the concepts in [6] apply: such AFs are not finitary (Corollary 3.4), not well-founded (Corollary 3.6), not uncontroversial (Corollary 3.7) and not limited controversial (Corollary 3.8).

Corollary 3.4. *If* $\langle IMP, \rightarrow\rangle$ *has a non-empty domination relation, then it is not finitary.*

Proof. If $\rightarrow \neq \varnothing$, then there are distinct $\mathbf{x},\mathbf{y} \in IMP$ such that for some non-empty $C \subseteq N$, $\mathbf{x} \rightarrow_C \mathbf{y}$. Therefore, $\{(1-t)\mathbf{x}+t\mathbf{y} \in IMP | t \in (0,1)\} \subseteq \mathbf{y}^- \subseteq IMP$, which means $\mathbf{y} \in IMP$ has uncountably infinitely many attackers. The result follows. □

We now generalise Theorem 3.3 to be able to compare two interpolated imputations along the open line segment between them.

Theorem 3.5. *(Double interpolation theorem) For* $\mathbf{x},\mathbf{y} \in IMP$ *and* $C \subseteq N$, *if* $\mathbf{x} \rightarrow_C \mathbf{y}$, *then* $(\forall s,t \in (0,1))$, *if* $s < t$, *then* $\mathbf{x} \rightarrow_C (1-s)\mathbf{x}+s\mathbf{y} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$.

Proof. For $\mathbf{z} := (1-s)\mathbf{x}+s\mathbf{y} \rightarrow_C \mathbf{y}$, let $u := \frac{t-s}{1-s} \in (0,1)$. Clearly, $(1-u)\mathbf{z}+u\mathbf{y} = (1-t)\mathbf{x}+t\mathbf{y}$, and by Theorem 3.3, $(1-s)\mathbf{x}+s\mathbf{y} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$. □

It follows from this that all such continuum AFs are not well-founded.

Corollary 3.6. *For* $\langle IMP, \rightarrow\rangle$, *if* $\rightarrow \neq \varnothing$, *then it is not well-founded.*

Proof. As $\rightarrow \neq \varnothing$, then consider $\mathbf{x} \rightarrow_C \mathbf{y}$. By Theorem 3.5, we have $s,t \in (0,1)$ such that if $s < t$, then $\mathbf{x} \rightarrow_C (1-s)\mathbf{x}+s\mathbf{y} \rightarrow_C (1-t)\mathbf{x}+t\mathbf{y} \rightarrow_C \mathbf{y}$. Define $\mathbf{z}_n := \left(1-\frac{1}{2^n}\right)\mathbf{x}+\frac{1}{2^n}\mathbf{y}$, for $n \in \mathbb{N}^+$. Clearly, $\mathbf{z}_{n+1} \rightarrow_C \mathbf{z}_n$ by Theorem 3.5. Therefore, the *IMP*-sequence $\{\mathbf{z}_n\}_{n\in\mathbb{N}^+}$ is an infinite backwards attacking chain, thus $\langle IMP, \rightarrow\rangle$ is not well-founded. □

Additionally, Theorem 3.3 shows that there is always a controversial argument.

Corollary 3.7. *For $\langle IMP, \rightarrow\rangle$, if $\rightarrow \neq \varnothing$, then the AF is not uncontroversial.*

Proof. As $\mathbf{x} \rightarrow_C \mathbf{y}$ means $\mathbf{x}$ (indirectly) attacks $\mathbf{y}$. We choose $t = \frac{1}{2} \in (0,1)$ in Theorem 3.3 such that $\mathbf{x} \rightarrow_C \frac{1}{2}(\mathbf{x}+\mathbf{y}) \rightarrow_C \mathbf{y}$, thus $\mathbf{x}$ indirectly defends $\mathbf{y}$. Therefore, $\mathbf{x}$ is controversial w.r.t. $\mathbf{y}$, which means $\langle IMP, \rightarrow\rangle$ is not uncontroversial. □

We show the weaker result of limited controversial is also never true.

Corollary 3.8. *For $\langle IMP, \rightarrow\rangle$, if $\rightarrow \neq \varnothing$, then the AF is not limited controversial.*

Proof. We construct an *IMP*-sequence $\{\mathbf{z}_k\}_{k\in\mathbb{N}}$ such that $(\forall k \in \mathbb{N})\, \mathbf{z}_{k+1}$ is controversial w.r.t. $\mathbf{z}_k$. Consider the infinite backwards attack chain from Corollary 3.6, such that for each $k \in \mathbb{N}$ and $\mathbf{z}_{k+1} \rightarrow_C \mathbf{z}_k$, we apply Theorem 3.3 with $t = \frac{1}{2}$, $\mathbf{x} = \mathbf{z}_{k+1}$ and $\mathbf{y} = \mathbf{z}_k$ to show that $\mathbf{z}_{k+1}$ also defends $\mathbf{z}_k$, and hence $\mathbf{z}_{k+1}$ is controversial w.r.t. $\mathbf{z}_k$, for all $k \in \mathbb{N}$. □

In summary, we have used the property of affine closure in a simplex to interpolate the domination $\mathbf{x} \rightarrow_C \mathbf{y}$ such that every payoff between $\mathbf{x}$ and $\mathbf{y}$ is attacked by $\mathbf{x}$ and attacks $\mathbf{y}$. It follows that $\langle IMP, \rightarrow\rangle$ is not finitary because $\mathbf{y}$ has uncountably infinitely many attackers. Further, $\langle IMP, \rightarrow\rangle$ is not well-founded because one can have an infinite backwards attack sequence from $\mathbf{y}$ with limit $\mathbf{x}$. Also, every intermediate point between $\mathbf{x}$ and $\mathbf{y}$ means that $\mathbf{x}$ is controversial w.r.t. $\mathbf{y}$, and interpolation means $\langle IMP, \rightarrow\rangle$ is also not limited controversial. From the perspective of abstract argumentation, the failure of these properties means we cannot invoke some results of [6, Section 2] to infer further properties of these AFs, e.g. that being uncontroversial means all preferred extensions are stable [6, Theorem 33(2)]. This means continuum AFs like those arising from cooperative games are non-trivial objects to analyse.

4. Conclusions, Future Work and Related Work

In this paper, we have investigated the continuum AFs arising from *m*-player essential transferable-utility cooperative games. In these AFs, the arguments represent the payoff distributions of all *m* players working together , and the attacks represent defection of some of the *m* players where they would each earn strictly more payoff . These AFs are "continuum" as they contain uncountably infinitely many arguments. We have shown that these AFs have several properties that are unlike finite AFs: they are not finitary, not well-founded, not uncontroversial, and not limited controversial. These results are important because they entail that such continuum AFs are challenging to deal with as we cannot simply use the results of [6, Section 2] to infer further properties.

As mentioned in Section 3, future work includes investigating conditions in which these continuum AFs are coherent and relatively grounded, which is challenging as our results show we cannot make use of simplifications such as [6, Theorem 33]. This could potentially contribute to cooperative game theory as articulating the conditions on $\langle N, v\rangle$ for when stable sets exist in $\langle IMP, \rightarrow\rangle$ is non-trivial; this is partly why game theorists moved away from cooperative games in the late 1970s [16]. Future work will investigate what further insights argumentation theory can offer.

As mentioned in Section 2, this paper builds on [6, Section 3.1] and [22]. However, we are not the first to investigate infinite AFs; they were investigated in [6] and furthered in [3] where general existence and uniqueness questions for extensions in infinite AFs are shown in an abstract setting. In contrast, this paper has provided an "authentic" example of infinite AFs that arise from cooperative games. We hope that future work will encourage further exchanges of ideas between argumentation and cooperative games.

References

[1] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[2] Pietro Baroni and Massimiliano Giacomin. Semantics of Abstract Argument Systems. In *Argumentation in Artificial Intelligence*, pages 25–44. Springer, 2009.

[3] Ringo Baumann and Christof Spanring. Infinite Argumentation Frameworks. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, pages 281–295. Springer, 2015.

[4] Olga N. Bondareva. Some Applications of Linear Programming Methods to the Theory of Cooperative Games. *Problemy Kibernetiki*, 10:119–139, 1963.

[5] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational Aspects of Cooperative Game Theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

[6] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77:321–357, 1995.

[7] Donald B. Gillies. Solutions to General Non-Zero-Sum Games. *Contributions to the Theory of Games*, 4(40):47–85, 1959.

[8] Thomas Jech. *Set Theory, the Third Millennium Edition, Revised and Expanded.* Springer Monographs in Mathematics. Springer, Berlin, 2003.

[9] William F. Lucas. A Game with No Solution. Technical report, RAND Corporation, Santa Monica, California, 1967.

[10] William F. Lucas. The Proof that a Game may not have a Solution. *Transactions of the American Mathematical Society*, 137:219–229, 1969.

[11] Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos Chesñevar, Wolfgang Dvořák, Marcelo. Falappa, et al. The Added Value of Argumentation. In Sascha Ossowski, editor, *Agreement Technologies*, pages 357–403. Springer, 2013.

[12] Iyad Rahwan and Guillermo R Simari. *Argumentation in Artificial Intelligence*, volume 47. Springer, 2009.

[13] Alvin E. Roth. A Lattice Fixed-Point Theorem with Constraints. *Bulletin of the American Mathematical Society*, 81(1):136–138, 1975.

[14] Alvin E. Roth. Subsolutions and the Supercore of Cooperative Games. *Mathematics of Operations Research*, 1(1):43–49, 1976.

[15] Alvin E Roth. A Fixed Point Approach to Stability in Cooperative Games. In *Fixed Points: Algorithms and Applications*, pages 165–180. Academic Press, 1977.

[16] Alvin E Roth and Robert B Wilson. How Market Design Emerged from Game Theory: A Mutual Interview. *Journal of Economic Perspectives*, 33(3):118–43, 2019.

[17] Lloyd S Shapley. A Value for n-Person Games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[18] Lloyd S. Shapley. On Balanced Sets and Cores. *Naval Research Logistics Quarterly*, 14(4):453–460, 1967.

[19] Lyn Carey Thomas. *Games, Theory and Applications*. Courier Corporation, 2012.

[20] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton university press, 1944.

[21] Anthony P. Young. Notes on Abstract Argumentation Theory. *ArXiv preprint arXiv:1806.07709*, 2018.

[22] Anthony P. Young, David Kohan Marzagão, and Josh Murphy. Applying Abstract Argumentation Theory to Cooperative Game Theory. *arXiv preprint arXiv:1905.10922*, 2019. Also available from `http://ceur-ws.org/Vol-2528/10_Young_et_al_AI3_2019.pdf,` last accessed 18/1/2020.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200525

Efficient Construction of Structured Argumentation Systems

Bruno YUN [a,1], Nir OREN [a] and Madalina CROITORU [b]
[a] *University of Aberdeen, Scotland*
[b] *University of Montpellier, France*

Abstract. We address the problem of efficient generation of structured argumentation systems. We consider a simplified variant of an ASPIC argumentation system and provide a backward chaining mechanism for the generation of structured argumentation graphs. We empirically compare the efficiency of this new approach with existing approaches (based on forward chaining) and characterise the benefits of using backward chaining for argumentation-based query answering.

Keywords. Argumentation, Backward chaining

1. Introduction

Logic-based argumentation is a powerful approach to reasoning with conflicting pieces of information. While argumentation traditionally generates arguments, most other approaches take a defeasible theory (DT) as input, together with a query. The output of the reasoning process is whether the query is, or is not, accepted by justified conclusions of the *saturated* DT, taking the interactions between conflicting facts into account. Most approaches to argumentation first compute all arguments, and detect conflicts between the arguments, thereby generating an argumentation graph [14, 19, 17]. Abstract argumentation semantics [8, 6] are then used to compute sets of justified arguments (called extensions), whose conclusions are compared to the query [7]. This follows the intuitions of what is known as *forward chaining* (FC) [15, 1]. The departure point of this work is the observation that FC is inappropriate for certain applications. Generating the entire set of arguments and identifying its extensions is computationally expensive [19, 18, 9], and one might want to answer a single query, in which case only arguments relevant to it should be generated. Here, we propose instead to use *backward chaining* (BC), focusing on an argumentation system (AS) based on a variant of ASPIC [4].

The focus of our approach is computational efficiency, though we recognise that in the worst case, the entire set of arguments may need to be generated to answer a query. However, we show in our empirical evaluation that this rarely occurs. We also analyse how the rules interact, proposing sufficient conditions to determine when, for a specific set of rules, fewer arguments will be generated by our approach.

[1] Corresponding Author: bruno.yun@abdn.ac.uk.

Our contributions are (1) the introduction of a new BC mechanism for ASPIC-like arguments; (2) a combinatorial structure that characterises the benefits of using BC over FC; and (3) an empirical evaluation demonstrating the potential impact of our approach.

The paper is structured as follows. In Section 2, we introduce the background notions. In Section 3, we introduce the *Graph of Rule Interaction*, a structure which captures how rules trigger each other. In Section 4, we describe how to generate arguments with BC. In Section 5, we show that the AS obtained via BC satisfies desirable properties and confirm them using an empirical evaluation. Section 6 summarises our approach.

2. Background

We begin by providing a brief overview of ASs built upon DTs. The notions presented here are used in the remainder of the paper. An argumentation formalism is usually built around an underlying logical language $\mathscr{L}$. We assume that $\mathscr{L}$ is made up of a set of literals and that we possess classical negation, i.e., we have a function "$\neg$" s.t. $\neg\psi = \phi$ iff $\psi = \neg\phi$ and $\neg\psi = \neg\phi$ iff $\psi = \phi$. A *strict rule* is then an expression $\mu : \phi_1, \dots, \phi_n \rightarrow \psi$ and a *defeasible rule* is an expression $\mu : \phi_1, \dots, \phi_n \Rightarrow \psi$ with $n \geq 0$ and $\{\phi_1, \dots, \phi_n, \psi\} \subseteq \mathscr{L}$. For a rule $r = \phi_1, \dots, \phi_n \rightsquigarrow \psi$, where $\rightsquigarrow \in \{\rightarrow, \Rightarrow\}$, we denote by $Body(r), Head(r), Name(r)$ and $Imp(r)$ the set $\{\phi_1, \dots, \phi_n\}$, the literal ψ, the literal μ, and $\rightsquigarrow$ respectively. A rule $\phi_1, \dots, \phi_n \rightsquigarrow \psi$ is said to be *applicable* on a set of literals $\mathscr{P} \subseteq \mathscr{L}$ iff $Body(r) \subseteq \mathscr{P}$. A set $\mathscr{P} \subseteq \mathscr{L}$ is said to be *consistent* iff there is no $\phi, \psi \in \mathscr{P}$ s.t. $\phi = \neg\psi$. The *closure* of $\mathscr{P}$ under a set of strict rules $\mathscr{S}$, denoted by $Cl_{\mathscr{S}}(\mathscr{P})$, is the minimal set s.t. (1) $\mathscr{P} \subseteq Cl_{\mathscr{S}}(\mathscr{P})$ and (2) for all strict rules $r = \phi_1, \dots, \phi_n \rightarrow \psi$ in $\mathscr{S}$ s.t. r is applicable to $Cl_{\mathscr{S}}(\mathscr{P})$, it holds that $\psi \in Cl_{\mathscr{S}}(\mathscr{P})$. A DT is $\mathscr{T} = (\mathscr{S}, \mathscr{D})$ where $\mathscr{S}$ is a set of strict rules and $\mathscr{D}$ is a set of defeasible rules [2].

Example 1. *We consider a DT containing the following information about John, a patient in a hospital:* r_1 : *"John has a prostate cancer"* ($\rightarrow c$)*;* r_2 : *"John is following a treatment for his cancer"* ($\rightarrow t$)*;* r_3 : *"John is a male patient"* ($\rightarrow m$)*;* r_4 : *"Studies show that there is no correlation between treating prostate cancer and bowel disorders"* ($\rightarrow \neg r_7$)*;* r_5 : *"John does not have abdominal pains"* ($\Rightarrow a$)*;* r_6 : *"If John does not have abdominal pain then he may not suffer from bowel disorder"* ($a \Rightarrow \neg b$)*;* r_7 : *"A patient with prostate cancer that is under treatment may suffer from bowel disorders"* ($c, t \Rightarrow b$)*. The DT is modelled by* $\mathscr{T} = (\mathscr{S}, \mathscr{D})$ *s.t.* $\mathscr{S} = \{r_1, r_2, r_3, r_4\}$ *and* $\mathscr{D} = \{r_5, r_6, r_7\}$*.*

Our AS is based on a version of ASPIC [4]. Here, an argument are formed by applying deductive rules [3], built upon other arguments. Given $\mathscr{T} = (\mathscr{S}, \mathscr{D})$, an argument A is of the form $A_1, \dots, A_n \rightsquigarrow \psi$, where $\{A_1, \dots A_n\}$ is a minimal set of arguments s.t. there exists an $r \in \mathscr{S} \cup \mathscr{D}$, where r is applicable to $\{Conc(A_1), \dots, Conc(A_n)\}$, $\rightsquigarrow = Imp(r)$ and $\psi = Head(r)$. Let $A = A_1, \dots, A_n \rightsquigarrow \psi$. Then the conclusion of A, denoted by $Conc(A)$, is ψ and the set of sub-arguments of A is $Sub(A) = Sub(A_1) \cup \dots \cup Sub(A_n) \cup \{A\}$. The top rule of A is $TR(A) = Conc(A_1), \dots, Conc(A_n) \rightsquigarrow \psi$. Given $\mathscr{T} = (\mathscr{S}, \mathscr{D})$, $Name : \mathscr{S} \cup \mathscr{D} \rightarrow \mathscr{L}$ returns the name of each rule and provides a handle for rules to prevent other rule applications.

[2] Please note that in our formalism, axioms (resp. ordinary premises) [14] are represented using strict (resp. defeasible) rules with empty bodies (c.f., ASPIC- [5])

Example 2 (Cont'd)**.** *There are 7 arguments:* $A_1 = \rightarrow c$*,* $A_2 = \rightarrow t$*,* $A_3 = \rightarrow m$*,* $A_4 = \Rightarrow h$*,* $A_5 = \rightarrow \neg r_7$*,* $A_6 = A_1, A_2 \Rightarrow b$ *and* $A_7 = A_4 \Rightarrow \neg b$*. Note that* $Name(c, d \Rightarrow b) = r_7$*.*

We consider that we are given a binary, total, reflexive and transitive preference relation $\preceq$ over arguments that reflects the quality of its underlying elements. One way of computing such a relation is to consider the type of rules (defeasible or strict) that are used in an argument and/or that defeasible rules have an associated strength, and *lifting* these preferences from rules to arguments [14], but we do not consider these aspects here. We write $A \prec B$ iff $A \preceq B$ and $B \npreceq A$, and $A \sim B$ iff $A \preceq B$ and $B \preceq A$.

A defeats B iff $B \preceq A$ and at least one of the following conditions is satisfied: (**Rebutting**) there exists $B' \in Sub(B)$ such that $Conc(B') = \neg Conc(A)$ and $Imp(TR(B'))$ is $\Rightarrow$[3] or (**Undercutting**) $Conc(A) = \neg Name(TR(B))$

An AS for a $\mathscr{T}$, denoted $\mathbb{AS}_{\mathscr{T}}$, is $(\mathscr{A}, \textsc{Def})$ where $\mathscr{A}$ is the set of arguments generated and $\textsc{Def} \subseteq \mathscr{A} \times \mathscr{A}$ is the defeat relation introduced above. Let $\mathbb{AS} = (\mathscr{A}, \textsc{Def})$, we say that $\mathbb{AS}' = (\mathscr{A}', \textsc{Def}')$ is a sub-system of $\mathbb{AS}$ iff $\mathscr{A}' \subseteq \mathscr{A}, \textsc{Def}' = \mathscr{A}' \times \mathscr{A}' \cap \textsc{Def}$.

Example 3 (Cont'd)**.** *If we assume that* $A_7 \prec A_6$ *and* $A_6 \preceq A_5$ *then* A_6 *is defeated by* A_5 *(undercutting) and* A_6 *defeats (rebutting)* A_7 *but* A_7 *does not defeat* A_6*. The AS corresponding to* $\mathscr{T}$ *is* $\mathbb{AS}_{\mathscr{T}} = (\{A_1, \ldots, A_7\}, \{(A_5, A_6), (A_6, A_7\})$*.*

3. The Graph of Rule Interaction

We define a new combinatorial structure over a DT called the Graph of Rule Interaction (GRI) that generalises the Graph of Rule Dependency [2]. The GRI captures both how rules trigger each other and identifies the possible conflicts between them. There are three main elements to the GRI: (1) a set of nodes representing the rules, (2) a set of *support links* representing how rules can activate each other and (3) a set of *attack links* representing conflicts amongst rules. While attacks links are binary, support links are many-to-one relationships as multiple rules can be necessary to trigger a rule.

Definition 1 (Graph of Rule Interaction)**.** *Let* $\mathscr{T} = (\mathscr{S}, \mathscr{D})$*. The GRI of* $\mathscr{T}$ *is* $GRI_{\mathscr{T}} = (\mathscr{N}, \mathscr{R}_s, \mathscr{R}_d)$*, where: (A)* $\mathscr{N} = \mathscr{S} \cup \mathscr{D} \cup \{\emptyset\}$ *represents the rules under consideration; (B)* $\mathscr{R}_s \subseteq 2^{\mathscr{N}} \times \mathscr{N}$ *s.t.* $\forall n_1 \in \mathscr{N}$ *and* $\forall N \subseteq \mathscr{N}, (N, n_1) \in \mathscr{R}_s$ *iff* $|N| = |Body(n_1)|$ *and* $\bigcup_{n \in N} Head(n) = Body(n_1)$*.* $\mathscr{R}_s$ *captures the support between rules and (C)* $\mathscr{R}_d \subseteq \mathscr{N} \times \mathscr{N}$ *s.t.* $\forall n_1, n_2 \in \mathscr{N}, (n_1, n_2) \in \mathscr{R}_d$ *iff at least one of the following conditions holds: (1)* $Head(n_1) = \neg Head(n_2)$ *and* $Imp(n_2) = \Rightarrow$ *or (2)* $Head(n_1) = \neg Name(n_2)$*.* $\mathscr{R}_d$ *captures potential defeats between rules. Note that* $Head(\emptyset) = Body(\emptyset) = Name(\emptyset) = \emptyset$*.*

Chains of rules can form where one rule is required for another to be applied, meaning that multiple rules can *support* others. We formalise this notion within the GRI.

Definition 2 (Support path)**.** *Let* $\mathscr{T} = (\mathscr{S}, \mathscr{D})$ *and* $GRI_{\mathscr{T}} = (\mathscr{N}, \mathscr{R}_s, \mathscr{R}_d)$*. The sequence* $(\mathscr{N}_1, \mathscr{N}_2, \ldots, \mathscr{N}_k)$ *is a support path to* $n \in \mathscr{S} \cup \mathscr{D}$ *in* $GRI_{\mathscr{T}}$ *iff all the following conditions are satisfied: (1)* $\forall i \in \{1, \ldots, k\}$*,* $\mathscr{N}_i \subseteq 2^{\mathscr{N}}$*, (2)* $\mathscr{N}_1 = \{\emptyset\}$ *and* $\mathscr{N}_k = \{\{n\}\}$ *and (3)* $\forall i \in \{2, \ldots, k\}$*,* $\forall N_j \in \mathscr{N}_i$ *and* $\forall n' \in N_j$*, there exists* $N' \in \mathscr{N}_{i-1}$ *s.t.* $(N', n') \in \mathscr{R}_s$*.*

[3] Note that we use *restricted rebut* and we did not evaluate unrestricted rebut due to space limitations.

An *activated* rule is a rule with a support path to it. The underlying idea is that such a rule's body can be obtained from other rules via the support path to it.

Definition 3 (Activated & Connected rule). *Let $\mathscr{T} = (\mathscr{S},\mathscr{D})$, $GRI_{\mathscr{T}} = (\mathscr{N},\mathscr{R}_s,\mathscr{R}_d)$ and $n,n' \in \mathscr{N}$. $r \in \mathscr{S} \cup \mathscr{D}$ is activated iff there exists a support path to r in $GRI_{\mathscr{T}}$. n is connected to n' iff there exists a sequence $(n_1,\dots,n_k)$ s.t. both of the following are satisfied: (1) for every $1 \leq i \leq k$, $n_i \in \mathscr{N}$ and n_i is activated and (2) for every $1 \leq i \leq k-1$, it holds that either $(n_i,n_{i+1}) \in \mathscr{R}_d$ or there exists $N \subseteq 2^{\mathscr{N}}$ s.t. $(N,n_{i+1}) \in \mathscr{R}_s$ with $n_i \in N$.*

Reasoning with BC requires a query that will be used to select the necessary rules in the original DT. We thus need to describe whether a rule is important for a given query. We refer to such rules as *potentially necessary* rules.

Definition 4 (Potentially necessary rule). *Let $l \in \mathscr{L}$ and $\mathscr{T} = (\mathscr{S},\mathscr{D})$. $r \in \mathscr{S} \cup \mathscr{D}$ is potentially necessary for l iff $\exists r' \in \mathscr{S} \cup \mathscr{D}$ s.t. $Head(r') = l$ and r is connected to r'.*

Example 4 (Cont'd). *$GRI_{\mathscr{T}} = (\mathscr{N},\mathscr{R}_s,\mathscr{R}_d)$ where $\mathscr{N} = \{\emptyset, r_1,r_2,r_3,r_4,r_5,r_6,r_7\}$, $\mathscr{R}_s = \{(\{\emptyset\},r_1),(\{\emptyset\}, r_2),(\{\emptyset\}, r_3),(\{\emptyset\}, r_4), (\{\emptyset\},r_5), (\{r_1,r_2\},r_7),(\{r_5\},r_6)\}$ and $\mathscr{R}_d = \{(r_4,r_7), (r_6,r_7),(r_7,r_6)\}$. The sequence $(\{\emptyset\},\{\{r_1\}, \{r_2\}\}, \{\{r_7\}\})$ is a support path to r_7. The rule r_5 is potentially necessary for b but r_3 is not.*

4. Backward Chaining for Argumentation

An argument for $l \in \mathscr{L}$ is an argument that concludes l. This notion is needed to define what an AS for a literal is.

Definition 5 (Argument for a literal). *Let $l \in \mathscr{L}$ and $\mathbb{AS}_{\mathscr{T}} = (\mathscr{A},\textsc{Def})$. $A \in \mathscr{A}$ is an argument for l iff $Conc(A) = l$. $\mathscr{A}_l \subseteq \mathscr{A}$ is the set of arguments for l in $\mathscr{A}$.*

Definition 6 (AS for a literal). *Let $l \in \mathscr{L}$ and $\mathbb{AS}_{\mathscr{T}} = (\mathscr{A},\textsc{Def})$. We say that the sub-system $\mathbb{AS}^l_{\mathscr{T}} = (\mathscr{A}',\textsc{Def}')$ of $\mathbb{AS}_{\mathscr{T}}$ is the AS for l (w.r.t. $\mathscr{T}$) iff $\mathscr{A}'$ is minimal (w.r.t. $\subseteq$) s.t. all the following are satisfied: **(1)** $\mathscr{A}_l \subseteq \mathscr{A}'$, **(2)** If $A \in \mathscr{A}'$ and $B \in \mathscr{A}$ s.t. $(B,A) \in \textsc{Def}$ then $B \in \mathscr{A}'$, **(3)** If $A \in \mathscr{A}'$ then $Sub(A) \subseteq \mathscr{A}'$ and (4) $\textsc{Def}' = \{(A,B) \in \textsc{Def} \mid A,B \in \mathscr{A}'\}$.*

Note that for a given DT $\mathscr{T}$, there is a *unique* AS for $l \in \mathscr{L}$ w.r.t. $\mathscr{T}$. Moreover, since $\mathbb{AS}^l_{\mathscr{T}} = (\mathscr{A}',\textsc{Def}')$ is a sub-system of $\mathbb{AS}_{\mathscr{T}} = (\mathscr{A},\textsc{Def})$, it has at most the same number of arguments and attacks, i.e. $|\mathscr{A}| \geq |\mathscr{A}'|$ and $|\textsc{Def}| \geq |\textsc{Def}'|$.

Example 5 (Cont'd). *$\mathbb{AS}^{\neg b}_{\mathscr{T}} = (\mathscr{A}',\textsc{Def}')$ where $\mathscr{A}' = \{A_1,A_2,A_4,A_5,A_6,A_7\}$ and $\textsc{Def}' = \{(A_5,A_6), (A_6,A_7)\}$. $\mathbb{AS}^{\neg b}_{\mathscr{T}}$ is the AS for $\neg b$ w.r.t. $\mathscr{T}$.*

We now focus on the generation of arguments from a DT, describing a two-step procedure for generating the AS using BC, i.e. from $\mathscr{T}$, we use BC to generate $\mathbb{AS}^l_{\mathscr{T}}$, for any $l \in L$. Our approach makes use of two algorithms: AL and ASG (see Figure 1).

AL: This algorithm generates $\mathscr{A}_l$. This procedure gathers all the rules that conclude l in R (line 5). It then constructs all the possible arguments that have a rule of R as top rule. (lines 6–9). Note that the parameter *seen* is used to avoid the generation of an infinite number of sub-arguments in case of rule cycles (for instance $p \to q$ and $q \to p$).

```
Require: $\mathscr{T} = (\mathscr{S}, \mathscr{D})$, $l \in \mathscr{L}$ and $seen \subseteq \mathscr{L}$
function AL($\mathscr{T}, l, seen$)
    $A \leftarrow \emptyset$
    if $l \in seen$ then
        return $\emptyset$
    $R \leftarrow \{r \in \mathscr{S} \cup \mathscr{D} \mid Head(r) = l\}$
    for $r \in R$ do
        $\mathscr{A}_B \leftarrow \emptyset$
        for $\phi \in Body(r)$ do
            $\mathscr{A}_B[\phi]$ = AL($\mathscr{T}, \phi, seen \cup \{l\}$)
    $Prod \leftarrow \bigtimes_{\phi \in Body(r)} \mathscr{A}_B[\phi]$
    for $\{A_1, \ldots, A_n\} \in Prod$ do
        $\rightsquigarrow_r \leftarrow Imp(r)$
        $A \leftarrow A_1, \ldots A_n \rightsquigarrow_r Head(r)$
        $\mathscr{A} \leftarrow \mathscr{A} \cup \{A\}$
    return $\mathscr{A}$
```

```
Require: $\mathscr{T} = (\mathscr{S}, \mathscr{D})$ and $l \in \mathscr{L}$
 1: function ASG($\mathscr{T}, l$)
 2:     $finished \leftarrow false$
 3:     $\mathscr{A}_{old} \leftarrow$ AL($\mathscr{T}, l, \emptyset$)
 4:     while not $finished$ do
 5:         $temp \leftarrow \mathscr{A}_{old}$
 6:         $\mathscr{A} \leftarrow temp$
 7:         for all $A \in temp$ do
 8:             $\mathscr{A} \leftarrow \mathscr{A} \cup Sub(A)$
 9:         $temp \leftarrow \mathscr{A}$
10:         for all $A \in temp$ do
11:             $\mathscr{A} \leftarrow \mathscr{A} \cup$ AL($\mathscr{T}, \neg Head(TR(A)), \emptyset$)
12:             $\mathscr{A} \leftarrow \mathscr{A} \cup$ AL($\mathscr{T}, \neg Name(TR(A)), \emptyset$)
13:         if $\mathscr{A} = \mathscr{A}_{old}$ then
14:             $finished \leftarrow true$
15:         else
16:             $\mathscr{A}_{old} \leftarrow \mathscr{A}$
17:     DEF $\leftarrow$ DEF-GENERATE($\mathscr{T}, \mathscr{A}, \preceq$)
18:     ($\mathscr{A}$, DEF) $\leftarrow$ AS-FILTER($\mathscr{A}$, DEF)
19:     return ($\mathscr{A}$, DEF)
```

Figure 1. Algorithms to generate $\mathscr{A}_l$ (left) and $\mathbb{AS}^l_{\mathscr{T}}$ (right)

ASG: This algorithm generates $\mathbb{AS}^l_{\mathscr{T}}$. It starts by generating $\mathscr{A}_l$, after which it successively adds sub-arguments and attacking arguments to the existing ones (lines 10-12). Once the arguments have been generated, `DEF-generate` computes the defeat relation — as described in Section 2 — by comparing pairs of arguments. As attacking arguments are not always defeaters (depending on the $\preceq$ relation chosen), we have to filter unnecessary arguments that do not fit the conditions of Definition 6. This is done by using calling the `AS-filter` (line 18) to perform the filtering.

5. Evaluation

Following Dung [8], let $\mathbb{AS} = (\mathscr{A}, \textsc{Def})$ be an AF and $\varepsilon \subseteq \mathscr{A}$. We say that ε is *conflict-free* iff there is no $a, b \in \varepsilon$ s.t. $(a, b) \in \textsc{Def}$. ε *defends* a iff for every $b \in \mathscr{A}$ s.t. $(b, a) \in \textsc{Def}$, there exists $c \in \varepsilon$ s.t. $(c, b) \in \textsc{Def}$. ε is *admissible* iff it is conflict-free and defends all its arguments. ε is a *complete extension* iff ε is admissible and contains all the arguments it defends. ε is a *preferred extension* iff it is a maximal (w.r.t. $\subseteq$) admissible set. ε is the *grounded extension* iff ε is a minimal (w.r.t. $\subseteq$) complete extension. We denote by $Ext(x, y)$ the function that returns the set of extensions of the AS x w.r.t. y, where $y \in \{pr, gr\}$ and pr (resp. gr) stands for the preferred (resp. grounded) semantics. Likewise, $Acc(x, y)$ is returns the accepted arguments of the AS x w.r.t. the semantics y.

Definition 7 (Status)**.** *Let $\mathbb{AS} = (\mathscr{A}, \textsc{Def})$ and $a \in \mathscr{A}$. a is* accepted *w.r.t. the preferred semantics (resp. grounded semantics) if $\forall E \in Ext(\mathbb{AS}, pr)$ (resp. $Ext(\mathbb{AS}, gr)$), $a \in E$. a is* rejected *w.r.t. the preferred semantics (resp. grounded semantics) if $\forall E \in Ext(\mathbb{AS}, pr)$ (resp. $Ext(\mathbb{AS}, gr)$), $a \notin E$ and a is* undecided *if it is neither accepted nor rejected.*

Definition 8 (Acceptability of a literal)**.** *Let $l \in \mathscr{L}$, $\mathscr{T}$ and $\mathbb{AS} = (\mathscr{A}, \textsc{Def})$. l is accepted w.r.t. the preferred (resp. grounded) semantics and $\mathbb{AS}$ iff there exists an $a \in \mathscr{A}$ s.t. $Conc(a) = l$ and $a \in Acc(\mathbb{AS}, pr)$ (resp. $a \in Acc(\mathbb{AS}, gr)$). Otherwise, l is rejected.*

Example 6 (Cont'd). *$Ext(\mathbb{AS}_{\mathscr{T}}, pr) = Ext(\mathbb{AS}_{\mathscr{T}}, gr) = \{\{A_1, A_2, A_3, A_4, A_5, A_7\}\}$ and $Ext(\mathbb{AS}^{\neg b}_{\mathscr{T}}, pr) = \{\{A_1, A_2, A_4, A_5, A_7\}\}$. Since $A_7 \in Acc(\mathbb{AS}^{\neg b}_{\mathscr{T}}, pr)$, $\neg b \in Acc_{\mathscr{L}}(\mathbb{AS}^{\neg b}_{\mathscr{T}}, pr)$. Note that m is accepted w.r.t. preferred/grounded in $\mathbb{AS}_{\mathscr{T}}$ but rejected in $\mathbb{AS}^{\neg b}_{\mathscr{T}}$.*

Proposition 1. *Let $l \in \mathscr{L}$, $\mathbb{AS}_{\mathscr{T}}$ be the AS for $\mathscr{T}$ and $\mathbb{AS}^{l}_{\mathscr{T}}$ be the AS for l. It holds that $l \in Acc_{\mathscr{L}}(\mathbb{AS}^{l}_{\mathscr{T}}, y)$ iff $l \in Acc_{\mathscr{L}}(\mathbb{AS}_{\mathscr{T}}, y)$, where $y \in \{pr, gr\}$.*

We now show that (1) in specific DTs (characterised via a sufficient condition) the AS for a literal has strictly fewer arguments than the corresponding original AS, (2) the rules that are not activated are not taken into account when constructing the arguments of an AS and (3) the GRI can be used to filter a DT $\mathscr{T}$ prior to the generation of $\mathbb{AS}^{l}_{\mathscr{T}}$.

Proposition 2. *Let $\mathscr{T} = (\mathscr{S}, \mathscr{D})$, $l \in \mathscr{L}$, $\mathbb{AS}_{\mathscr{T}} = (\mathscr{A}, \text{DEF})$, $\mathbb{AS}^{l}_{\mathscr{T}} = (\mathscr{A}', \text{DEF}')$ be the AS for l. If there exists $r \in \mathscr{S} \cup \mathscr{D}$ such that:*

- *r is activated and not potentially necessary for l then $|\mathscr{A}'| < |\mathscr{A}|$.*
- *r is not activated then $\mathbb{AS}_{\mathscr{T}} = \mathbb{AS}_{\mathscr{T}'}$, where $\mathscr{T}' = (\mathscr{S} \setminus \{r\}, \mathscr{D} \setminus \{r\})$.*
- *r is not potentially necessary for l then $\mathbb{AS}^{l}_{\mathscr{T}} = \mathbb{AS}^{l}_{\mathscr{T}'}$ where $\mathbb{AS}^{l}_{\mathscr{T}'}$ is the AS for l w.r.t. $\mathscr{T}' = (\mathscr{S} \setminus \{r\}, \mathscr{D} \setminus \{r\})$.*

The third item of Proposition 2 shows that filtering DTs to only keep potentially necessary rules is possible when generating the AS for a literal. If $\mathscr{T}$ contains rules that are not potentially necessary for a literal, this filtering reduces the time taken to answer a query. Moreover, the GRI only has to be computed once, and can then be stored in memory and reused for multiple queries. The proposed framework is inspired from previous approaches based on DT pre-processing [18].

Empirical Evaluation

To test our approach, we use existing benchmarks and DTs to compare the effectiveness of reasoning using BC and FC in the context of argumentation. To this end, we use existing DTs [12] as we are not aware of other standard benchmarks for instantiated argumentation. Due to space constraints, we only considered four theories from that work (tree, level, levels and teams). In **tree(n,k)**, the rules form a k-branching tree of depth n where the literal p_0 is the root. In these theories, every literal occurs only once. In **level(n)**, there is a cascade of n disputed conclusions, i.e. there are rules $\Rightarrow p_i$ and $p_{i+1} \Rightarrow \neg p_i$, for $0 \leq i \leq n$. In **levels(n)**, for odd i, the latter rule has a superior strength when compared to even rules. Finally, in **teams(n)**, every literal is disputed with two rules for p_i and two rules for $\neg p_i$, and the rules for p_i are superior to the rules for $\neg p_i$. To obtain the $\preceq$, we use the last-link principle described in [14].

For the FC procedure, we generated all arguments (5 times) using a breadth-first naive approach. For the BC procedure, for all theories, we randomly selected ten literals and generated the ASs for those literals. An upper limit of 200 minutes was set for all runs. Table 1 is split in three parts: FC, BC and DT filtration (by removing non-potentially necessary rules). In the *Forward* columns, we depict the mean time, the number of arguments generated, and the number of defeats of the graph[4]. In the *Backward* columns, we show, across all literals on non-timed out instances, the mean time for the generation

[4] Time does not include defeat generation, their number is calculated based on the structure of the theory.

Theory	Forward			Backward			Filter		
	Mean time (s)	# args.	# defeats	Mean time (s)	Mean # arguments	# Succ. instances	Mean # rules	% Filtration	Mean time (ms)
tree(n,k)									
$n = 8, k = 3$	5712.7	59049	0	19.6	15.8	10	11	99.93%	484.6
$n = 9, k = 3$	Timeout	196830	0	53.5	3.5	10	3.5	99.99%	810.5
level(n)									
$n = 10$	0.1	19	26	0.03	10.8	10	10.8	43.16%	41
$n = 1000$	6.7	1999	2996	830.5	716.6	10	716.6	64.15%	132.60
$n = 5000$	181.7	9999	14996	3927.2	602.3	3	3386	66.14%	302.30
$n = 10000$	678.9	19999	29996	Timeout	-	0	8423	57.88%	490.40
levels(n)									
$n = 10$	0.1	19	18	0.04	14	10	14	26.32%	46.5
$n = 1000$	6.7	1999	1998	1245.3	841.2	10	841.2	57.92%	160.9
$n = 5000$	155.9	9999	9998	96542.02	3702	2	5804.4	41.95%	451.6
$n = 10000$	696.8	19999	19998	428.5	163	1	10798.2	46.0%	555
teams(n)									
$n = 3$	0.4	176	272	0.26	3.1	10	3.1	97.89%	88
$n = 4$	1.6	736	1568	1.62	19.9	10	17.1	97.1%	131.2
$n = 5$	26.8	3008	8256	5.28	23.8	10	20.6	99.14%	198.3
$n = 6$	539.7	12160	254335	18.35	3.8	10	3.8	99.96%	369.2
$n = 7$	11613.2	48896	1401159	84.07	14.1	10	12.9	99.97%	866.5

Table 1. Summary of the empirical evaluation

of arguments, the mean number of arguments in the AS for the literal and the number of successful (non -timeout) instances. In the *Filter* columns, we show the number of rules after the filtration, the percentage of the number of rules filtered, and the mean time used for obtaining the filtered DT. We make three important observations: **(1)** For the tree and teams DTs, all the runs were successful and the BC was significantly faster in generating the arguments than the FC. It is worth noting that while the FC procedure times out after $n = 9$, the BC procedure is able to provide an answer in less than 5 minutes. (2) For the level DTs, the BC takes longer than the FC (even if it generates fewer arguments). From $n = 5000$ onward, most instances timeout. Note that the BC takes longer than the FC for these instances because it checks and generates all the arguments that can potentially attack the existing arguments. (3) In the tree and teams DTs, we obtain fewer arguments with the BC compared to the FC. The gap in the number of arguments means that the process of verification does not cause a serious overhead in the computation time.

6. Discussion and Future Work

We introduced the notion of BC argumentation and illustrated our approach with an ASPIC-style structured AS. We analysed the links between the AS generated using the BC procedure and the FC procedure w.r.t. argumentation semantics and showed an empirical comparison of the time needed to generate the arguments for both procedures.

Our work is motivated by the need for efficient query answering frameworks that do not need to generate the whole set of arguments [19, 18]. Our work relates with existing BC-based works such as DeLP [10] or ABA [16]. However, our focus is explicitly on ASPIC-like systems. There are also similarities between BC and proof dialogues [13], though most such dialogues operate on abstract ASs.

We have identified several potential avenues of future work. First, we intend to create additional benchmarks for instantiated ASs by replicating the properties of existing DTs [11]. Second, we recognise that there are similarities between the process we use, and different search algorithms. We intend to evaluate these different strategies, as well as heuristics for guiding the expansion process, and their effects on performance. Finally, integrating lifting rules for preferences (e.g. weakest link, elitist or democratic orderings [14]) could provide optimisations regarding argument expansion.

References

[1] O. Arieli, A. Borg, and C. Straßer. Prioritized Sequent-Based Argumentation. In *AAMAS 2018*, pages 1105–1113, 2018.

[2] J.-F. Baget, F. Garreau, M.-L. Mugnier, and S. Rocher. Extending Acyclicity Notions for Existential Rules. In *ECAI-14*, pages 39–44, 2014.

[3] P. Besnard and A. Hunter. *Elements of Argumentation.* MIT Press, 2008.

[4] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[5] M. Caminada, S. Modgil, and N. Oren. Preferences and unrestricted rebut. In *Proc. COMMA*, pages 209–220, 2014.

[6] M. Caminada and B. Verheij. On the existence of semi-stable extensions. *argumentation*, 3:4, 2010.

[7] M. Croitoru and S. Vesic. What Can Argumentation Do for Inconsistent Ontology Query Answering? In *SUM 2013*, pages 15–29, 2013.

[8] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.*, 77(2):321–358, 1995.

[9] P. E. Dunne and M. Wooldridge. Complexity of Abstract Argumentation. In *Argumentation in Artificial Intelligence*, pages 85–104. 2009.

[10] A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138, 2004.

[11] B. Konat, J. Lawrence, J. Park, K. Budzynska, and C. Reed. A Corpus of Argument Networks: Using Graph Properties to Analyse Divisive Issues. In *Proc. 10th Int'l Conf. on Language Resources and Evaluation*, 2016.

[12] M. J. Maher, A. Rock, G. Antoniou, D. Billington, and T. Miller. Efficient Defeasible Reasoning Systems. *Int. J. on A.I. Tools*, 10(4):483–501, 2001.

[13] S. Modgil and M. Caminada. Proof Theories and Algorithms for Abstract Argumentation Frameworks. In *Argumentation in Artificial Intelligence*, pages 105–129. 2009.

[14] S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[15] E. Salvat and M.-L. Mugnier. Sound and Complete Forward and backward Chaining of Graph Rules. In *ICCS 1996*, pages 248–262, 1996.

[16] F. Toni. A tutorial on assumption-based argumentation. *Argument & Computation*, 5(1):89–117, 2014.

[17] B. Yun, M. Croitoru, S. Vesic, and P. Bisquert. DAGGER: Datalog+/- Argumentation Graph GEneRator. In *AAMAS 2018*, pages 1841–1843, 2018.

[18] B. Yun, S. Vesic, and M. Croitoru. Toward a More Efficient Generation of Structured Argumentation Graphs. In *COMMA 2018*, 2018.

[19] B. Yun, S. Vesic, M. Croitoru, P. Bisquert, and R. Thomopoulos. A Structural Benchmark for Logical Argumentation Frameworks. In *IDA*, pages 334–346, 2017.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200526

Sets of Attacking Arguments for Inconsistent Datalog Knowledge Bases

Bruno YUN [a,1], Srdjan VESIC [b] and Madalina CROITORU [c]
[a] *University of Aberdeen, United Kingdom*
[b] *CRIL - CNRS & Univ. Artois, France*
[c] *University of Montpellier, France*

Abstract. Logic-based argumentation is a well-known approach for reasoning with inconsistent logic knowledge bases. Such frameworks have been shown to suffer from a major practical drawback consisting of a large number of arguments and attacks. To address this issue, we provide an argumentation framework that considers sets of attacking arguments and provide a theoretical analysis of the new framework with respect to its syntactic and semantic properties. We provide a tool for generating such argumentation frameworks from a Datalog knowledge base and study their characteristics.

Keywords. argumentation, datalog, SETAF

1. Introduction

In this paper, we place ourselves in the setting of logic-based argumentation instantiated over Datalog. The use of this language ensures that the work of this paper studies potentially real world argumentation graphs and unveils genuine structural behaviour. Logic-based argumentation is a well known approach for reasoning with inconsistent logic knowledge bases (KB). While its strength, in the instantiated case, might not lie in its reasoning efficiency, particularly when compared to other inconsistent tolerant reasoning methods such as ASP [23] or dedicated tools [12]. Its added value is two fold. First, its explanatory power benefits to increase the scrutability of the system by users [3,8]. Second, the use of ranking semantics can induce a stratification of the inconsistent KB [2] that might be of use for query answering techniques [27].

Starting from an inconsistent KB (composed of a set of factual knowledge and an ontology stating positive and negative rules about the factual knowledge), one can attempt to generate the arguments and the attacks corresponding to the KB using existing logic-based AFs: Deductive argumentation [9], ASPIC+ [20], Assumption-Based Argumentation (ABA) [24,11] or DeLP [17]. However, none of these argumentation frameworks (AF) are straightforwardly applicable in the context of Datalog. Indeed, the aforementioned frameworks are not usable without adding or removing any rules or facts in the KB. Let us now illustrate this statement. In the case of ASPIC+, we cannot instantiate it because the definition of the contrariness relation is not general enough to account

[1] Corresponding Author: University of Aberdeen, United Kingdom; E-mail: bruno.yun@ed.ac.uk

for negative constraints. Let us show this on an example. Suppose we are given three facts about the shape and taste of a biscuit: (f_1) "the biscuit has a square shape", (f_2) "the biscuit has a round shape" and (f_3) "the biscuit is sweet". Now, further suppose that there are no rules and one negative constraint: the biscuit cannot have a square and round shape at the same time. As the fact f_3 is a *free-fact* (i.e. it is not involved in any minimal conflict), there is no way to define its contrary intuitively without modifying the set of rules of the KB. Namely, the third item of *Definition 5.1* in the work of [20] specifies that each formula of the language must have at least one contradictory, which is not the case for the latter fact in our example. Of course, it is possible to declare that a fact "the biscuit is not sweet" is the contradictory of f_3. However, for each contradictory, the corresponding negative constraint has to be added to the KB. In the case of ABA, although it is abstract enough to function with a language that has neither implication nor negation, it needs a contrariness function that returns a single contrary sentence for each formula of the language. This is not enough in the case where a fact appears in multiple conflicts and the language does not allow for the disjunction. In the case of DeLP, we cannot instantiate it since the original work only consider ground rules.

Specifically crafted instantiations for Datalog, such as the instantiations of Croitoru and Vesic [14], Yun et al. [28] and Arioua et al. [4], have been proven to respect the argumentation rationality desiderata [1,13] and to output a set of extensions equivalent to the set of repairs [19,10] of the KB (i.e. the maximum consistent sets of facts w.r.t. inclusion). Unfortunately, it was shown that these instantiations suffer from a major drawback: a large number of arguments and attacks [25]. This problem even occurs in the case where there are no rules in the KB (for instance, a graph with 13 arguments and 30 attacks can be generated with a meagre KB with solely 4 facts, no positive rules and a single negative rule). As a consequence, the argumentation graph for a "normally-sized" KB cannot be held in main memory, requires dedicated large-graph visualisation tools, and, despite their polynomial complexity regarding the number of arguments, still poses combinatorial challenges for the computation of ranking techniques. The question that arose is whether or not we can find more efficient AFs for Datalog. To this end, we provide an AF that considers *sets of attacking arguments* (n-ary attacks) [22,21,16] and possesses arguments that are built upon other arguments (*à la* ASPIC+) and n-ary attacks. We show that this new framework retains desirable properties with fewer arguments and attacks compared to the existing frameworks.

There are three main contributions in this paper. First, we introduce a logic-based AF with n-ary attacks for an inconsistent KB expressed using Datalog. Second, we provide a theoretical analysis of the new AF w.r.t. its syntactic and semantic properties. Last, we provide a tool for generating this AF from a KB expressed in Datalog Plus (DLGP) format and study its performance in terms of argumentation graph compression rate and generation time.

The structure of the paper is as follows. In Section 2, we recall the necessary definitions of Datalog and the AF of [28] and [4]. In Section 3, we introduce a new AF and study its theoretical properties. In Section 4, we empirically compare the two frameworks w.r.t. the number of arguments and attacks on a set of KBs.

2. Background

We start by introducing the Datalog language[2]. It is composed of formulae built with the usual quantifier ($\forall$) and *only* two connectors: implication ($\rightarrow$) and conjunction ($\wedge$) and is composed of facts, rules and negative constraints. A *fact* is a ground atom of the form $p(t_1,\dots,t_k)$ where p is a predicate of arity k and t_i, with $i \in [1,\dots,k]$, constants. A *(positive) rule* r is of the form $\forall \overrightarrow{X}, \overrightarrow{Y}\ B_r[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow H_r[\overrightarrow{Y}]$ where B_r and H_r are closed atoms or conjunctions of closed atoms, respectively called the body and the head of r, and $\overrightarrow{X}, \overrightarrow{Y}$ their respective vectors of variables. For simplicity purposes, we will consider that the rules only have one atom in the head. This is not a big assumption as it has been proved that an arbitrary set of rules can be transformed into a set of rules with atomic head; see the work of [7] for more details. However, note that the results of this paper can be extended to the case where rule heads are not atomic.

Let X be a set of variables and T be a set of terms (constants or variables). A substitution of X to T is a function from X to T. A homomorphism π from a set of atoms S to a set of atoms S' is a substitution of the variables of S with the terms of S' s.t. $\pi(S) \subseteq S'$. A *rule is applicable* to a set of facts $\mathscr{F}$ iff there exists a homomorphism [5] from its body to $\mathscr{F}$. Applying a rule to a set of facts (also called *chase*) consists of adding the set of atoms of its head to the facts according to the application homomorphism. A *negative constraint* is a rule r of the form $\forall \overrightarrow{X} B_r[\overrightarrow{X}] \rightarrow \perp$ where B_r is a closed atom or conjunctions of closed atoms, $\overrightarrow{X}$ the respective vector of variables and $\perp$ is *absurdum*.

Definition 1 (Knowledge base). *A KB $\mathscr{K}$ is a tuple $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ where $\mathscr{F}$ is a finite set of facts, $\mathscr{R}$ a set of positive rules and $\mathscr{N}$ a set of negative constraints.*

Example 1. *Suppose that one is indecisive about what to eat for an appetiser. He decides that the dish should contain salted cucumbers, sugar, yogurt, not be a soup and be edible. However, he finds out that combining together salted cucumbers, sugar and yogurt may not be a good idea. Furthermore, combining salted cucumbers with yogurt is a dish called "tzaziki" which is a famous greek soup. We model the situation with the KB $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, where:*

- $\mathscr{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
- $\mathscr{R} = \{\forall x(contains(x, saltC) \wedge contains(x, yogurt) \rightarrow tzaziki(x))\}$
- $\mathscr{N} = \{\forall x(contains(x, saltC) \wedge contains(x, sugar) \wedge contains(x, yogurt) \rightarrow \perp), \forall x(tzaziki(x) \wedge notSoup(x) \rightarrow \perp)\}$

In the Ontology Based Data Access (OBDA) setting, rules and constraints are used to "access" different data sources. These sources are prone to inconsistencies. We assume that the rules of the KB are compatible with the negative constraints, i.e. the union of those two sets is satisfiable [19]. Indeed, the ontology is believed to be reliable as it is the result of a robust construction by domain experts. However, as data can be heterogeneous due to merging and fusion, the data is assumed to be the source of inconsistency.

[2] For simplicity purposes we use the Datalog language but this work can be easily extended to the Datalog± formalism if we restrict ourselves to the class of FES rules.

The *saturation* of a set of facts $\mathscr{F}$ by $\mathscr{R}$ is the set of all possible atoms and conjunctions of atoms that are entailed, after using all rule applications from $\mathscr{R}$ over $\mathscr{F}$ until a fixed point. The output of this process is called the closure and is denoted by $\mathbb{SAT}_{\mathscr{R}}(\mathscr{F})$. A set $\mathscr{F}$ is said to be *$\mathscr{R}$-consistent* if no negative constraint hypothesis can be entailed, i.e. $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(\mathscr{F}) \not\models \bot$. Otherwise, $\mathscr{F}$ is said to be *$\mathscr{R}$-inconsistent*. We introduce the notion of repair (maximal consistent subset) and free-fact.

Definition 2 (Repair)**.** *A repair of $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ is $X \subseteq \mathscr{F}$ s.t. X is $\mathscr{R}$-consistent and there exists no X' s.t. $X \subset X'$ and X' is $\mathscr{R}$-consistent. The set of all repairs of a KB $\mathscr{K}$ is denoted by $Repair(\mathscr{K})$.*

Definition 3 (Free-fact)**.** *Let $\mathscr{K}$ be a KB, a fact $f \in \mathscr{F}$ is a free-fact iff for every repair $R \in Repair(\mathscr{K}), f \in R$.*

Example 2 (Cont'd Example 1)**.** *In our example, there are three repairs, each representing one alternative: yogurt with sugar (which is common), tzaziki or sugar with salter cucumbers (sweet pickles). Namely, we have that $Repair(\mathscr{K}) = \{R_1, R_2, R_3\}$, where:*

- $R_1 = \{contains(m, saltC), contains(m, yogurt), edible(m)\}$,
- $R_2 = \{contains(m, sugar), contains(m, saltC), notSoup(m), edible(m)\}$,
- $R_3 = \{contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$.

Here, $edible(m)$ is a free-fact.

We now recall the AF provided by [28] and [4] and based on the original framework of [14]. This AF has deductive arguments and an asymmetric attack relation based on the notion of undermining. An argument a attacks an argument b if the conclusion of the argument a is incompatible with one element of the hypothesis of the argument b.

Definition 4 (Argumentation framework $\mathfrak{AS}'$)**.** *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB. The corresponding AF, denoted by $\mathfrak{AS}'_{\mathscr{K}}$, is the pair $(\mathscr{A}', \mathscr{C}')$ with $\mathscr{C}' \subseteq \mathscr{A}' \times \mathscr{A}'$ such that:*

- *An argument $a' \in \mathscr{A}'$ is a tuple (H, C) with H a non-empty $\mathscr{R}$-consistent subset of $\mathscr{F}$ and C a set of facts s.t. (1) $C \subseteq \mathbb{SAT}_{\mathscr{R}}(H)$ and (2) there is no $H' \subset H$ s.t. $C \subseteq \mathbb{SAT}_{\mathscr{R}}(H')$. The support H of an argument a' is denoted by $Supp(a')$ and the conclusion C by $Conc(a')$.*
- *a' attacks b', denoted by $(a', b') \in \mathscr{C}'$, iff there exists $\varphi \in Supp(b')$ s.t. $Conc(a') \cup \{\varphi\}$ is $\mathscr{R}$-inconsistent.*

Example 3 (Cont'd Example 1)**.** *The AF $\mathfrak{AS}'_{\mathscr{K}}$ has 33 arguments and 360 attacks. Moreover, a'_1 defined by $(\{contains(m, saltC), contains(m, yogurt)\}, \{tzaziki(m)\})$ attacks argument a'_2 defined by $(\{notSoup(m)\}, \{notSoup(m)\})$ but a'_2 does not attack a'_1 because $\{notSoup(m)\}$ is not $\mathscr{R}$-inconsistent with either the atom $contains(m, saltC)$ or $contains(m, yogurt)$.*

The AF $\mathfrak{AS}'_{\mathscr{K}}$ generated from a KB $\mathscr{K}$, has been proven to possess good properties such as the equivalence between the set of repairs and the set of preferred (resp. stable) extensions, the desirable postulates and the equivalence results for query answering in the OBDA field [14].

However, one of the main drawbacks of this method is the huge number of arguments. Indeed, [25] proved that the number of arguments is exponential w.r.t. the number of free-facts. Moreover, it was empirically shown that for a KB with eight facts, six rules and two ternary negative constraints, we might generate 11,007 arguments and 23,855,104 attacks [28].

3. New Argumentation framework $\mathfrak{AS}$

In this section, we show a novel AF for generating arguments and attacks from an inconsistent KB. We also show that this AF possesses all of the desirable properties of $\mathfrak{AS}'_{\mathscr{K}}$.

Note that although the framework described in this section has some similarities with the ASPIC+ framework, the ASPIC+ cannot be directly instantiated with Datalog because the language does not have the negation and the contrariness function is not general enough for this language. Moreover, when instantiating ASPIC+, one usually has to add all the tautologies of the language in the set of rules to guarantee that the result will be consistent, i.e. to satisfy the rationality postulates defined by [13]. To avoid adding this huge number of rules and also with the goal of decreasing the number of arguments, we propose not to add them. However, the cost of forgetting to add those rules (and the arguments generated using them) would result in a violation of rationality postulates. We propose to solve this problem in a more elegant way. Namely, we allow for the use of sets of attacking arguments (i.e. n-ary attacks).

Definition 5 (Argumentation framework $\mathfrak{AS}$)**.** *Let us consider the KB $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$. The corresponding AF, denoted by $\mathfrak{AS}_{\mathscr{K}}$, is the pair $(\mathscr{A}, \mathscr{C})$ with $\mathscr{C} \subseteq (2^{\mathscr{A}} \setminus \{\emptyset\}) \times \mathscr{A}$ such that:*

- *An argument $a \in \mathscr{A}$ is either **(1)** a fact f, where $f \in \mathscr{F}$ s.t. $Conc(a) = f$ and $Prem(a) = \{f\}$ or **(2)** $a_1, \ldots, a_n \rightarrow f'$ where $a_1, \ldots, a_n \in \mathscr{A}$ s.t. there exists a tuple (r, π) where $r \in \mathscr{R}, \pi$ is a homomorphism from the body of r to $\{Conc(a_1), \ldots, Conc(a_n)\}$ and f' is the resulting atom from the rule application. $Conc(a) = f'$ and $Prem(a) = Prem(a_1) \cup \cdots \cup Prem(a_n)$. Note that in both cases, $Prem(a)$ must be $\mathscr{R}$-consistent.*
- *An attack in $\mathscr{C}$ is a pair (X, a) s.t. X is minimal for set inclusion s.t. $\bigcup_{x \in X} Prem(x)$ is $\mathscr{R}$-consistent and there exists $\varphi \in Prem(a)$ s.t. $(\bigcup_{x \in X} Conc(x)) \cup \{\varphi\}$ is $\mathscr{R}$-inconsistent.*

With a slight abuse of notation, we also use the notation $Conc(a)$ to refer to the conclusion of an argument in $\mathfrak{AS}$. However, the conclusion is not a set anymore (see Definition 4). The reason is that $\mathscr{K}$ can be processed w.l.o.g. to contain only rules with atomic head [7].

Notation: Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $X \subseteq \mathscr{F}$ be a set of facts and $X' \subseteq \mathscr{A}$ be a set of arguments of $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$. We define the set of arguments generated by X as $Arg(X) = \{a \in \mathscr{A} \mid Prem(a) \subseteq X\}$ and the base of a set of arguments X' as $Base(X') = \bigcup_{x' \in X'} Prem(x')$. We define $Concs(X') = \bigcup_{x' \in X'} Conc(x')$.

In case of binary attacks [15], a set of arguments X is said to attack an argument a iff there exists $b \in X$ s.t. b attacks a. We need a similar notion here except that we already have a notion of attack from a set towards an argument. In order not to mix up the two notions, we introduce the notation $\mathscr{C}^*$, which stands for the saturated set of attacks. For example, if $(\{a,b\},c) \in \mathscr{C}$ then each set X' containing a and b (i.e. s.t. $\{a,b\} \subseteq X'$) attacks c too (i.e. $(X',c) \in \mathscr{C}^*$).

Definition 6 (Saturated set of attacks)**.** *Let $\mathfrak{AS} = (\mathscr{A},\mathscr{C})$ be an AF. The saturated set of attacks of $\mathfrak{AS}$ is $\mathscr{C}^* = \{(X,a) \mid$ there exists $(X',a) \in \mathscr{C}$ with $X' \subseteq X \subseteq \mathscr{A}\}$.*

Example 4 (Cont'd Example 1)**.** *The argumentation graph $\mathfrak{AS}_{\mathscr{K}}$ is composed of the six following arguments and 11 attacks: $a_1 = contains(m,sugar)$, $a_2 = contains(m,saltC)$, $a_3 = contains(m,yogurt)$, $a_4 = notSoup(m)$, $a_5 = edible(m)$ and $a_6 = a_2,a_3 \rightarrow tzaziki(m)$. An example attack of $\mathscr{C}$ is $(\{a_1,a_2\},a_3)$.*

From $\mathscr{K}$, one can build an AF $\mathfrak{AS}_{\mathscr{K}}$ with sets of attacking arguments (see Definition 5). Please note that although the work of Yun et al. [26] seems similar, it is based on building all arguments using Definition 4, filtering specific arguments and filling up the loss of information induced by the missing arguments with sets of attacking arguments to keep the rationality postulates. Let us illustrate the difference between the framework of Yun et al. [26] and the new AF on a KB with 3 facts and a single negative constraint on those three facts. In the framework of Yun et al., there will be six arguments and nine attacks whereas there are three arguments and three attacks in the new AF.

3.1. Argumentation framework properties of $\mathfrak{AS}_{\mathscr{K}}$

The AF $\mathfrak{AS}$ is an instantiation of the abstract SETAF framework proposed by Nielsen and Parsons [21,22]. For the purpose of the paper being self-contained, we recall the necessary definitions.

Definition 7 (Argumentation semantics)**.** *Let $\mathfrak{AS} = (\mathscr{A},\mathscr{C})$, $\mathscr{C}^*$ the corresponding saturated set of attacks and $S_1,S_2 \subseteq \mathscr{A}$. We say that: S_1 is* ***conflict-free*** *iff there is no argument $a \in S_1$ s.t. $(S_1,a) \in \mathscr{C}^*$. S_1 attacks S_2 iff there exists $a \in S_2$ s.t. $(S_1,a) \in \mathscr{C}^*$[3]. S_1 defends an argument a iff for every $S_2 \subseteq \mathscr{A}$ s.t. $(S_2,a) \in \mathscr{C}$, we have that $(S_1,S_2) \in \mathscr{C}^*$. S_1 is said to be* ***admissible*** *if each argument in S_1 is defended by S_1. An admissible set S_1 is called a* ***preferred extension*** *if there is no admissible set $S_2 \subseteq \mathscr{A}$, $S_1 \subset S_2$. A conflict-free set S_1 is a* ***stable extension*** *if S_1 attacks all arguments in $\mathscr{A} \setminus S_1$. An admissible set S_1 is called a* ***grounded extension*** *if S_1 is minimum (w.r.t. $\subseteq$) s.t. it contains every argument defended by S_1.*

The set of all preferred (resp. stable and grounded) extensions of an AF $\mathfrak{AS}$ is denoted by $Ext_p(\mathfrak{AS})$ (resp. $Ext_s(\mathfrak{AS})$ and $Ext_g(\mathfrak{AS})$). The output of an AF for an argumentation semantics is $Output_x(\mathfrak{AS}_{\mathscr{K}}) = \bigcap_{E \in Ext_x(\mathfrak{AS}_{\mathscr{K}})} Concs(E)$ where $x \in \{s,p,g\}$.

Example 5 (Cont'd Example 4)**.** *The preferred (resp. stable) extensions of $Ext_p(\mathfrak{AS}_{\mathscr{K}})$ (resp. $Ext_s(\mathfrak{AS}_{\mathscr{K}})$) are $E_1 = \{a_2,a_3,a_5,a_6\}$, $E_2 = \{a_1,a_2,a_4,a_5\}$ and $E_3 = \{a_1,a_3,a_4,a_5\}$. The grounded extension is $E_{GE} = \{a_5\}$*

[3] By abuse of notation, we will use the notation $(S_1,S_2) \in \mathscr{C}^*$ for the case when S_1 attacks a set of arguments S_2.

We now show that there is a correspondence between the set of preferred (resp. stable) extensions and the set of repairs.

Proposition 1 (Preferred & Stable Characterisation)**.** *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $x \in \{s,p\}$. Then, $Ext_x(\mathfrak{AS}_{\mathscr{K}}) = \{Arg(A') \mid A' \in Repair(\mathscr{K})\}$*

Example 6 (Cont'd Example 5)**.** *As explained in Proposition 1, we have a correspondence between repairs and preferred (resp. stable) extensions. Hence:*

- $E_1 = Arg(\{contains(m,saltC), contains(m,yogurt), edible(m)\})$,
- $E_2 = Arg(\{contains(m,sugar), contains(m,saltC), notSoup(m), edible(m)\})$
- $E_3 = Arg(\{contains(m,sugar), contains(m,yogurt), notSoup(m), edible(m)\})$.

Next, we show the equivalence between the non-attacked arguments and the arguments generated from free-facts.

Corollary 1 (Non-attacked characterisation)**.** *Let $\mathscr{K}$ be a KB, $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ and $a \in \mathscr{A}$. There exists no S s.t. $(S,a) \in \mathscr{C}$ iff $Prem(a) \subseteq \bigcap\limits_{R \in Repair(\mathscr{K})} R$.*

Note that although it is tempting to say that the non-attacked arguments do not contribute to attacks because they are based on free-facts, this is not true in the general case. In the next proposition, we show that the grounded extension is equal to the intersection of the preferred extensions. Note that the grounded extension is always included in the intersection of the preferred extensions in the general case.

Proposition 2 (Grounded & Preferred)**.** *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $Ext_g(\mathfrak{AS}_{\mathscr{K}}) = \{E_{GE}\}$. Then $E_{GE} = \bigcap\limits_{E \in Ext_p(\mathfrak{AS}_{\mathscr{K}})} E$*

We show the equality between the grounded extension and arguments generated by the intersection of all the repairs.

Proposition 3 (Grounded Characterisation)**.** *Let $\mathfrak{AS}_{\mathscr{K}}$ be an AF and $Ext_g(\mathfrak{AS}_{\mathscr{K}}) = \{E_{GE}\}$. Then $E_{GE} = Arg(\bigcap\limits_{R \in Repair(\mathscr{K})} R)$.*

Example 7 (Cont'd Example 5)**.** *We have that the grounded extension $E_{GE} = E_1 \cap E_2 \cap E_3 = \{a_5\}$ and that the grounded extension is $E_{GE} = \{a_5\} = Arg(\{edible(m)\})$.*

We now show that for any arbitrary KB $\mathscr{K}$, the generated AF $\mathfrak{AS}_{\mathscr{K}}$ does not contain self-attacking arguments.

Proposition 4 (Self-attacking Arguments)**.** *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. There is no $(S,t) \in \mathscr{C}$ s.t. $t \in S$.*

In Proposition 5 below, we show that an attacked argument is always defended by a set of arguments.

Proposition 5 (Defense)**.** *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. If there is $(S,t) \in \mathscr{C}$ then there exists $(S',s) \in \mathscr{C}$ s.t. $s \in S$.*

We introduce the definition of cycle for our AF.

Definition 8 (Cycle). *A cycle in $\mathfrak{AS} = (\mathscr{A}, \mathscr{C})$ is a sequence of attacks in $\mathscr{C}$ of the form $((S_1,t_1),\ldots,(S_n,t_n))$ s.t. for every $i \in \{1,\ldots,n-1\}, t_i \in S_{i+1}$ and $t_n \in S_1$.*

The following corollary follows directly from Proposition 5 and shows that if the number of arguments is finite then there exists at least one cycle in the framework.

Corollary 2 (Cycle Existence). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$ be an AF. If $|\mathscr{A}|$ is finite and non empty and $\mathscr{C} \neq \emptyset$ then there exists a cycle in $\mathfrak{AS}_{\mathscr{K}}$.*

Example 8 (Cont'd Example 4). *The sequence of attacks $((\{a_2,a_3\},a_1),(\{a_1,a_3\},a_2))$ is a cycle in $\mathfrak{AS}_{\mathscr{K}}$.*

Contrary to the AF described in Definition 4 where the number of arguments can be exponential even in the case where the set of rules is empty, we show that in the framework described in Definition 5, the set of arguments is at most equal to the number of facts.

Observation 1 (Argument upper-bound). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ s.t. $\mathscr{R} = \emptyset$ and $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$, then $|\mathscr{A}| \leq |\mathscr{F}|$.*

In the next proposition, we show an upper bound to the number of attacks w.r.t. the number of arguments.

Proposition 6 (Attack upper-bound). *Let $\mathfrak{AS}_{\mathscr{K}} = (\mathscr{A}, \mathscr{C})$. If $|\mathscr{A}| = n$ then $|\mathscr{C}| \leq n \times (2^{n-1} - 1)$.*

In the general case, this upper-bound on attacks is almost never reached because of the minimality condition on attacks.

3.2. Rationality postulates

In this section, we prove that the framework we propose in this paper satisfies the rationality postulates for instantiated AFs. We first prove the indirect consistency postulate.

Proposition 7 (Indirect consistency). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a KB, $\mathfrak{AS}_{\mathscr{K}}$ be the corresponding AF and $x \in \{s,p,g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathscr{K}})$, $Concs(E)$ is $\mathscr{R}$-consistent and $Output_x(\mathfrak{AS}_{\mathscr{K}})$ is $\mathscr{R}$-consistent.*

Proof. Let E be a stable or preferred extension of $\mathfrak{AS}_{\mathscr{K}}$. From Proposition 1, there exists a repair $A' \in Repair(\mathscr{K})$ s.t. $E = Arg(A')$. By definition, $Concs(E) = \mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(A')$. Formally, $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(A')) = \mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E))$. Since $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}$ is idempotent, this means that we have $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(A') = \mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E))$. Since it holds that $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(A') \not\models \bot$, then $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E)) \not\models \bot$ and $Concs(E)$ is $\mathscr{R}$-consistent.

Let us consider the case of grounded semantics. Denote E_{GE} the grounded extension of $\mathfrak{AS}_{\mathscr{K}}$. We just proved that for every $E \in Ext_p(\mathfrak{AS}_{\mathscr{K}})$, it holds that $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E)) \not\models \bot$. Since the grounded extension is a subset of the intersection of all the preferred extensions, and since there is at least one preferred extension [22], say E_1, then $E_{GE} \subseteq E_1$. Since $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E_1)) \not\models \bot$ then $\mathbb{SAT}_{\mathscr{R}\cup\mathscr{N}}(Concs(E_{GE})) \not\models \bot$ and $Concs(E_{GE})$ is $\mathscr{R}$-consistent.

Consider the case of stable or preferred semantics. We prove that $Output_x(\mathfrak{AS}_{\mathcal{K}})$ is $\mathcal{R}$-consistent. Recall that $Output_x(\mathfrak{AS}_{\mathcal{K}}) = \bigcap_{E \in Ext_x(\mathfrak{AS}_{\mathcal{K}})} Concs(E)$. Since every KB has at least one repair then, there is at least one stable or preferred extension E. From the definition of the output, $Output_x(\mathfrak{AS}_{\mathcal{K}}) \subseteq Concs(E)$. Since $Concs(E)$ is $\mathcal{R}$-consistent then $Output_x(\mathfrak{AS}_{\mathcal{K}})$ is $\mathcal{R}$-consistent. Note that since there is only one grounded extension, we get that $\mathbb{SAT}_{\mathcal{R}}(Output_g(\mathfrak{AS}_{\mathcal{K}})) = \mathbb{SAT}_{\mathcal{R}}(Concs(E_{GE}))$. □

Since our instantiation satisfies indirect consistency then it satisfies direct consistency. Indeed, if a set is $\mathcal{R}$-consistent, then it is consistent. Thus, we obtain the following corollary.

Corollary 3 (Direct consistency). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB, $\mathfrak{AS}_{\mathcal{K}}$ the corresponding AF and $x \in \{s,p,g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathcal{K}}), Concs(E) \not\models \bot$ and $Output_x(\mathfrak{AS}_{\mathcal{K}}) \not\models \bot$.*

Proposition 8 shows that the AF satisfies Closure.

Proposition 8 (Closure). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB, $\mathfrak{AS}_{\mathcal{K}}$ be the corresponding AF and $x \in \{s,p,g\}$. Then, for every $E \in Ext_x(\mathfrak{AS}_{\mathcal{K}}), Concs(E) = \mathbb{SAT}_{\mathcal{R}}(Concs(E))$ and $Output_x(\mathfrak{AS}_{\mathcal{K}}) = \mathbb{SAT}_{\mathcal{R}}(Output_x(\mathfrak{AS}_{\mathcal{K}}))$.*

	Existing Framework $\mathfrak{AS}'_{\mathcal{K}}$			New Framework $\mathfrak{AS}_{\mathcal{K}}$					
$\mathcal{K}$	# Arg.	# Att.	Gen. Time	# Arg.	% Arg. ↓	# Att.	% Att. ↓	Gen. Time	% Time ↓
A_1	22	128	160	5	77,27	6	93,75	276,00	-81,48
A_2	25	283	133	7	72,00	8	92,93	342,00	-183,57
A_3	85	1472	399,5	7	91,76	9	99,26	369,50	1,66
B	5967	11542272	533089	14	99.77	20.5	99.99	7814.5	98.08

Table 1. Comparison of the median number of arguments, attacks and generation time needed (in ms) between the two frameworks $\mathfrak{AS}_{\mathcal{K}}$ and $\mathfrak{AS}'_{\mathcal{K}}$ on the sets of KBs A_1, A_2, A_3 and B.

4. Empirical Analysis

We now compare our approach with the existing AF for Datalog w.r.t. the number of arguments and the number of attacks. All experiments were conducted on a Debian computer with an Intel Xeon E5-1620 processor and 64GBs of RAM. We chose to work with the set of KBs extracted from the study of [28,26]. These inconsistent KBs are composed of two main sets:

- A set A composed of 108 KBs. A is further split into three smaller sets of KBs: A set A_1 of 31 KBs without rules, two to seven facts, and one to three negative constraints, a set A_2 of 51 KBs generated by fixing the size of the set of facts and adding negative constraints until saturation and a set A_3 of 26 KBs with ternary negative constraints, three to four facts and one to three rules.
- A set B of 26 KBs with eight facts, six rules and one or two negative constraints. This set contains more free-facts than the KBs in set A.

For each of these two sets, we compare the number of arguments and attacks of the new framework defined in Definition 5 with the one of Definition 4. We provided a tool based on the Graph of Atom Dependency defined by [18] and the Graal Java Toolkit [6] for generating the new AF from an inconsistent KB expressed in the DLGP format. The tool is available online at: https://www.dropbox.com/sh/dlpmr07gqvpuc61/AABDgwfHJRNVYcsqpDg7kMfEa?dl=0

4.1. Experimental results

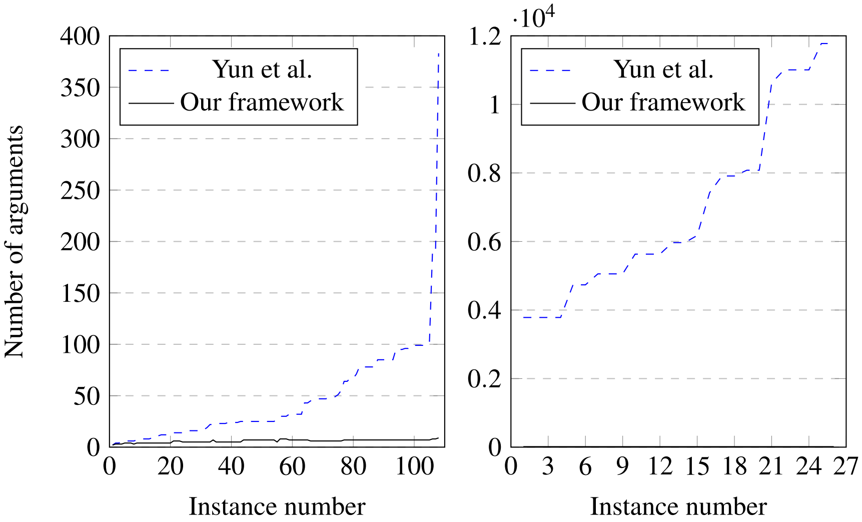


Figure 1. Comparison of the number of arguments between the two AFs on sets A (left) and B (right).

In Table 1, we show the number of arguments and attacks of the two frameworks $\mathfrak{AS}$ and $\mathfrak{AS}'$ for the two sets of KBs (A and B). We make the following observations: First, contrary to the framework $\mathfrak{AS}'$, there is no exponential increase in the number of arguments with the number of free-facts in $\mathfrak{AS}$ as seen with the KBs in set B. Moreover, for all the KBs considered in sets A and B, the number of arguments and attacks in $\mathfrak{AS}$ is less or equal to the number of arguments and attacks in $\mathfrak{AS}'$. We can notice that the efficiency brought by this new framework is obvious in the case where the KBs contain more free facts (see Figures 1). Second, when the set of facts and the set of rules are fixed and only the set of negative constraint is modified, the number of arguments of $\mathfrak{AS}$ seems to be unchanged whereas in $\mathfrak{AS}'$, it is varying. $\mathfrak{AS}'$ is also much denser than $\mathfrak{AS}$. Indeed, the median density[4] of $\mathfrak{AS}'$ is 26.34% and 31,03% whereas the median density of $\mathfrak{AS}$ is 4,69% and 0.02% for the set A and B respectively. Third, the generation of $\mathfrak{AS}$ is slower than the one for $\mathfrak{AS}'$ when the number of arguments and attacks is relatively low (see A_1, A_2 and Figure 2) but when the number of arguments and attacks increases, we can notice that the generation of $\mathfrak{AS}$ is much faster (see B and Figure 2).

[4] The density is equal to the number of attacks divided by the maximum number of possible attacks. In the case of a directed graph, the maximum number of attacks is given by $n(n-1)$ where n is the number of nodes. In the case of $\mathfrak{AS}$, we use the formula in Proposition 6 to obtain the maximum number of attacks.

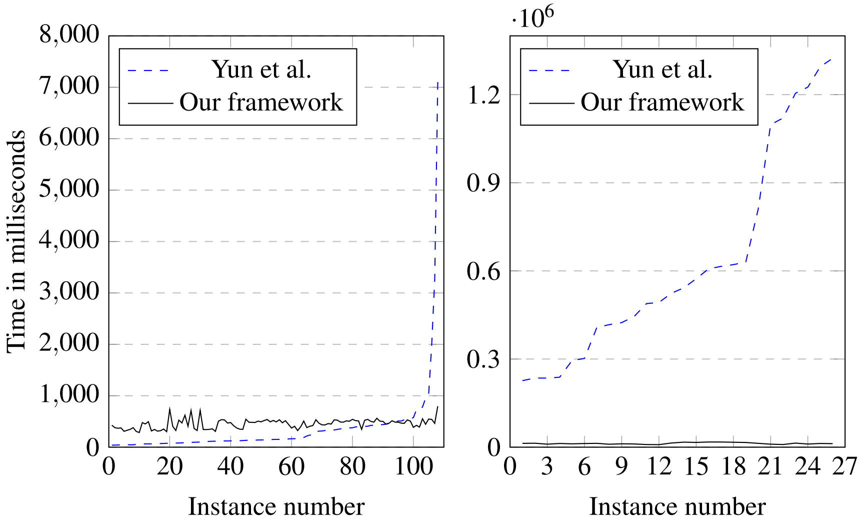


Figure 2. Generation time needed between the two AFs on sets A (left) and B (right).

5. Conclusion

We introduced a new logic-based AF with n-ary attacks that is built over an inconsistent Datalog KB and analysed the syntactic and semantic properties of this AF. We showed that it has all of the desirable properties of the existing AF for Datalog. Namely: (1) the rationality postulates for instantiated AFs defined by [13] are satisfied, (2) there is a bijection between the stable (resp. preferred) extensions and the sets of arguments generated from the repairs, (3) the grounded extension is equal to both the intersection of the preferred extensions and the set of arguments generated from free-facts, (4) the non-attacked arguments are generated from the free-facts and for each attacked argument there exists a set of arguments that defends it. (5) there are no self-attacking arguments and there is at least one cycle if the set of arguments is finite, (6) we give an upper-bound on the number of arguments and the number of attacks.

Second, we provided a tool for generating this n-ary AF from a knowledge base expressed in DLGP format and used it to conduct en empirical comparison between this n-ary framework and the existing AF [28,4] w.r.t. the number of arguments, attacks and time needed for the generation. We highlighted that this n-ary framework possesses fewer arguments and attacks than the existing framework mainly because it avoids the problem of the exponential increase of arguments when free-facts are added. Moreover, although the generation of the new framework is slower than the existing framework when the number of arguments and attacks is low, as soon as the number of arguments and attacks increases, the generation of n-ary framework is faster than for the existing framework.

References

[1] L. Amgoud. Postulates for logic-based argumentation systems. *Int. J. Approx. Reasoning*, 55(9):2028–2048, 2014.

[2] L. Amgoud and J. Ben-Naim. Argumentation-based Ranking Logics. In *AAMAS 2015*, pages 1511–1519, 2015.

[3] A. Arioua, M. Croitoru, and P. Buche. DALEK: A Tool for Dialectical Explanations in Inconsistent Knowledge Bases. In *COMMA 2016*, pages 461–462, 2016.

[4] A. Arioua, M. Croitoru, and S. Vesic. Logic-based argumentation with existential rules. *Int. J. Approx. Reasoning*, 90:76–106, 2017.

[5] J.-F. Baget, S. Benferhat, Z. Bouraoui, M. Croitoru, M.-L. Mugnier, O. Papini, S. Rocher, and K. Tabia. Inconsistency-Tolerant Query Answering: Rationality Properties and Computational Complexity Analysis. In *JELIA 2016*, pages 64–80, 2016.

[6] J.-F. Baget, M. Leclère, M.-L. Mugnier, S. Rocher, and C. Sipieter. Graal: A Toolkit for Query Answering with Existential Rules. In *RuleML 2015*, pages 328–344, 2015.

[7] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

[8] P. Besnard, A. J. García, A. Hunter, S. Modgil, H. Prakken, G. R. Simari, and F. Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.

[9] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

[10] M. Bienvenu. On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies. In *AAAI 2012*, 2012.

[11] A. Bondarenko, F. Toni, and R. A. Kowalski. An Assumption-Based Framework for Non-Monotonic Reasoning. In *LPNMR*, pages 171–189, 1993.

[12] C. Bourgaux. *Inconsistency Handling in Ontology-Mediated Query Answering*. PhD thesis, Université Paris-Saclay, Paris, Sept. 2016.

[13] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[14] M. Croitoru and S. Vesic. What Can Argumentation Do for Inconsistent Ontology Query Answering? In *SUM 2013*, pages 15–29, 2013.

[15] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.*, 77(2):321–358, 1995.

[16] G. Flouris and A. Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *IJAR*, 109:55–86, June 2019.

[17] A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138, 2004.

[18] A. Hecham, P. Bisquert, and M. Croitoru. On the Chase for All Provenance Paths with Existential Rules. In *RuleML+RR 2017*, pages 135–150, 2017.

[19] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-Tolerant Semantics for Description Logics. In *RR 2010*, pages 103–117, 2010.

[20] S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[21] S. H. Nielsen and S. Parsons. Computing Preferred Extensions for Argumentation Systems with Sets of Attacking Arguments. In *COMMA 2006*, pages 97–108, 2006.

[22] S. H. Nielsen and S. Parsons. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In N. Maudet, S. Parsons, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 54–73. Springer Berlin Heidelberg, 2007.

[23] M. Ostrowski and T. Schaub. ASP modulo CSP: The clingcon system. *TPLP*, 12(4-5):485–503, 2012.

[24] F. Toni. A tutorial on assumption-based argumentation. *Argument & Computation*, 5(1):89–117, 2014.

[25] B. Yun, M. Croitoru, P. Bisquert, and S. Vesic. Graph Theoretical Properties of Logic Based Argumentation Frameworks. In *AAMAS 2018*, pages 2148–2149, 2018.

[26] B. Yun, S. Vesic, and M. Croitoru. Toward a More Efficient Generation of Structured Argumentation Graphs. In *COMMA 2018*, 2018.

[27] B. Yun, S. Vesic, M. Croitoru, and P. Bisquert. Inconsistency Measures for Repair Semantics in OBDA. In *IJCAI 2018*, pages 1977–1983, 2018.

[28] B. Yun, S. Vesic, M. Croitoru, P. Bisquert, and R. Thomopoulos. A Structural Benchmark for Logical Argumentation Frameworks. In *IDA 2017*, pages 334–346, 2017.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200527

A Discussion Game for the Grounded Semantics of Abstract Dialectical Frameworks

Atefeh KESHAVARZI ZAFARGHANDI, Rineke VERBRUGGE and Bart VERHEIJ
Department of Artificial Intelligence, Bernoulli Institute, University of Groningen, The Netherlands

Abstract. Abstract dialectical frameworks (ADFs) have been introduced as formalism for the modeling and evaluating argumentation. However, the role of discussion in evaluating of arguments in ADFs has not been clarified well so far. We focus on the grounded semantics of ADFs and provide the grounded discussion game. We show that an argument is acceptable (deniable) in the grounded interpretation of an ADF without any redundant links if and only if the proponent of a claim has a winning strategy in the grounded discussion game.

Keywords. Abstract argumentation frameworks, Abstract dialectical frameworks, Discussion games.

1. Introduction

Argumentation has received increased attention within artificial intelligence, since the remarkable paper of Dung [1], in which abstract argumentation frameworks (AFs) are presented. Abstract dialectical frameworks (ADFs) introduced in [2] are expressive generalizations of AFs in which the logical relations among arguments can be represented. Applications of ADFs have been presented in legal reasoning [3,4] and text exploration [5].

Although dialectical methods have a role in determining semantics of both AFs and ADFs, the roles are not immediately obvious from the definition of semantics. To cover this gap, quite a number of works have been presented to show that semantics of AFs can be interpreted in terms of structural discussion [6,7,8,9,10,11]. Further, in [12] it is shown that the structural discussion method has been used in human-machine interaction.

Because of the special structure of ADFs, existing methods used to interpret semantics of AFs cannot be reused in ADFs. To address this problem, we have presented the first existing game for ADFs [13]. That game characterizes the preferred semantics. In this work we focus on the grounded semantics of ADFs.

A key question is 'How is it possible to evaluate arguments in a given ADF?' Answering this question leads to the introduction of several types of semantics, defined based on three-valued interpretations. Different semantics reflect different types of point of view about the acceptance or denial of arguments.

In ADFs an interpretation is called *admissible* if it does not contain any unjustifiable information. Most of the semantics of ADFs are based on the concept of admissibility. An interpretation is *complete* if it exactly contains justifiable information. In addition, an interpretation is *grounded* if it collects all the information that is beyond any doubt. Each ADF has a unique grounded interpretation, which can be the trivial interpretation. Hence for the grounded semantics the credulous and the skeptical decision problems coincide. Further, in the hierarchy, grounded semantics have the lowest computational complexity [14]. However, by indicating whether an argument is credulously acceptable (deniable) in a given ADF under grounded semantics we have the answer of the skeptical decision problem of the argument in question under complete semantics.

In this work we present a game that can answer the credulous and therefore the skeptical decision problem of a given ADF, called the *grounded discussion game*. In [15] it is shown that each ADF is equivalent with an ADF without any redundant links. Thus, without loss of generality, the current game is presented over the subclass of ADFs that do not have redundant links. This game works locally by considering those ancestors of an argument in question that can affect the evaluation of the argument in the grounded interpretation. In this way, the grounded decision problem can be answered without constructing the full grounded interpretation. Further, the current methodology can be used to answer the decision problems under grounded semantics of formalisms that can be represented as ADFs, such as AFs.

In Section 2, we present the relevant background. Then, in Section 3, we present the *grounded discussion game* that can capture the notion of grounded semantics. In Section 4 we present soundness and completeness of the method.

2. Background

The basic definitions in this section are derived from those given in [2,16,17].

Definition 1. *An abstract dialectical framework (ADF) is a tuple $F = (A, L, C)$ where:*

- *A is a finite set of arguments (statements, positions), denoted by letters;*
- *$L \subseteq A \times A$ is a set of links among arguments;*
- *$C = \{\varphi_a\}_{a \in A}$ is a collection of propositional formulas over arguments, called acceptance conditions.*

An ADF can be represented by a graph in which nodes indicate arguments and links show the relation among arguments. Each argument a in an ADF is labelled by a propositional formula, called acceptance condition, φ_a over $par(a)$ such that, $par(a) = \{b \mid (b,a) \in L\}$. The acceptance condition of each argument clarifies under which condition the argument can be accepted [2,16,17]. Further, acceptance conditions indicate the set of links implicitly, thus, there is no need of presenting L in ADFs explicitly.

An argument a is called an *initial argument* if $par(a) = \{\}$. An *interpretation* v (for F) is a function $v : A \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, that maps arguments to one of the three truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$). Truth values can be ordered via the information ordering relation $<_i$ given by $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair of truth values are related by $<_i$. Relation $\leq_i$ is the reflexive and transitive closure of $<_i$. Further, v is called *trivial*, and v is denoted by $v_{\mathbf{u}}$, if $v(a) = \mathbf{u}$ for each $a \in A$. Further, v is called a two-valued interpretation if for each $a \in A$ either $v(a) = \mathbf{t}$ or $v(a) = \mathbf{f}$. Interpretations can be ordered via $\leq_i$ with respect to their information content. Let $\mathcal{V}$ be the set of all interpretations for an ADF F. It is said that an interpretation v is an *extension* of another interpretation w, if $w(a) \leq_i v(a)$ for each $a \in A$, denoted by $w \leq_i v$. Further, we denote the update of an interpretation v with a truth value $x \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ for an argument b by $v|^b_x$, i.e. $v|^b_x(b) = x$ and $v|^b_x(a) = v(a)$ for $a \neq b$.

Semantics for ADFs can be defined via the *characteristic operator* Γ_F which maps interpretations to interpretations. Given an interpretation v (for F), the partial valuation of φ_a by v, is $\varphi^v_a = \varphi_a[b/\top : v(b) = \mathbf{t}][b/\bot : v(b) = \mathbf{f}]$, for $b \in par(a)$. Applying Γ_F on v leads to v' such that for each $a \in A$, v' is as follows:

$$v'(a) = \begin{cases} \mathbf{t} & \text{if } \varphi^v_a \text{ is irrefutable (i.e., } \varphi^v_a \text{ is a tautology)}\,, \\ \mathbf{f} & \text{if } \varphi^v_a \text{ is unsatisfiable (i.e., } \varphi^v_a \text{ is a contradiction),} \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

From now on whenever there is no ambiguity, in order to make three-valued interpretations more readable, we rewrite them by the sequence of truth values, by choosing the lexicographic order on arguments. For instance, $v = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{f}\}$ can be represented by the sequence $\mathbf{tuf}$. The semantics of ADFs are defined via the characteristic operator as in Definition 2.

Definition 2. *Given an ADF F, an interpretation v is:*

- *admissible in F iff $v \leq_i \Gamma_F(v)$, denoted by adm;*
- *preferred in F iff v is $\leq_i$-maximal admissible, denoted by prf;*
- *complete in F iff $v = \Gamma_F(v)$, denoted by com;*
- *a (two-valued) model of F iff v is two-valued and $\Gamma_F(v) = v$, denoted by mod,*
- *the grounded interpretation of F iff v is the least fixed point of Γ_F, denoted by grd.*

The notion of an argument being accepted and the symmetric notion of an argument being denied in an interpretation are as follows.

Definition 3. *Let $F = (A, L, C)$ be an ADF and let v be an interpretation of F.*

- *An argument $a \in A$ is called acceptable with respect to v if φ^v_a is irrefutable.*
- *An argument $a \in A$ is called deniable with respect to v if φ^v_a is unsatisfiable.*

One of the main decision problems of ADFs is whether an argument is credulously acceptable (deniable) under σ semantics, for $\sigma \in \{adm, prf, com, mod, grd\}$. Given an ADF $F = (A, L, C)$, an argument $a \in A$ and a semantics $\sigma \in \{adm, prf, com, mod, grd\}$, argument a is *credulously acceptable (deniable)* under σ if there exists a σ interpretation v of F in which a is acceptable (a is deniable, respectively).

In ADFs, relations between arguments can be classified into four types, reflecting the relationship of attack and/or support that exists between the arguments. These are listed in the following definition.

Definition 4. *Let $D = (S, L, C)$ be an ADF. A relation $(b, a) \in L$ is called*

- supporting *(in D) if for every two-valued interpretation v, $v(\varphi_a) = \mathbf{t}$ implies $v|^b_\mathbf{t}(\varphi_a) = \mathbf{t}$;*
- attacking *(in D) if for every two-valued interpretation v, $v(\varphi_a) = \mathbf{f}$ implies $v|^b_\mathbf{t}(\varphi_a) = \mathbf{f}$;*
- redundant *(in D) if it is both attacking and supporting;*
- dependent *(in D) if it is neither attacking nor supporting.*

Note that the grounded discussion game presented in Section 3 is presented on ADFs without redundant links. Further, in the current work we say that the truth value of *a is presented* in v, if $v(a) = \mathbf{t}/\mathbf{f}$. In addition, for each operator f, the nth power of f is defined inductively i.e. $f^n = f(f^{n-1})$.

3. Grounded Discussion Games

In this section we present a discussion game to answer the credulous (skeptical) decision problem under grounded semantics in a given ADF F does not have any redundant relation, without loss of generality, since any ADF has an equivalent of ADF of this kind; see *Theorem* 4.2.13 of [15].

A grounded discussion game (GDG) is a dispute between a proponent (P) and an opponent (O). We now explain how a GDG works. However, for the formal definition of GDG you may skip it an go to Definition 5. A GDG is started by a claim of P about the truth value of argument a in the grounded interpretation of a given ADF. That is, P believes that the trivial interpretation $g_0 = v_\mathbf{u}$ can be extended to the grounded interpretation that contains the initial claim. O challenges P by asking whether a is an initial argument. If P finds that a is an initial argument and presents the truth value of a to O, then O has to check whether this value is the same as the initial claim. In this case P wins if the checking of O leads to a positive answer. On the other hand, if P answers that a is not an initial argument, then O asks whether an ancestor of a is an initial argument. If P finds that there is no initial argument in the ancestors of a, then the game is stopped and O wins the game.

However, if a is not an initial argument but P finds that b is an ancestor of a which is also an initial argument, then P updates the information of g_0 with $g = g_0|^b_x$, such that x is the truth value of b in the grounded interpretation F. Further, in this step a set of arguments in the shortest paths, between a and b, are presented by P to O. Note that it is possible that there exists more than one shortest path between two arguments. Actually, by presenting g, P says that g can be extended to the grounded interpretation of F.

Now, O checks a piece of information presented in g and the initial claim. If g contains the initial claim, then the game halts and P wins the games. If the information of g is in contradiction with the initial claim, then O wins the game. Since a is not an initial argument, this checking step by O does not lead to acceptance or rejection of the initial claim. That is, presenting of g by P did not convince O about the initial claim.

Thus, O asks P whether P can extend the information of g to an interpretation that contains the initial claim. To this end, P evaluates the acceptance conditions of the children of the argument presented in g under the information of g and presents g'. Then, O continues the game. If O indicates that g' contains the initial claim, then the game stops. If g and g' contain the same piece of information, O asks P for a new initial ancestor of a. Otherwise, O asks P to extend g' more.

The game continues between P and O alternately. P tries to extend the information of g_0 to an interpretation that contains the initial claim to support the belief. O tries to challenge P by either: 1. checking the information of the interpretation which is presented by P as an answer, or 2. asking whether the argument presented in the initial claim is an initial argument, or 3. requesting P to find an ancestor of a which is an initial argument, or 4. requesting P to extend the information of the answer given by P to an interpretation that contains the initial claim. In Example 1 we show how the game works before presenting the formal definitions. If desire you may skip Example 1 and go to Definition 5.

Example 1. *Let $F = (\{a,b,c,d,e,f\}, \{\varphi_a : \bot, \varphi_b : \neg a \vee \neg e, \varphi_c : b \wedge f, \varphi_d : e \wedge \neg c, \varphi_e : \neg f, \varphi_f : \top\})$ be a given ADF, depicted in Figure 1. We know that $grd(F) = \mathbf{fttfft}$. P claims that d is deniable with respect to the grounded interpretation of F. That is, by the initial claim P believes that $d \mapsto \mathbf{f}$ belongs to the grounded interpretation. In other words, the claim of P says that $g_0 = v_\mathbf{u}$ can be extended to the grounded interpretation that contains the initial claim.*

- *P says $g_0 = v_\mathbf{u}$ can be extended to the grounded interpretation of F that contains $d \mapsto \mathbf{f}$.*
- *O asks P whether d is an initial argument.*
- *P checks the acceptance condition of d and the answer is 'no, d is not an initial argument'. Thus, the information of g_0 does not change. For technical reasons we let $g_1 = g_0$.*
- *O challenges P by asking whether any of the ancestors of d is an initial argument.*
- *P checks the acceptance conditions of the parents of d, namely c and e; neither of them is an initial argument. Then, P goes one step further and checks the parents of c and e, which are b and f. Here, f is an initial argument. Since P finds an ancestor of a which is an initial argument, P stops searching. By $\varphi_f : \top$, f is acceptable in the grounded interpretation of F. Thus, P presents interpretation $g_2 = g_1|^f_\mathbf{t} =$ **uuuuut** and set $Ancestors(d,g_1) = \{d,e,c,f\}$, which contains the arguments on the shortest paths between the initial claim d and the initial argument f, that is presented in g_2 but not in g_1. P claims that g_2 can be extended to the grounded interpretation of F that contains the initial claim.*
- *Then O checks the information that is presented by g_2. Since g_2 does not contain any information about the initial claim, O asks P whether P can extend g_2.*
- *To this end, P evaluates the truth value of the children of f that are in $Ancestors(d,g_1)$ under g_2. The children of f that appear in that set are c and e. Thus, P evaluates $\varphi_c^{g_2} \equiv b \wedge \top \equiv b$ and $\varphi_e^{g_2} \equiv \bot$. That is, e is deniable with respect to the grounded interpretation of F. Thus, P presents $g_3 = g_2|^e_\mathbf{f} =$ **uuuuft** to O as an extension of g_2 and P claims that g_3 can be extended to the grounded interpretation of F that contains the initial claim.*

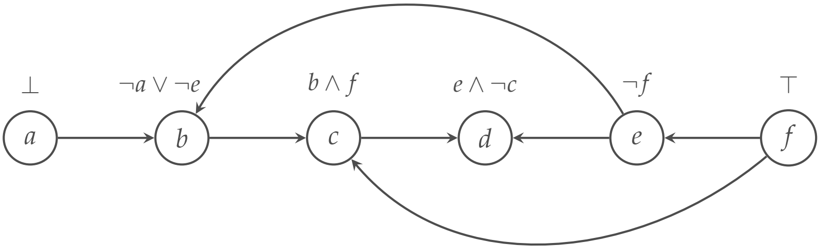


Figure 1. ADF of Examples 1

- *O finds that g_3 extends the information of g_2 and it does not present any information in contrast with the initial claim. However, g_3 does not contain any information about the initial claim. Thus, O asks P whether P can extend g_3 to an interpretation that contains the initial claim.*
- *Again P evaluates the only child of e in set Ancestors(a, g_1), namely d, under g_3. This attempts leads to $g_4 =$* **uuufft**.
- *O checks the information given by g_4. Since g_4 contains the initial claim, the discussion between P and O halts here and P wins the game.*

Here, P does not present the grounded interpretation of F, however, P presents a constructive proof for the initial claim. That is, to indicate the initial claim, P works on the truth value of the argument in question locally. Thus, the grounded discussion game can answer the credulous decision problem under the grounded semantics of an ADF without indicating the truth value of all arguments in the grounded interpretation.

Definition 5. *Let $F = (A, R, C)$ be an ADF, let a be an argument and let S be a set of arguments. Function Par(S) shows the set of parents of the elements of S; function child(a) designates the set of children of a; and function anc(a) presents the set all ancestors of a, defined formally in the following.*

- $Par(S) = \bigcup_{a \in S} par(a)$,
- $child(a) = \{b \mid (a, b) \in R\}$,
- $anc(a) = \bigcup_{n=1}^{m} Par^n(a)$ *such that there exists m with* $Par^m(a) \subseteq \bigcup_{i=1}^{m-1} Par^i(a)$.

Note that whenever S contains only one argument a, $Par(S) = par(a)$ and we write $Par(a)$ for $Par(\{a\})$. The aim of $anc(a)$ is to collect a's ancestors and condition $Par^m(a) \subseteq \bigcup_{i=1}^{m-1} Par^i(a)$ is a guarantee that the function does not go into a loop. If $b \in anc(a)$ is an initial argument, then we call it an *initial ancestor of a*.

The grounded discussion game is defined based on the following moves; some of them are functional moves. For instance, $Eval(g)$ is a unary function, defined over interpretations. Some of them are statement moves to present a claim or a request for instance, $IniAnc(a, g)$ is a statement move by which O asks P to find an initial ancestor of a which is not presented in g.

- $IniClaim(a, x)$: with this statement move P presents her/his beliefs that a is assigned to x such that $x \in \{\mathbf{t}, \mathbf{f}\}$ in the grounded interpretation of F.

- *Ini*(a): with this statement move O asks P whether a is an initial argument.
- *CheckIni*$(a) : A \rightarrow \mathcal{V}$: with this functional move P checks whether a is an initial argument.
- *Check*(g_{i-1}, g_i): with this move O compares the information presented in g_{i-1} and g_i, i.e. whether $g_{i-1} <_i g_i$ or $g_{i-1} \sim_i g_i$.
- *IniAnc*(a,g): with this statement move O asks P to present at least one initial ancestor of a which is not presented in g, together with its truth value.
- *NewIniAnc*$(a,g) : A \times \mathcal{V} \rightarrow \mathcal{V}$: with this functional move P presents initial ancestors of a which are requested by O in *IniAnc*(a,g).
- *Ancestors*$(a,g) : A \times \mathcal{V} \rightarrow 2^A$: with this functional move P presents the set of arguments in the shortest paths between a and the elements of *NewIniAnc*(a,g).
- *Extend*(g): with this statement move O requests P to extend g.
- *Eval*$(g) : \mathcal{V} \rightarrow \mathcal{V}$: with this functional move P evaluates the truth value of the children of the arguments presented in g which appears in the last *Ancestors*$(a,-)$ under g.

In the game, P has the responsibility of constructing a proof for the initial claim. On the other hand, O aims to block the discussion by finding a contradiction or challenging P in such a way that P cannot answer the challenge.

- The game between P and O starts with *IniClaim*(a,x) by which P presents a belief about the truth value of argument a, namely x in the grounded interpretation of F. In this step, intuitively, P believes that $g_0 = v_{\mathbf{u}}$ can be extended to the grounded interpretation that contains the claim.
- Then, O applies statement *Ini*(a), asks whether a is an initial argument.
- Now, it is P's turn to apply function *CheckIni*$(a) : A \rightarrow \mathcal{V}$ to check the acceptance condition of a. If a is an initial argument, then the output of *CheckIni*(a) is $g_1 = g_0|^a_{\mathbf{t}/\mathbf{f}}$. Otherwise, $g_1 = g_0$.
- By *Check*(g_{i-1}, g_i), O checks whether $g_{i-1} <_i g_i$ or $g_{i-1} \sim_i g_i$.
  - ∗ If $g_{i-1} <_i g_i$ and g_i contains the initial claim or the negation of the initial claim, then the game stops.
  - ∗ If $g_{i-1} <_i g_i$ and g_i does not contain any information about the initial claim, then O requests P to extend g_i. That is, O applies *Extend*(g_i).
  - ∗ If $g_{i-1} \sim_i g_i$,
    - ∗ if g_i is the output of either *CheckIni*(a) or *Eval*(g_{i-1}), then O asks P to present a new initial ancestor of a. That is, O applies *IniAnc*(a, g_{i-1}),
    - ∗ if g_i is the output of *NewIniAnc*(a, g_{i-1}), then the game stops.
- After statement move *IniAnc*(a, g_i) by O, P applies function *NewIniAnc*(a, g_i) to find new initial ancestors of a. The output of this function is interpretation g_{i+1} with $g_{i+1} = g_i|^b_{\mathbf{t}/\mathbf{f}}$ such that b is an initial ancestor of a, that was not presented in g_i. This function will be defined precisely in the following.

- Further, after move $\mathit{IniAnc}(a,g_i)$ presented by O, P presents a set of arguments between the initial claim and the elements of $\mathit{NewIniAnc}(a,g_i)$, with the shortest distance, by applying function $\mathit{Ancestors}(a,g_i) : A \times \mathcal{V} \to 2^A$. If there are more than one shortest path between the initial claim and an element of $\mathit{NewIniAnc}(a,g_i)$, then $\mathit{Ancestors}(a,g_i)$ presents the arguments of all paths with the shortest length.
- After statement move $\mathit{Extend}(g_i)$ presented by O, P applies function $\mathit{Eval}(g_i) : \mathcal{V} \to \mathcal{V}$. The output of this function is interpretation g_{i+1} with $g_{i+1} = g_i|^b_{\varphi_b^{g_i}}$ such that b is a child of an argument that is presented in g_i that also appears in the last output of $\mathit{Ancestors}(a,-)$.

The only function that needs more explanation is $\mathit{NewIniAnc}(a,g)$, by which P tries to find the truth values of the initial ancestors of a that are not presented in g. To this end, P uses the modification of the function *anc*, defined in Definition 5, which is called $\mathit{NewAnc}(a,g) : A \times \mathcal{V} \to 2^A$. This function is a binary function that takes the argument a and interpretation g, and returns the set of ancestors of a. However, if there exists an initial ancestor of a, the truth value of which is not indicated in g, then the function stops. This is the reason why this function is called the new ancestors of a with respect to g.

$\mathit{NewAnc}(a,g) = \bigcup_{n=1}^{m} \mathit{Par}^n(a)$ such that there exists m such that $(\mathit{Par}^m(a) \subseteq \mathit{Par}^{m-1}(a)) \vee (\exists p \in \mathit{Par}^m(a)$ such that $\varphi_p \equiv \top / \bot \wedge p$ was not presented in $g)$

Then among the elements of $\mathit{NewAnc}(a,g)$, P looks for the initial arguments. Function $\mathit{NewIniAnc}(a,g) : A \times \mathcal{V} \to \mathcal{V}$, presented in the following, takes a and g, and updates g by adding the truth values of the initial ancestors of a that appear in $\mathit{NewAnc}(a,g)$.

$\mathit{NewIniAnc}(a,g) = g|^b_{\varphi_b^g}$ such that $b \in \mathit{NewAnc}(a,g)$ and b is an initial argument.

Definition 6. *Let $F = (A,R,C)$ be an ADF. A grounded discussion game for credulous acceptance (denial) of $a \in A$ is a sequence $[g_0,\ldots,g_n]$ such that the following conditions hold:*

- $g_0 = v_{\mathbf{u}}$;
- $g_1 = \mathit{CheckIni}(a)$;
- *for* $0 \leq i < n$, $g_i \leq_i g_{i+1}$;
- g_n *contains either*
  - ∗ *the initial claim, or*
  - ∗ *the negation of the initial claim, or*
  - ∗ g_{n-1} *is the output of* $\mathit{NewIniAnc}(a,g_{n-2})$ *and* $g_{n-1} \sim g_n$.
- *for* $1 < i < n$, *if* $g_{i-1} <_i g_i$, *then* g_{i+1} *is the output of* $\mathit{Eval}(g_i)$;
- *for* $0 < i < n$, *if* $g_{i-1} \sim g_i$, *then* g_{i+1} *is the output of* $\mathit{NewIniAnc}(a,g_i)$.

Definition 7. *Let F be a given ADF. Let $[g_0,\ldots,g_n]$ be a grounded discussion game for credulous acceptance (denial) of an argument.*

- *P wins the game if g_n satisfies the initial claim,*
- *O wins the game if g_n satisfies the negation of the initial claim, or $g_{n-1} = NewIniAnc(a,g_{n-2})$ and $g_{n-1} \sim g_n$.*

Example 2 is an instance of a game in which O wins.

Example 2. *Let $F = (\{a,b,c\},\{\varphi_a : \neg b, \varphi_b : \neg c, \varphi_c : \neg a\})$ be an ADF. We know that $grd(F) = v_{\mathbf{u}}$. P claims that b is acceptable in the grounded interpretation of F.*

- *$IniClaim(b,\mathbf{t})$: P believes that g_0 can be extended to the grounded interpretation of F in which b is acceptable.*
- *O asks $Ini(b)$.*
- *P applies $CheckIni(b)$ to answer the challenge. The output is $g_1 = g_0$.*
- *O applies $Check(g_0,g_1)$. Since $g_0 \sim g_1$ and g_1 is the output of $CheckIni(b)$, O requests $IniAnc(b,g_1)$.*
- *To answer $IniAnc(b,g_1)$, P applies $NewIniAnc(b,g_1)$. To this end, first P computes $NewAnc(b,g_1) = \{a,b,c\}$. Since none of them is an initial argument, then the output of $NewIniAnc(b,g_1)$ is $g_2 = g_1$.*
- *O applies $Check(g_1,g_2)$, which leads to $g_1 \sim g_2$. Since g_2 is an output of function $NewIniAnc(b,g_1)$, the game stops and by Definition 7, O wins the game.*

That is, the initial claim of P that b is acceptable with respect to the grounded interpretation of F is false. This corresponds with the fact that the grounded interpretation $v_{\mathbf{u}}$ of F does not satisfy the belief of P.

4. Soundness and Completeness

In this section we show that the presented method is sound and complete. To show the completeness, first we show that in an ADF without any redundant links, the grounded interpretation assigns the truth value of an argument to $\mathbf{t}/\mathbf{f}$ if it is either an initial argument or its truth value is affected by the initial ancestors. This corollary is the direct result of Lemma 1.

Lemma 1. *Let F be an ADF without any redundant link, that does not have any initial argument. Then the grounded interpretation of F is $v_{\mathbf{u}}$.*

Proof. Toward a contradiction, assume that F does not contain an initial argument and $grd(F) \neq v_{\mathbf{u}}$. Let a be an arbitrary argument. We show that $\varphi_a^{v_{\mathbf{u}}}$ is neither irrefutable nor unsatisfiable. Since F does not have any initial argument, a has a parent.

- Consider that a has a parent b such that (b,a) is a dependent link. By the definition of dependent link, there are two-valued interpretations v, w such that $v(\varphi_a) = \mathbf{t}$ and $v|_{\mathbf{t}}^{b}(\varphi_a) \neq \mathbf{t}$, and $w(\varphi_a) = \mathbf{f}$ and $w|_{\mathbf{t}}^{b}(\varphi_a) \neq \mathbf{f}$. Thus, $v,w \in [v_{\mathbf{u}}]_2$ and $v(\varphi_a) \neq w(\varphi_a)$. Therefore, $\varphi_a^{v_{\mathbf{u}}}$ is neither irrefutable nor unsatisfiable.

- Consider that none of the parents of a is dependent. Construct the two-valued interpretation v in which 1. $b \mapsto \mathbf{f}$ if (b,a) is an attacker, and 2. $b \mapsto \mathbf{t}$ if (b,a) is a supporter. Construct the two-valued interpretation w in which 1. $b \mapsto \mathbf{t}$ if (b,a) is an attacker, and 2. $b \mapsto \mathbf{f}$ if (b,a) is a supporter. That is, $v,w \in [v_{\mathbf{u}}]_2$. If either $a \notin par(a)$ or (a,a) is a supporter, then $v(\varphi_a) \equiv \mathbf{f}$ and $w(\varphi_a) \equiv \mathbf{t}$. Thus, $\varphi_a^{v_{\mathbf{u}}}$ is neither irrefutable nor unsatisfiable. If $a \in par(a)$ and (a,a) is an attacker, then $v(\varphi_a) = w(\varphi) = \mathbf{u}$. Thus, $\varphi_a^{v_{\mathbf{u}}}$ is neither irrefutable nor unsatisfiable.

Thus, the assumption that $a \mapsto \mathbf{t}/\mathbf{f} \in grd(F)$ is wrong. Hence, the unique grounded interpretation of F is $v_{\mathbf{u}}$. □

Corollary 2. *Every argument that is acceptable (deniable) with respect to the grounded interpretation of ADF F, without any redundant links, either is an initial argument or has at least one initial ancestor.*

Proof. Let F be an ADF without redundant links. Assume that a is an argument that is not an initial one and does not have any initial ancestor. By the proof method of Lemma 1, $\varphi_a^{v_{\mathbf{u}}}$ is neither irrefutable nor unsatisfiable. Thus, a is neither acceptable nor deniable with respect to the grounded interpretation of F. □

Theorem 3. *(Soundness) Let F be a given ADF. If there is a grounded discussion game for an initial claim of P in which P wins, then the grounded interpretation of F satisfies the initial claim of P.*

Proof. Suppose that the initial claim of P is that 'a is acceptable (deniable) in the grounded interpretation'. Let $[g_0,\ldots,g_n]$ be a grounded discussion game for the initial claim of P, that is, g_n satisfies the initial claim. We show that the grounded interpretation v of F satisfies the initial claim. By the definition the grounded interpretation of F is the least fixed point of the characteristic operator. That is, there exists m such that $\Gamma_F^m(v_{\mathbf{u}}) = v$. We show that $g_n \leq_i v$.

In the grounded discussion game if $n = 1$, that is $[g_0,g_1]$, then a is an initial argument. Thus, clearly $g_1 \leq_i \Gamma_F(v_{\mathbf{u}})$. Since Γ is a monotonic operator, $g_1 \leq_i v$. Consider that in the grounded discussion game $n > 1$. By induction on n it is easy to show that for each m with $0 \leq m \leq n$, $g_m \leq_i v$ holds.

Therefore, in the grounded discussion game $[g_0,\ldots,g_n]$ for any i with $0 \leq i \leq n$, $g_i \leq_i v$ holds. In specific, $g_n \leq_i v$. Thus, the initial claim of P is satisfied in the grounded interpretation of F.

□

Definition 8. *Let F be an ADF. The distance from argument a to b in F is the distance from a to b in the associated directed graph of F, denoted by $d(a,b)$. That is, $d(a,b)$ is the length of a shortest directed path from a to b in the directed graph associated to F.*

Theorem 4. *(Completeness) Let F be a given ADF without any redundant links. If a is acceptable (deniable) in the grounded interpretation of F, then there is the grounded discussion game for the initial claim of accepting (denying) of a.*

Proof. Let F be an ADF and let v be the grounded interpretation of F. Further, let a be an argument which is accepted (denied) with respect to v. Since F does not have any redundant links, by Corollary 2, either a is an initial argument or a has at least one initial ancestor. We construct a grounded discussion game for the initial claim of $a \mapsto \mathbf{t}/\mathbf{f}$ in which P wins. Let $g_0 = v_{\mathbf{u}}$. If a is an initial argument, then $g_1 = g_0|^{a}_{\mathbf{t}/\mathbf{f}}$. Thus, $[g_0, g_1]$ is the grounded discussion game, in which $g_1 = \mathit{CheckIni}(a)$, that satisfies the initial claim.

If a is not an initial claim, then let $g_{1_1} = g_0$ and list the set of initial ancestors of a, for instance $L = [a_1, \ldots, a_k]$. Assume that L is ordered based on the distance to a, increasingly. That is, $d(a_i, a) \leq d(a_{i+1}, a)$, for i with $1 \leq i < k$. Let us categorize L based on the distance of arguments to a. For instance, let $L_1 = \{a_1\} \cup B$ such that $B = \{a_i \mid d(a_i, a) = d(a_1, a)\}$. If $B \neq \{\}$, then m is an integer such that $d(a_m, a) = d(a_1, a)$ and $m > i$ for $a_i \in B$, otherwise, $m = 1$. Let $L_2 = \{a_i \mid d(a_i, a) = d(a_{m+1}, a)\}$. Continue this process. Since L is finite, there exists p such that $L = \bigcup_{i=1}^{p} L_i$.

Let $g_{2_1} = g_{1_1}|^{b}_{v(b)}$ such that $b \in L_1$. For $j \geq 1$, for $i \geq 2$, 1. if $g_{i_j} > g_{i-1_j}$, then let $g_{i+1_j} = g_{i_j}|^{b}_{v(b)}$ such that b is a child of an argument in L_j that is on a path between a and an element of L_j. 2. If $g_{i_j} \sim g_{i-1_j}$, then let $g_{i+1_j} = g_{i_j}|^{b}_{v(b)}$ such that $b \in L_{j+1}$. If any of the g_{i_j} satisfies the initial claim, then stop the above loop.

Because the number of arguments on the paths between a and elements of L is finite, then the above loop will stop. Consider that the above loop halts in g_{i_j}. We claim that $D = [g_0, \ldots, g_{i_j}]$ is the GDG that satisfies the initial claim. To show that D is a GDG it is enough to show that D satisfies the fourth item of Definition 6. Assume that $a \mapsto \mathbf{t} \in v$. Toward a contradiction, assume that $a \mapsto \mathbf{t} \notin g_{i_j}$. Since each element of D is the update of the previous interpretation in D by updating the truth value of a b with $v(b)$, it is not possible that $a \mapsto \mathbf{f} \in g_{i_j}$. On the other hand, $a \mapsto \mathbf{u} \in g_{i_j}$ means that there is c initial ancestor of a that $v(c) = \mathbf{u}$. It is a contradiction that v is the grounded interpretation of F. □

5. Conclusion

Grounded discussion games between two agents are presented in this work to answer the credulous decision problem of ADFs under grounded semantics. Since each ADF is equivalent with an ADF without any redundant links, we present the game over this subclass of ADFs. If the graph associated to a given ADF is disconnected, then the current method only checks the ancestors of the argument in question to answer the decision problem and not the whole graph. Thus, in general, even in the worst case, the presented method does not coincide with the least-fixed-point algorithm of grounded interpretation. Further, the method is sound and complete. In each move, P tries to show that the initial claim can be in an extension of the trivial interpretation, and O tries to challenge P by checking the content of the interpretation presented by P and either finding the initial claim or requesting P to extend the interpretation or find a new initial ancestor. As future work, we are investigating a game for infinite ADFs and for ADFs for which the acceptance conditions are not restricted to propositional formulas.

Acknowledgements. Supported by the Center of Data Science & Systems Complexity (DSSC) Doctoral Programme, at the University of Groningen.

References

[1] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.

[2] G. Brewka and S. Woltran, "Abstract dialectical frameworks," in *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pp. 102–111, 2010.

[3] L. Al-Abdulkarim, K. Atkinson, and T. J. M. Bench-Capon, "A methodology for designing systems to reason with legal cases using abstract dialectical frameworks," *Artificial Intelligence and Law*, vol. 24, no. 1, pp. 1–49, 2016.

[4] L. Al-Abdulkarim, K. Atkinson, and T. J. M. Bench-Capon, "Abstract dialectical frameworks for legal reasoning," in *Legal Knowledge and Information Systems JURIX*, vol. 271 of *Frontiers in Artificial Intelligence and Applications*, pp. 61–70, IOS Press, Amsterdam, 2014.

[5] E. Cabrio and S. Villata, "Abstract dialectical frameworks for text exploration," in *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART (2)*, pp. 85–95, SCITEPRESS-Science and Technology Publications, Lda, 2016.

[6] H. Jakobovits and D. Vermeir, "Dialectic semantics for argumentation frameworks," in *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, pp. 53–62, ACM Press, New York, 1999.

[7] H. Prakken and G. Sartor, "Argument-based extended logic programming with defeasible priorities," *Journal of Applied Non-classical Logics*, vol. 7, no. 1-2, pp. 25–75, 1997.

[8] S. Modgil and M. Caminada, "Proof theories and algorithms for abstract argumentation frameworks," in *Argumentation in Artificial Intelligence* (G. R. Simari and I. Rahwan, eds.), pp. 105–129, Springer, 2009.

[9] M. Caminada, "Argumentation semantics as formal discussion," in *Handbook of Formal Argumentation* (P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, eds.), pp. 487–518, 2018.

[10] P. M. Dung and P. M. Thang, "A sound and complete dialectical proof procedure for sceptical preferred argumentation," in *Proceedings of the LPNMR-Workshop on Argumentation and Nonmonotonic Reasoning (ArgNMR07)*, pp. 49–63, 2007.

[11] F. H. van Eemeren, B. Garssen, E. C. W. Krabbe, A. F. S. Henkemans, B. Verheij, and J. H. M. Wagemans, eds., *Handbook of Argumentation Theory*. New York: Springer, 2014.

[12] R. Booth, M. Caminada, and B. Marshall, "DISCO: A web-based implementation of discussion games for grounded and preferred semantics," in *Proceedings of Computational Models of Argument COMMA* (S. Modgil, K. Budzynska, and J. Lawrence, eds.), pp. 453–454, IOS Press, Amsterdam, 2018.

[13] A. Keshavarzi Zafarghandi, R. Verbrugge, and B. Verheij, "Discussion games for preferred semantics of abstract dialectical frameworks," in *European Conference on Symbolic and Quantitative Approaches with Uncertainty* (G. Kern-Isberner and Z. Ognjanovic, eds.), pp. 62–73, Springer, Berlin, 2019.

[14] H. Strass and J. P. Wallner, "Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory," *Artificial Intelligence*, vol. 226, pp. 34–74, 2015.

[15] S. Polberg, *Developing the abstract dialectical framework*. PhD thesis, PhD thesis, TU Wien, Institute of Information Systems, 2017.

[16] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, and S. Woltran, "Abstract dialectical frameworks revisited," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pp. 803–809, 2013.

[17] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, and S. Woltran, "Abstract dialectical frameworks. An overview," *IFCoLog Journal of Logics and their Applications (FLAP)*, vol. 4, no. 8, 2017.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200528

Case-Based Reasoning with Precedent Models: Preliminary Report

Heng ZHENG [a,1], Davide GROSSI [a,b,c] and Bart VERHEIJ [a]
[a] *Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, The Netherlands*
[b] *Amsterdam Center for Law and Economics, University of Amsterdam, The Netherlands*
[c] *Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands*

Abstract. Formalizing case-based reasoning is an important topic in AI and Law, which has been discussed using various approaches, such as formal dialogue games, abstract dialectical frameworks. In this paper we model case-based reasoning by using the formal argument semantics of case models. With the precedent models we present, the validity of legal arguments in the case-based reasoning process can be shown formally. We also present a case study of precedent models in a real legal domain and evaluate the validity of arguments in case-based reasoning.

Keywords. legal argumentation, case-based reasoning, precedent

1. Introduction

This paper focuses on the formalization of case-based reasoning within the case model formalism developed in [1]. Case-based reasoning is an important topic in AI and Law. Research in this topic is associated with argumentation as arguments are main outcomes in case-based reasoning, which are given by the parties in courts to defend their positions. Notions in case-based reasoning such as case comparisons and legal argument evaluation are good examples of computational argumentation theory.

As summarized by Bench-Capon [2], HYPO and its successors have exercised great influence in the study of case-based reasoning. Ashley and other researchers model legal reasoning by representing cases with factors [3, 4, 5, 6, 7, 8]. With a current situation, users can retrieve precedents according to the shared factors between precedents and the situation, and try to analogize and distinguish them.

Case-based reasoning can be formalized and connected with other reasoning approaches. Hafner and Berman discuss the role of context in case-based reasoning [9]. Wyner and his colleagues discuss the relation between cases and arguments [10]. Prakken and Sartor [11] model case-based reasoning in a formal dialogue game and combine case-based reasoning with rule-based reasoning. Prakken and his colleagues [12] also

[1] Corresponding Author: Heng Zheng, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands; E-mail: h.zheng@rug.nl.

model case-based reasoning with ASPIC+ framework [13]. Horty and Bench-Capon [14] present a formalism on case-based reasoning with reason models that combine facts, rules and outcomes in precedents. Horty formalizes dimensions in case-based reasoning instead of standard factors [15]. Al-Abdulkarim and her colleagues [16] formalize CATO with abstract dialectical frameworks [17]. Other approaches include dialectical arguments [18], ontologies in OWL [19] and abstract argumentation [20].

In [1], Verheij presents a form of argumentation theory which formally defines the validity of arguments through case models [21]. A case model is a set of cases combined with a preference relation. The theory has been implemented in a Prolog program [22]. With case models, the validity of generated arguments and attacks can be distinguished in coherent, presumptive and conclusive validity. It provides formal semantics for combining rules, arguments and cases [1]. With this approach, we can analyze case-based reasoning with logical tools and evaluate arguments formally.

Outline: we show the theory part of the precedent model formalism based on Verheij's approach in Section 2. We give a case study of precedent models in a real legal domain in Section 3. We discuss how arguments' validity can be used in case-based reasoning in Section 4. We also compare our approach with others in the discussion section.

2. Precedent models

The precedent models defined in this section are based on the case models formalism addressed by Verheij [1]. The formalism introduced in this paper uses a propositional logic language L generated from a set of propositional constants. We write $\neg$ for negation, $\wedge$ for conjunction, $\vee$ for disjunction, $\leftrightarrow$ for equivalence. The associated classical, deductive, monotonic consequence relation is denoted $\models$.

Precedents consist of factors and outcomes. Both *factors* and *outcomes* are literals. A literal is either a propositional constant or its negation. We use $F \subseteq L$ to represent a set of factors, $O \subseteq L$ to represent a set of outcomes. The sets F and O are disjoint and consist only of literals. If a propositional constant p is in F (or O), then $\neg p$ is also in F (respectively in O). A factor represents an element of a case, namely a factual circumstance. Its negation describes the opposite fact. For instance, if a factor φ is "A kills B", then its negation $\neg\varphi$ is "A does not kill B". An outcome always favors a side in the precedent, its negation favors the opposite side. For instance, an outcome ω is "A is guilty", its negation $\neg\omega$ is "A is not guilty".

Following existing work in case-based reasoning, a *precedent* is a logical consistent conjunction of factors and outcomes. If a precedent contains an outcome, then we say it is a *proper precedent*. If a precedent doesn't have any outcome, then it is a *situation* that describes a current case. The outcomes of these situations need to be decided upon.

Definition 1 (Precedents) *A* precedent *is a logically consistent conjunction of distinct factors and outcomes $\pi = \varphi_0 \wedge \varphi_1 \wedge \ldots \wedge \varphi_m \wedge \omega_0 \wedge \omega_1 \wedge \ldots \wedge \omega_{n-1}$, where m and n are non-negative integers. We say that $\varphi_0, \varphi_1, ..., \varphi_m$ are the* factors *of π, $\omega_0, \omega_1, ..., \omega_{n-1}$ are the* outcomes *of π. If $n = 0$, then we say that π is a* situation *with no outcomes, otherwise π is a* proper precedent.

Notice that both m and n can be equal to 0. When $m = 0$, there is one single factor. When $n = 0$, the precedent has no outcome and the empty conjunction $\omega_0 \wedge \ldots \wedge \omega_{n-1}$ is

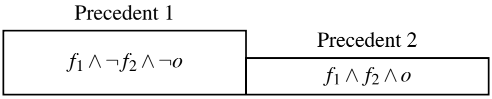

Precedent 1
$f_1 \wedge \neg f_2 \wedge \neg o$
Precedent 2
$f_1 \wedge f_2 \wedge o$

Figure 1. An example of precedent model

equivalent to $\top$. Figure 1 shows two precedents. f_1, f_2 and $\neg f_2$ are factors, o and $\neg o$ are outcomes.

We do not assume precedents are complete descriptions. That is, factors may exist which do not occur in the precedent. Furthermore, we do not assume that the negation of a factor holds when the factor does not occur in the precedent.

A *precedent model* is a set of logically incompatible precedents forming a total preorder that can represent a preference among the precedents. This preference relation is determined by the purpose of the models, for instance, in the precedent model shown in Section 3, precedents are equally preferred because of the setting of HYPO.

Definition 2 (Precedent models) *A* precedent model *is a pair* $(P, \geq)$*, such that: P is a set of precedents; for all* $\pi, \pi' \in P$ *with* $\pi \neq \pi'$, $\pi \wedge \pi' \models \bot$*; and* $\geq$ *is a total preorder on P.*

The strict weak order $>$ between two precedents π and π' is standardly associated with a total preorder $\geq$ which is defined as $\pi > \pi'$ if and only if it is not the case that $\pi' \geq \pi$ (for π and $\pi' \in P$). When $\pi > \pi'$, we say that π is (strictly) preferred to π'. The associated equivalence relation $\sim$ is defined as $\pi \sim \pi'$ if and only if $\pi \geq \pi'$ and $\pi' \geq \pi$.

Precedent models are case models as defined in [1, Definition 2.1]:

Proposition 1 *Let* $(P, \geq)$ *be a precedent model. The following properties hold, for all* π, π' *and* $\pi'' \in P$*:*

1. $\not\models \neg\pi$;
2. *If* $\not\models \pi \leftrightarrow \pi'$*, then* $\models \neg(\pi \wedge \pi')$;
3. *If* $\models \pi \leftrightarrow \pi'$*, then* $\pi = \pi'$;
4. $\pi \geq \pi'$ *or* $\pi' \geq \pi$;
5. *If* $\pi \geq \pi'$ *and* $\pi' \geq \pi''$*, then* $\pi \geq \pi''$.

The proof is straightforward and is omitted. Figure 1 shows an example of a precedent model. The preference relation of this model is Precedent 1 > Precedent 2, and is denoted by the size of the boxes directly.

The definitions of arguments, attacks and their validities in case models [1] can be applied to precedent models. Precedents are considered as the cases made by arguments. In Figure 1, Precedent 1 is the case made by the argument from $f_1 \wedge \neg f_2$ to $\neg o$.

Definition 3 (Arguments [1]) *An* argument *from* χ *to* ρ *is a pair* (χ, ρ) *with* χ *and* $\rho \in L$. *For* $\lambda \in L$*, if* $\chi \models \lambda$*,* λ *is a* premise *of the argument; if* $\rho \models \lambda$*,* λ *is a* conclusion*; if* $\chi \wedge \rho \models \lambda$*,* λ *is a* position *in the case made by the argument. We say that* χ *expresses the* full premise *of the argument,* ρ *the* full conclusion*, and* $\chi \wedge \rho$ *its* full position*, also referred to as the* case made *by the argument.*

Arguments have three kinds of validities. If an argument is logically implied by one of the precedents in a precedent model, then the argument is *coherently valid* in the precedent model. If all precedents in a precedent model logically implying an argument's full premise also logically imply its full conclusion, then the argument is *conclusively valid* in the precedent model. If an argument's conclusion is logically implied by a precedent which is the most preferred among the precedents that logically imply the argument's full premise, then the argument is *presumptively valid* in the precedent model.

Definition 4 (Validity of arguments [1]) *Let $(P,\geq)$ be a precedent model. We define, for all $\chi,\rho \in L$:*

1. *argument (χ,ρ) is* coherently valid *with respect to the precedent model if and only if $\exists\pi \in P : \pi \models \chi \wedge \rho$. We then write $(P,\geq) \models (\chi,\rho)$;*
2. *argument (χ,ρ) is* conclusively valid *with respect to the precedent model if and only if $\exists\pi \in P : \pi \models \chi \wedge \rho$ and for all $\pi \in P$: if $\pi \models \chi$, then $\pi \models \chi \wedge \rho$. We then write $(P,\geq) \models \chi \Rightarrow \rho$;*
3. *argument (χ,ρ) is* presumptively valid *with respect to the precedent model if and only if $\exists\pi \in P$:*
   (a) $\pi \models \chi \wedge \rho$; and
   (b) for all $\pi' \in P$: if $\pi' \models \chi$, then $\pi \geq \pi'$.
   We then write $(P,\geq) \models \chi \rightsquigarrow \rho$.

In the precedent model of Figure 1, the following are true:

1. $(P,\geq) \models (f_1,o)$, as Precedent 2 logically implies this argument;
2. $(P,\geq) \models f_2 \Rightarrow o$, as all precedents in the model which logically imply f_2 also logically imply o.
3. $(P,\geq) \models f_1 \rightsquigarrow \neg o$, as Precedent 1 logically implies $\neg o$ and that is the most preferred precedent which logically implies f_1 in the model.

Definition 5 (Successful attacks [1]) *Let $(P,\geq)$ be a precedent model, and (χ,ρ) be a presumptively valid argument:*

1. *$\tau \in L$ is a* successful attack *of argument (χ,ρ) if and only if $(\chi \wedge \tau,\rho)$ is not a presumptively valid argument, then we say $(P,\geq) \models \chi \rightsquigarrow \rho \times \tau$;*
2. *If argument $(\chi \wedge \tau,\rho)$ is not coherently valid, then we say successful attack τ is* excluding*;*
3. *If $\exists\pi \in P$, and $\pi \models \chi \wedge \tau$, then we say π provides* grounding *for the successful attack τ;*

In the precedent model shown above, we have $(P,\geq) \models f_1 \rightsquigarrow \neg o \times f_2$, it is an excluding successful attack, Precedent 2 provides grounding for this successful attack.

The comparisons are based on Definition 3.1 and 3.4 in [1]. *Analogies* between two precedents are the properties that follow logically from both two precedents. *Distinctions* are the unshared properties between two precedents, namely the properties that only follow logically from one of the precedents but not the other one.

Definition 6 (Precedent comparisons) *Let $\pi,\pi' \in L$ be two precedents, we define:*

1. *a sentence $\alpha \in L$ is an* analogy between π and π' *if and only if $\pi \vee \pi' \models \alpha$.*
2. *a sentence $\delta \in L$ is a π-π'* distinction *if and only if $\pi \models \delta$ and $\pi' \not\models \delta$.*

A π-π' distinction is a distinction in π with respect to π'. Both π-π' distinctions and π'-π distinctions are called the distinctions between π and π'.

Comparing the two precedents in Figure 1, f_1 is an analogy between them, f_2 is a distinction in Precedent 2 with respect to Precedent 1. Note that the outcomes are also distinctions between them as these precedents are decided differently.

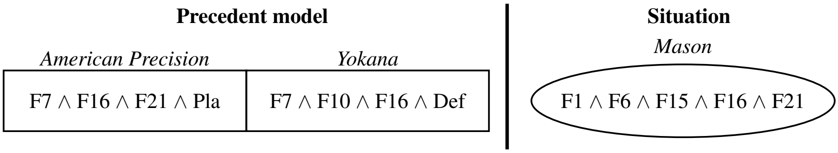


Figure 2. Precedent model for the *Mason* case

3. Case study: HYPO in precedent models

In this section, we give a case study of precedent models using a legal domain studied in HYPO, namely the trade secret law in the United States [3, 23].

The precedent model $(P, P \times P)$ in this case study contains two precedents (i.e. the *Yokana* case[2] and the *American Precision* case[3]), which have been discussed in [23, Chapter 3.3.2]. Notice that $P \times P$ denotes the trivial preference relation where all precedents are equivalent. The current situation is adapted from the *Mason* case[4]. The precedents in this model have equal preference. We assume the set of outcomes $O = \{\text{Pla}, \text{Def}\}$ and $\models \text{Def} \leftrightarrow \neg\text{Pla}$. Pla stands for plaintiff wins the claim, Def stands for defendant wins the claim. The *Yokana* case favors defendant and the *American Precision* case favors plaintiff. As shown in Figure 2, both of them share some factors with the *Mason* case.

In the case-based reasoning process with these two precedents, arguments [23, Figure 3.2] can be generated for discussing the current situation:

1. $(\text{F16}, \text{Def})$ Defendant cites the *Yokana* case in order to give a statement that defendant should win the case. F16 is an analogy between the *Yokana* case and situation.
2. $(\text{F10}, \text{Pla})$ and $(\text{F6} \wedge \text{F15} \wedge \text{F21}, \text{Pla})$ Plaintiff distinguishes the *Yokana* case by pointing out distinctions (F10 and $\text{F6} \wedge \text{F15} \wedge \text{F21}$) in order to suggest that the situation should be decided differently.
3. $(\text{F16} \wedge \text{F21}, \text{Pla})$ Plaintiff also cites a more on point counterexample (the *American Precision* case) which shares more factors ($\text{F16} \wedge \text{F21}$) with the situation.
4. $(\text{F1}, \text{Def})$ and $(\text{F7}, \text{Def})$ Defendant distinguishes the counterexample, namely the *American Precision* case, by using factor F1 and F7.

The evaluation of these arguments is shown as below:

$(P, P \times P) \models \text{F16} \rightsquigarrow \text{Def}$ $\quad$ $(P, P \times P) \not\models (\text{F10}, \text{Pla})$
$(P, P \times P) \not\models (\text{F6} \wedge \text{F15} \wedge \text{F21}, \text{Pla})$ $\quad$ $(P, P \times P) \models \text{F16} \wedge \text{F21} \Rightarrow \text{Pla}$
$(P, P \times P) \not\models (\text{F1}, \text{Def})$ $\quad$ $(P, P \times P) \models \text{F7} \rightsquigarrow \text{Def}$

The evaluation shows that arguments for analogizing the *Yokana* case and the *American Precision* case with the *Mason* case are at least presumptively valid in the model, while most of the arguments for distinguishing them with the situation are incoherent.

Some factors can be considered as successful attacks of arguments in case-based reasoning. For instance:

$(P, P \times P) \models \text{F16} \wedge \text{F21} \rightsquigarrow \text{Pla} \times \text{F1}$ $\quad$ $(P, P \times P) \models \text{F16} \rightsquigarrow \text{Def} \times \text{F6}$
$(P, P \times P) \models \text{F16} \rightsquigarrow \text{Def} \times \text{F15}$ $\quad$ $(P, P \times P) \models \text{F16} \rightsquigarrow \text{Def} \times \text{F21}$

[2] Midland-Ross Corp. v. Yokana, 293 F.2d 411 (3rd Cir.1961)

[3] American Precision Vibrator Co. v. National Air Vibrator Co., 764 S.W.2d 274 (Tex.App.-Houston [1st Dist.] 1988)

[4] Mason v. Jack Daniel Distillery, 518 So.2d 130 (Ala.Civ.App.1987)

According to Definition 5, all these attacks are excluding. The *American Precision* case provides grounding for the attack F21 on argument $(\mathrm{F16},\mathrm{Def})$, and there is no precedent that can provide grounding for other successful attacks above.

4. Preliminary evaluation of arguments' validity in case-based reasoning

In this section, we discuss the validity of arguments in case-based reasoning. The approach we use to construct precedent models allows us to evaluate the arguments' validity in case-based reasoning, which can give reasons for why such argument moves in the reasoning process can be taken. For instance, the argument citing the *Yokana* case is presumptively valid, but the one based on *American precision* is even conclusive, and hence stronger in a formal sense.

The evaluation of arguments' validity provides a strategy to manipulate arguments in case-based reasoning, which is aiming for improving arguments' validity. By this action, the arguer's standpoint can be more acceptable to the judge. For instance, in the case study, if defendants find a favorable precedent which contains factor F7 and F1, then the level of validity of the argument they used for distinguishing the *American Precision* case with the *Mason* case will become conclusive, which makes their distinction more acceptable.

Using an incoherent argument can make sense and break new ground. A decision based on such an argument can be considered as going beyond the current legal status modeled in the precedent model. After adding a precedent incorporating such a groundbreaking decision, the previously incoherent argument can become coherent in the adapted model. For instance, if *Mason*'s decision favors plaintiff, then in the current precedent model $(\mathrm{F1} \wedge \mathrm{F6} \wedge \mathrm{F15} \wedge \mathrm{F16} \wedge \mathrm{F21},\mathrm{Pla})$ is incoherent, but in the precedent model with the decided version of *Mason* included as third precedent it is coherent.

Precedents can be compared not only by the shared factors, but also by the preference relation in precedent models. The precedent model in Section 3 has equal preference, but this relation can change if the precedents are from different court level. Even with the same facts, a higher level court can make a decision which is opposite to the decision of a lower level court. Assume for instance that the precedent model in the case study has another precedent from a higher court level with the same factors as the *American Precision* case but opposite outcome, and it is more preferred than other precedents, then the argument $(\mathrm{F16} \wedge \mathrm{F21},\mathrm{Def})$ is presumptively valid, while $(\mathrm{F16} \wedge \mathrm{F21},\mathrm{Pla})$ is only coherent. Although they share the same factors with the situation, the precedent from higher court level can still be considered as a better one, since the argument for citing it has stronger validity.

5. Discussion

In this section, we compare our precedent models with other relevant research.

Starting with HYPO, we observe that HYPO represents factors with dimensions, which can represent graduality or strength (very low - low - neutral - high - very high). In precedent models, factors are more similar to the notion in CATO [7], namely all factors are binary, and either can be found in a case or not. A pair of opposite factors in our models can be considered as the two extremes (very low and very high) of a dimension.

Precedent models are an extension of the case model formalism in [1], which briefly discussed case analogies and distinctions. Cases in case models are abstract in the sense that factors and outcomes are not distinguished as in our notion of precedents. With the precedent models, we therefore are able to describe elements of case-based reasoning in formal logic while staying closer to notions studied in HYPO.

In other case-based reasoning models [11, 12, 14, 16], case-based reasoning has been modeled in terms of a formal dialogue game [11], in terms of ASPIC+ framework [12], in terms of a reason model [14], and in terms of abstract dialectical frameworks [16]. The theory we use is in terms of a propositional logical language.

Precedents in [11] are represented with sets of rules, expressing which factors favor an outcome and which detract from it. They also describe factors as a kind of rules in order to represent the conflict resolution between the pro and con factors. Precedents in [12] are sets of factors, they use predicates in a first-order language to describe factors in precedents. Horty and Bench-Capon [14] represent precedents as a combination of rules, facts and outcome. The representation method used by Al-Abdulkarim and her colleagues is related to the factor hierarchy used in CATO [7]. In our precedent models, precedents are represented with conjunctions of factors and outcomes instead of sets or hierarchies. Rules can be translated to our arguments, and therefore the validity of rules can vary in our models.

The meaning of the preference relation is also different. In [11, 12, 14], the preference relation is used inside a precedent, it is either between rules [11] which is determined by which rule has the priority, or between factors [12, 14] which is determined by the outcome of the precedent. A similar notion of preference relation is also used in [16], which comes from prioritized abstract dialectical frameworks. It is used for comparing factors in the hierarchy. In our approach, the preference relation is a relation between precedents. For instance, in our precedent model for HYPO-style reasoning (Section 3), precedents are with equal preference. In the case model for Dutch tort law [1], the preference of cases are different in order to represent which cases are exceptional.

6. Conclusion

In this paper, we study case-based reasoning with precedent models and how to formalize arguments' validity with the formal logical semantics of precedent models. This approach shows that Verheij's formalism [1], which uses a formal, logical language, can be used to formally model elements of case-based reasoning. The use of precedent models allows for the logical evaluation of arguments grounded in past cases. Since past cases can be considered as a kind of data, and valid arguments show patterns that hold in the data, the approach provides a step in the development of hybrid AI systems that combine structured knowledge grounded in data [24].

References

[1] B. Verheij. Formalizing Arguments, Rules and Cases. In *Proceedings of the sixteenth International Conference on Articial Intelligence and Law*, ICAIL 2017, pages 199–208. ACM, New York, 2017.

[2] T. Bench-Capon. HYPO's legacy: introduction to the virtual special issue. *Artificial Intelligence and Law*, 25(2):205–250, 2017.

[3] K. D. Ashley. *Modeling Legal Arguments: Reasoning with Cases and Hypotheticals*. MIT Press, Cambridge, 1990.

[4] E. L. Rissland and K. D. Ashley. A Case-Based System for Trade Secrets Law. In *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, ICAIL 1987, pages 60–66. ACM, New York, 1987.

[5] L. K. Branting. Building explanations from rules and structured cases. *International Journal of Man-Machine Studies*, 34(6):797–837, 1991.

[6] D. B. Skalak and E. L. Rissland. Arguments and Cases: An Inevitable Intertwining. *Artificial Intelligence and Law*, 1(1):3–44, 1992.

[7] V. Aleven. Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence*, 150(1-2):183–237, 2003.

[8] K. D. Ashley and S. Brüninghaus. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2):125–165, 2009.

[9] C. L. Hafner and D. H. Berman. The Role of Context in Case-Based Legal Reasoning: Teleological, Temporal, and Procedural. *Artificial Intelligence and Law*, 10(1–3):19–64, 2002.

[10] A. Wyner, T. Bench-Capon, and K. Atkinson. Three Senses of “Argument”. In G. Sartor, P. Casanovas, R. Rubino, and N. Casellas, editors, *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, pages 146–161. Springer, Berlin, 2008.

[11] H. Prakken and G. Sartor. Modelling Reasoning with Precedents in a Formal Dialogue Game. *Artificial Intelligence and Law*, 6:231–287, 1998.

[12] H. Prakken, A. Wyner, T. Bench-Capon, and K. Atkinson. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *Journal of Logic and Computation*, 25(5):1141–1166, 05 2013.

[13] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[14] J. Horty and T. Bench-Capon. A factor-based definition of precedential constraint. *Artificial Intelligence and Law*, 20(2):181–214, 2012.

[15] J. Horty. Reasoning with dimensions and magnitudes. *Artificial Intelligence and Law*, 27(3):309–345, 2019.

[16] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon. A methodology for designing systems to reason with legal cases using Abstract Dialectical Frameworks. *Artificial Intelligence and Law*, 24(1):1–49, 2016.

[17] G. Brewka, H. Strass, S. Ellmauthaler, J.P. Wallner, and S. Woltran. Abstract Dialectical Frameworks Revisited. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 803–809. AAAI Press, 2013.

[18] B. Roth and B. Verheij. Dialectical Arguments and Case Comparison. In T.F. Gordon, editor, *Legal Knowledge and Information Systems. JURIX 2004: The Seventeenth Annual Conference*, pages 99–108. IOS Press, Amsterdam, 2004.

[19] A. Wyner. An ontology in OWL for legal case-based reasoning. *Artificial Intelligence and Law*, 16(4):361, 2008.

[20] K. Cyras, K. Satoh, and F. Toni. Abstract Argumentation for Case-Based Reasoning. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 549–552. AAAI Press, 2016.

[21] B. Verheij. Correct Grounded Reasoning with Presumptive Arguments. In L. Michael and A. Kakas, editors, *15th European Conference on Logics in Artificial Intelligence, JELIA 2016. Larnaca, Cyprus, November 9–11, 2016. Proceedings (LNAI 10021)*, pages 481 – 496. Springer, Berlin, 2016.

[22] H. Zheng, M. Xiong, and B. Verheij. Checking the Validity of Rule-Based Arguments Grounded in Cases: A Computational Approach. In M. Palmirani, editor, *Legal Knowledge and Information Systems. JURIX 2018: The Thirty-first Annual Conference*, volume 313, pages 220 – 224. IOS Press, Amsterdam, 2018.

[23] K. D. Ashley. *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, Cambridge, 2017.

[24] B. Verheij. Artificial Intelligence as Law. Presidential Address to the Seventeenth International Conference on Artificial Intelligence and Law. *Artificial Intelligence and Law*, 28:181–206, 2020.

Demonstrations

This page intentionally left blank

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200530

deliberate – Online Argumentation with Collaborative Filtering

Markus BRENNEIS [a,1], Martin MAUVE [a]
[a] *Department of Computer Science, University of Düsseldorf, Germany*

Abstract. We demonstrate *deliberate*, a full-stack web application to exchange arguments with other users. Collaborative filtering utilizing a specialized metric, which considers the structure of the argumentation tree, is used to suggest arguments which the user is likely to accept.

Keywords. online argumentation, artificial intelligence, collaborative filtering

1. Introduction

Exchanging arguments and keeping track of counter-arguments is important in a world of filter bubbles. *deliberate* is a tool which focuses on providing a broad overview of arguments to reduce the bias due to selective exposure, reduce insecurity about one's opinion, and possibly also change one's opinion when seeing other arguments.

A new concept in our application is pre-filtering the presented arguments using algorithms which use collaborative filtering to show arguments the user will probably accept.

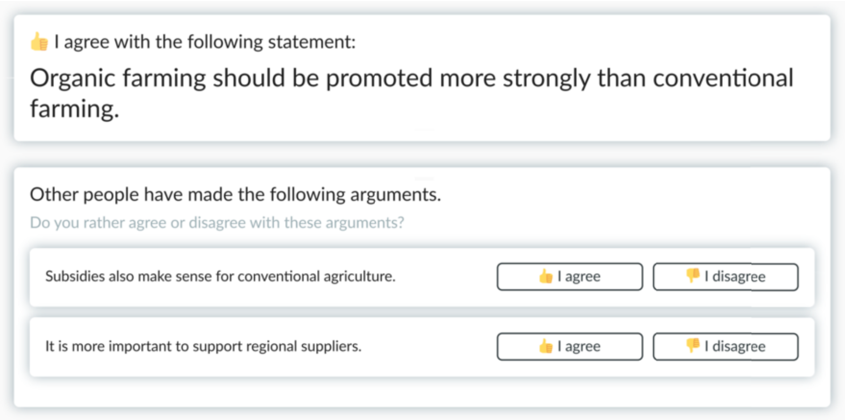
2. *deliberate* – A (Neutral?) Webapp for Exchanging Arguments

deliberate is built around a central statement which is being discussed. The user is first asked for their initial opinion on it, how sure they are about their opinion, and what their most important argument is. They can select an argument from a list of arguments already given by other users, search the database of all arguments, or add a new one, which is similar to other applications for online argumentation.

Using the collected information about the user's opinion, more pro and/or contra arguments previously provided by other users are suggested. The user can indicate that they like or dislike these arguments, sort their arguments by importance, and go deeper into the argumentation graph by selecting a statement. The argumentation graph is based on the IBIS model [2], where nodes are statements and edges are arguments, but the user has not to be aware of this theoretical background.

Unlike similar applications, every list of suggested arguments is pre-filtered using collaborative filtering, which has several advantages. The user only sees arguments which

[1] Corresponding Author: Markus Brenneis, Heinrich-Heine-Universität, Universitätsstraße 1, 40225 Düsseldorf, Germany; E-mail: markus.brenneis@uni-duesseldorf.de; member of the Manchot research group *Decision-making with the help of Artificial Intelligence*, use case politics.



👍 I agree with the following statement:

Organic farming should be promoted more strongly than conventional farming.

Other people have made the following arguments.

Do you rather agree or disagree with these arguments?

Subsidies also make sense for conventional agriculture.

It is more important to support regional suppliers.

Figure 1. Screenshot of *deliberate*, depicting confrontation with other arguments in a confrontational mode.

are more relevant for them first; thus, they have to read less text, can concentrate on personally relevant arguments, and do not have to read arguments which might be uninteresting.

The filtering uses a new pseudo-metric which takes the characteristics of an argumentation graph into account. For instance, it considers opinions for arguments deeper in the graph as less important, takes into account which arguments are used, and which arguments are rated more important for one's opinion than others. Using this metric, users which are most similar to oneself are determined, and a weighted-average of those users' opinions is calculated. The arguments which have the highest agreement in this average are displayed first.

In a currently running study, we are evaluating the effects of different filtering methods (including and excluding collaborative filtering, showing only arguments against one's own opinion, and others) on the formation of opinion and perception of neutrality.

3. Related Work

In kialo[2], users can exchange arguments in hierarchical pro/con lists, where arguments are sorted by impact, but unlike in our application, the lists are not pre-filtered or sorted based on the users' profile. The mobile application introduced in [1] uses collaborative filtering to predict the agreement of a user with a not yet rated statement; they use, however, a simpler cosine metric which does not incorporate the graph structure, and do not use it for pre-selecting the arguments displayed.

References

[1] Althuniyan, N., Sirrianni, J.W., Rahman, M.M., Liu, X.F.: Design of mobile service of intelligent large-scale cyber argumentation for analysis and prediction of collective opinions. In: International Conference on AI and Mobile Services. pp. 135–149. Springer (2019)

[2] Kunz, W., Rittel, H.W.J.: Issues as elements of information systems, vol. 131. Citeseer (1970)

[2] `https://www.kialo.com/`

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200531

An Implementation of Argument-Based Discussion Using ASPIC-

Martin CAMINADA [a,1] Sören UEBIS [b]
[a] *Cardiff University, Cardiff, UK*
[b] *FernUniversität in Hagen, Hagen, Germany*

Keywords. argument-based discussion, chatbot, ASPIC-

An often mentioned advantage of argumentation theory (compared to other formalisms for non-monotonic reasoning) is that it is based on concepts of human reasoning. However, quite some of the argumentation semantics are defined in terms of fixpoints [1] which, although appealing to mathematicians, do not seem to coincide with how most humans tend to reason in everyday life. In order to bring argument-based entailment closer to human intuitions, we propose to use formal discussion as a bridge technology. For this, we are applying argument-based discussion theory [3] which reformulates argument-based reasoning as the ability to win a particular type of discussion.[2] More specifically, an argument is in the grounded extension iff a proponent of the argument has a winning strategy in the Grounded Discussion Game [3].

In the context of abstract argumentation theory, an implementation of the Grounded Discussion Game (as well as of the Preferred Discussion Game) is already available [2]. With the current demonstrator, however, we are going one step further by basing the discussion not on abstract arguments, but on rule-based arguments that are constructed from an underlying knowledge base. For this, we base ourselves on the ASPIC- framework, which is a variant of ASPIC+ where the definition of attack is more suitable for interactive applications [4].

Our demonstrator, called ABDA (Argument-Based Discussion using ASPIC-) is written in Python3, does not require any non-standard libraries, and has been tested to work under both Windows and Linux. The knowledge base is stored in a file called `aspic-rules.txt`. The file starts with a number of strict rules (such as `a, b, c -> d`), each on its own line. After that comes a blank line, followed by a number of defeasible rules (such as `a, b, c => d [r1]` where `r1` is the name of the rule, to be used for purposes of undercutting [4]), each on its own line. These defeasible rules come in blocks consisting of several lines, which are seperated by blank lines. Defeasible rules in the same block have the same strength, whereas those in later blocks have a higher strength than those in earlier blocks. For instance, if the file contains three defeasible rules, followed by a blank

[1] Corresponding Author. Email: CaminadaM@cardiff.ac.uk

[2] One of the advantages of [3] above previous approaches (e.g. [6,5]) is that it avoids an exponential blowup in the number of moves required. We refer to [3] for details.

line, and then two other defeasible rules, then the first three rules have strength 1 and the last two rules have strength 2.

The demonstrator can be started from the command line, and takes as parameters `-wl` (to implement the *weakest link* principle [4]) or `-ll` (to implement the *last link* principle [4]), as well as `-do` (to implement the *democratic order* [4]) or `-eo` (to implement the *elitist order* [4]).

Once the demonstrator has been started, it is possible to query the inference engine if a particular statement is justified (that is, if the statement is the conclusion of an argument in the grounded extension), e.g. `warranted car_safe`. The system would then reply with either `car_safe is warranted` or `car_safe is not warranted`. The user can then ask for explanation and start a discussion with the system, e.g. `discuss car_safe`. If the statement is justified, the system will assume the role of the proponent and the user the role of the opponent. If the statement is not justified, the user will assume the role of the proponent and the system the role of the opponent. As the discussion is sound and complete for grounded semantics [3], the system is able to play a winning strategy.

At the moment, the arguments played in the game are written in a nested, machine readable way, as specified by ASPIC- (a format that is very close to ASPIC+). However, in future work we aim to be able to convert between machine readable (structured) arguments and arguments in (controlled) natural language. The overall aim is to bring human-to-computer discussion as close as possible to human-to-human discussion. For instance, when applied to the medical domain, talking to the system should resemble as much as possible talking to a more senior colleague.

The source code of ABDA, together with examples of knowledge bases, can be downloaded from `http://users.cs.cf.ac.uk/CaminadaM/demonstrators.html`

References

[1] P. Baroni, M.W.A. Caminada, and M. Giacomin. An introduction to argumentation semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.

[2] R. Booth, M.W.A. Caminada, and B. Marshall. DISCO: A web-based implementation of discussion games for grounded and preferred semantics. In S. Modgil, K. Budzynska, and J. Lawrence, editors, *Proceedings of COMMA 2018*, pages 453–454. IOS Press, 2018.

[3] M.W.A. Caminada. A discussion game for grounded semantics. In E. Black, S. Modgil, and N. Oren, editors, *Theory and Applications of Formal Argumentation (proceedings TAFA 2015)*, pages 59–73. Springer, 2015.

[4] M.W.A. Caminada, S. Modgil, and N. Oren. Preferences and unrestricted rebut. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Computational Models of Argument; Proceedings of COMMA 2014*, pages 209–220. IOS Press, 2014.

[5] S. Modgil and M.W.A. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[6] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200532

AGNN: A Deep Learning Architecture for Abstract Argumentation Semantics

Dennis CRAANDIJK [a,b], Floris BEX [b,c]
[a] *National Police Lab AI, Netherlands Police*
[b] *Department of Information and Computing Sciences, Utrecht University*
[c] *Institute for Law, Technology and Society, Tilburg University*

Keywords. argumentation semantics, neural-symbolic reasoning, graph neural networks

1. Introduction

An increasing amount of research is being directed towards designing deep learning models that learn on problems from symbolic domains [2]. One domain in symbolic AI that is relatively unexplored in this regard is *computational argumentation*. Much of the theory in computational argumentation is built on Dung's [3] work on abstract argumentation frameworks, in which he introduces acceptability semantics that define which sets of arguments (*extensions*) can be reasonably accepted given an argumentation framework (AF) of arguments and attacks (often represented as a graph). With such semantics, it can be determined if an argument can be *credulously accepted* (it is contained in some extensions) or *sceptically accepted* (it is contained in all extensions).

In [1] we propose AGNN: a deep learning approach that is able to learn to solve several core problems in abstract argumentation almost perfectly. In this demonstration we show AGNN's underlying architecture; what the model learns in order to solve an argumentation problem and how it differs from symbolic algorithms.

2. Argumentation Graph Neural Network

Most current approaches solve acceptance problems by translating the problem to a symbolic formalism for which a dedicated solver exists. AGNN (argumentation graph neural network) learns a message passing algorithm to determine sceptical and credulous acceptance and enumerate extensions under 4 well-known argumentation semantics. Experimental results demonstrate that the AGNN can almost perfectly predict acceptability and enumerate extensions, and scales well for larger argumentation frameworks (100-200 arguments).Our learning-based approach to determining argument acceptance shows that sub-symbolic deep learning techniques can accurately solve a problem that could previously only be solved by sophisticated symbolic solvers. Furthermore, analysing the behaviour of the message-passing algorithm shows that the AGNN learns to adhere to ba-

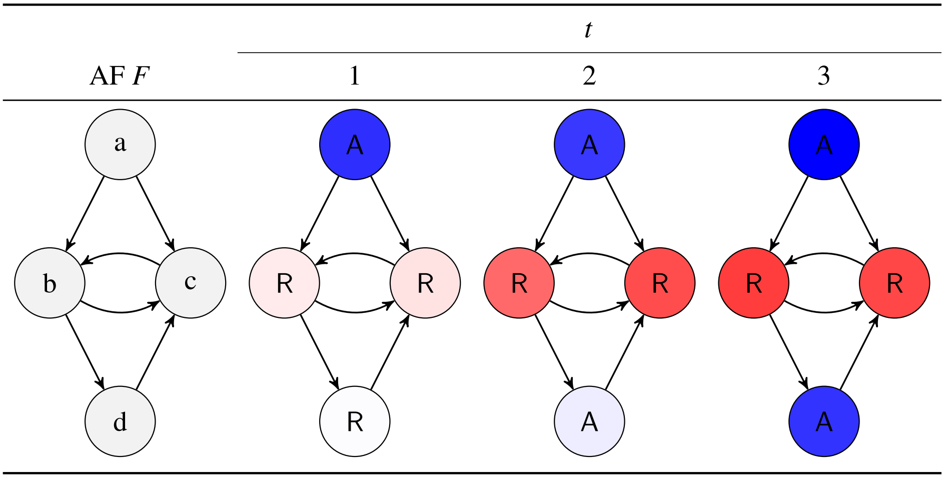


Figure 1. The acceptance predictions AGNN makes after the first three message passing iterations on the AF $F = (\{a,b,c,d\},\{(a,b),(a,c),(b,c),(b,d),(c,b),(d,c)\}$ with respect to the grounded semantics. The label and colour of each arguments denote whether the argument is predicted to be A accepted or R rejected where a darker colour indicates a higher confidence prediction (from: [1]).

sic principles of argument semantics as identified in the literature, and in the case of acceptance under the grounded semantics exhibits behaviour similar to a well-established symbolic labelling algorithm [4].

3. Demonstration

We demonstrate our Python implementation of an AGNN and show:

- how the AGNN architecture enables learning a neural message passing algorithm
- how the parameters of this algorithms can be optimised to predict argument acceptance almost perfectly under different semantics
- that the AGNN learns to adhere to basic principles of argument semantics
- how the learned algorithm differs from symbolic algorithms

We do so by graphically demonstrating AGNN's behaviour on an AF (cf. Figure 1).

References

[1] Dennis Craandijk and Floris Bex. Deep learning for abstract argumentation semantics. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, 2020. in press.

[2] Artur S. d'Avila Garcez, Tarek R. Besold, Luc De Raedt, Peter Földiák, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luís C. Lamb, Risto Miikkulainen, and Daniel L. Silver. Neural-symbolic learning and reasoning: Contributions and challenges. In *Proceedings of the 2015 AAAI Spring Symposia*, Palo Alto, 2015. Stanford University.

[3] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[4] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In Iyad Rahwan and Guillermo R Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. 2009.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200533

Navigating Arguments and Hypotheses at Scale

Rory DUTHIE [a,1], John LAWRENCE [a], Chris REED [a], Jacky VISSER [a] and Dimitra ZOGRAFISTOU [a]
[a] *Centre for Argument Technology, University of Dundee, UK*

Keywords. Hypotheses, Arguments, Navigation, Centrality

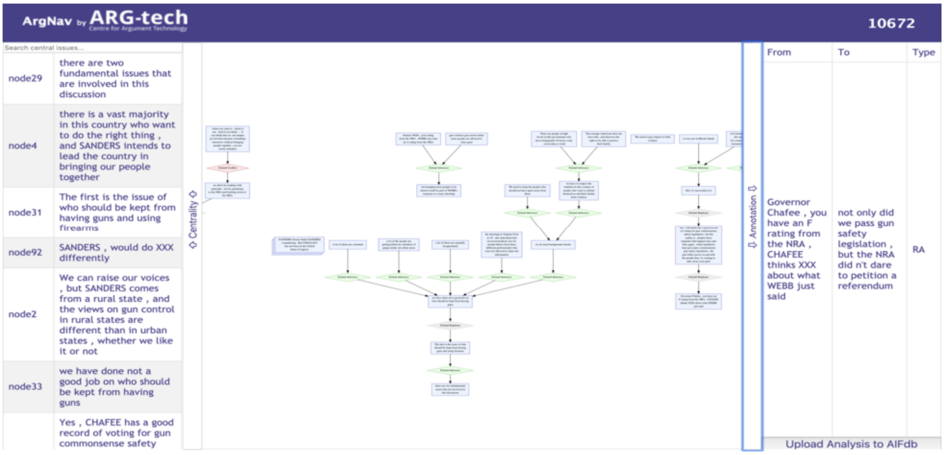
Over the past decades, freely available software for annotating and navigating argument structures have been a staple of the argumentation community. These tools have catered for two main goals: the creation of large corpora of argument; and, enhancing critical thinking and reasoning skills – with the rise in fake news sparking new research in argument technology [3]. The intelligence analysis community has focused on similar lines of research [5]. Specifically, tools are available which allow for the creation of multiple hypotheses and the extraction of evidence to support or contradict using documents from multiple sources such as news articles and social media [1]. There is also a growing demand within the field of argument mining for the creation of large datasets containing argument structures, which has so far been satisfied through crowd-sourced annotation and the construction of dispersed argument annotation teams [4].

Despite the advances in both the intelligence analysis and argument mining areas of the argumentation community, the issue remains of efficiently exploring such argument structures through visual means, and allowing the manual connection of multiple argument analyses. ArgNav[2] provides the ability to visually explore argument structures and further annotate separate analyses within AIFdb [2]. Visual exploration makes use of a combination of centrality measures, collapsing argument sub-graphs, and automatic panning and zooming, whilst annotation utilises simple point and click actions for long distance relation creation (see Figure 1 for the user interface).

A single backend technology, python, is used for the creation of ArgNav with argument structures requested from AIFdb, as either single maps or full corpora, in JSON and SVG format, and subsequently parsed using the networkx library to provide eigenvector centrality scores for propositions. Three front-end technologies (HTML, CSS and JavaScript) display SVG images of the argument structure and D3.js and Jquery allow the collapsing of sub-graphs by clicking propositions, automatic panning and zooming to propositions through clicks in the centrality panel, and annotation of intertextual and intermap correspondence [6] by clicking two nodes which provides a dialogue box for users to select an AIF relation. Finally, analyses can be saved to AIFdb using python which creates an AIF JSON structure from the selected relations. Testing on the US2016

[1] Corresponding Author: Rory Duthie, Centre for Argument Technology, University of Dundee, UK; E-mail: rduthie001@dundee.ac.uk

[2] Website available at https://argnav.arg.tech/ and code at https://github.com/roryduthie/ArgNav



node29	there are two fundamental issues that are involved in this discussion
node4	there is a vast majority in this country who want to do the right thing , and SANDERS intends to lead the country in bringing our people together
node31	The first is the issue of who should be kept from having guns and using firearms
node92	SANDERS , would do XXX differently
node2	We can raise our voices , but SANDERS comes from a rural state , and the views on gun control in rural states are different than in urban states , whether we like it or not
node33	we have done not a good job on who should be kept from having guns
	Yes , CHAFEE has a good record of voting for gun commonsense safety

From	To	Type
Governor Chafee , you have an F rating from the NRA , CHAFEE thinks XXX about what WEBB just said	not only did we pass gun safety legislation , but the NRA did n't dare to petition a referendum	RA

Figure 1. The ArgNav user interface (UI). Central issues are displayed on the left side of the UI ordered by eigenvector centrality, the large-scale argument maps are displayed in the centre of the UI through SVG, and the annotation panel on the right side of the UI shows annotated relations.

corpus in AIFdb containing 8099 propositions and 3772 conflict and support relations shows that ArgNav facilitates the efficient navigation of argument maps and corpora at large scale, in an easy to use way.

Acknowledgements

We gratefully acknowledge that this work was supported in part by the Defence Science & Technology Laboratory in the UK.

References

[1] Federico Cerutti, Timothy Norman, Alice Toniolo, and Stuart Middleton. CISpaces.org: from fact extraction to report generation. In Sanjay Modgil, Katarzyna Budzynska, John Lawrence, and Katarzyna Budzynska, editors, *Computational Models of Argument - Proceedings of COMMA 2018*, Frontiers in Artificial Intelligence and Applications, pages 269–280, Netherlands, 2018. IOS Press.

[2] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. Aifdb: Infrastructure for the argument web. In Bart Verheij, Stefan Szeider, and Stefan Woltran, editors, *Computational Models of Argument - Proceedings of COMMA 2012*, Frontiers in Artificial Intelligence and Applications, pages 515–516, Netherlands, 2012. IOS Press.

[3] John Lawrence, Jacky Visser, and Chris Reed. Bbc moral maze: Test your argument. In Sanjay Modgil, Katarzyna Budzynska, John Lawrence, and Katarzyna Budzynska, editors, *Computational Models of Argument - Proceedings of COMMA 2018*, Frontiers in Artificial Intelligence and Applications, pages 465–466, Netherlands, 2018. IOS Press.

[4] Chris Reed, Katarzyna Budzynska, John Lawrence, Martın Pereira-Farina, Dominic De Franco, Rory Duthie, Marcin Koszowy, Alison Pease, Brian Pluss, Mark Snaith, Debela Tesfaye, and Jacky Visser. Large-scale deployment of argument analytics. In *In Argumentation and Societythe workshop at the 7th International Conference on Computational Models of Argument (COMMA 2018)*, 2018.

[5] Beth M Sundheim. Third Message Understanding Evaluation and Conference (MUC-3): Phase 1 Status Report. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 301–305, USA, 1991. ACL.

[6] Jacky Visser, Rory Duthie, John Lawrence, and Chris Reed. Intertextual correspondence for integrating corpora. In *Proceedings of the 11th LREC*, pages 3511–3517, 2018.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200534

The ASPARTIX System Suite

Wolfgang DVOŘÁK [a], Sarah A. GAGGL [b] Anna RAPBERGER [a]
Johannes P. WALLNER [a] and Stefan WOLTRAN [a]
[a] *Institute of Logic and Computation, TU Wien, Austria*
[b] *TU Dresden, Germany*

1. System Description

In this system description we briefly describe the ASPARTIX system for reasoning with different abstract argumentation formalisms.

The ASPARTIX system was one of the first systems that supported efficient reasoning for a broad collection of abstract argumentation semantics starting with the work of Egly et al. [1,2] and has been continuously expanded and improved since then (see, e.g., [3,4,5]). From the very beginning the system was not limited to Dung's abstract argumentation frameworks (AFs) [6] but supported several enhancements and generalizations of AFs by, e.g., preferences or recursive attacks. Most recently, it has been extended by support for argumentation frameworks with collective attacks and claim-augmented argumentation frameworks and has been optimized for ICCMA'19 [5].

ASPARTIX is based on answer-set programming (ASP) and the idea of characterizing argumentation semantics via fixed ASP encodings. With an encoding of a semantics one can easily apply state-of-art systems for ASP to solve diverse reasoning tasks or to enumerate all extensions of a given framework. We briefly sketch the basic workflow of ASPARTIX on AFs. Given an AF in the apx format of ICCMA [7] as input, ASPARTIX delegates the main reasoning to an answer set programming solver (e.g., clingo [8]), with answer set programs encoding the argumentation semantics and reasoning tasks. The basic workflow is shown in Figure 1, i.e., the AF is given in apx format (facts in the ASP language), and the AF semantics and reasoning tasks are encoded via ASP rules, possibly utilizing further ASP language constructs. For more information on the ASPARTIX system and its derivatives in general, the interested reader is referred to the systems web-page: www.dbai.tuwien.ac.at/research/argumentation/aspartix/.

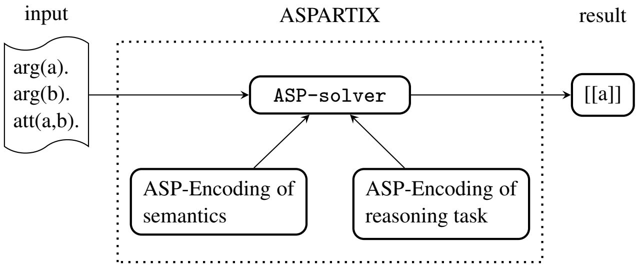


Figure 1. Basic workflow of ASPARTIX

2. Supported Argumentation Formalisms

The core of the ASPARTIX system is its support for Dung AFs [6] and a wide range of semantics, thereby facilitating enumeration of extensions as well as skeptical and credulous acceptance. On top of that there is support for several argumentation formalisms that enhance Dung AFs which are typically implemented by either combining new ASP encodings with the encodings for Dung AFs or by modifying encodings for Dung AFs to match the needs of the argumentation formalism at hand. Currently, ASPARTIX supports the following abstract argumentation formalisms: (a) Preference-based Argumentation Frameworks (PAFs) [9], (b) Value-based Argumentation Frameworks (VAFs) [10], (c) Bipolar Argumentation Frameworks [11], (d) Extended Argumentation Frameworks that allow for attacks on attacks [12], (e) Argumentation framework with recursive attacks (AFRAs) [13], (f) Argumentation framework with collective attacks (SETAFs) [14], (g) Claim-augmented argumentation frameworks (CAFs) [15].

Acknowledgments This research has been funded by DFG grant 389792660 as part of TRR 248; and by the Austrian Science Fund (FWF): P30168 and W1255-N23.

References

[1] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. ASPARTIX: implementing argumentation frameworks using answer-set programming. ICLP 2008, LNCS 5366, pages 734–738. 2008.

[2] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[3] Wolfgang Dvořák, Sarah Alice Gaggl, Thomas Linsbichler, and Johannes Peter Wallner. Reduction-based approaches to implement Modgil's extended argumentation frameworks. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation- Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday* , LNCS 9060, pages 249–264. 2015.

[4] Wolfgang Dvořák, Alexander Greßler, and Stefan Woltran. Evaluating SETAFs via answer-set programming. SAFA 2018, pages 10–21. CEUR-WS.org, 2018.

[5] Wolfgang Dvořák, Anna Rapberger, Johannes Peter Wallner, and Stefan Woltran. ASPARTIX-V19 - an answer-set programming based system for abstract argumentation. FoIKS 2020, LNCS 12012, pages 79–89. 2020.

[6] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.

[7] Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. *Artif. Intell.*, 279, 2020.

[8] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Clingo = ASP + control: Preliminary report. *CoRR*, abs/1405.3694, 2014.

[9] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Ann. Math. Artif. Intell.*, 34(1-3):197–215, 2002.

[10] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.

[11] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. ECSQARU 2005, LNCS 2571, pages 378–389. 2005.

[12] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[13] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: argumentation framework with recursive attacks. *Int. J. Approx. Reason.*, 52(1):19–37, 2011.

[14] Søren Holbech Nielsen and Simon Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. ArgMAS 2006, LNCS 4766, pages 54–73. 2006.

[15] Wolfgang Dvořák and Stefan Woltran. Complexity of abstract argumentation under a claim-centric view. *Artif. Intell.*, 285: 103290, 2020.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200535

decide: Supporting Participatory Budgeting with Online Argumentation

Björn EBBINGHAUS [a,1], Martin MAUVE [a]
[a] *Department of Computer Science, University of Düsseldorf, Germany*

Keywords. online argumentation, participatory budgeting, decision making based on argumentation

1. Introduction

With *decide* we want to enable a large crowd of participants to decide on a complex issue, such as how to make the best use of a given budget. In particular, we are interested in understanding how online argumentation and online prioritization schemes can be combined to support collective decision-making.

We have used decide to let our students collectively decide on how to use a real-world budget to improve the computer science course of study at Heinrich-Heine-University (HHU) [1]. In our demo we will show the set up used in that experiment and report on the outcome.

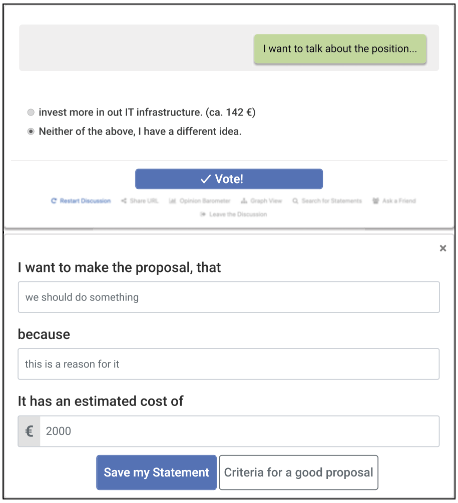
2. The *decide* collective decision system

decide employs a three-step approach to collective decision-making. In the first step participants can introduce proposals. For each proposal an estimated cost is provided by that participant. All participants then use dialog-based argumentation [2] to argue about the validity and priority of the proposals. This is shown in Figure 1.

In the second step the proposals are validated. That is to say in our specific experiment we checked if there are any reasons why any of the proposals cannot be realized even if the proposed resources were allocated to it. For example, one proposal required significant construction work which was not feasible. The remaining proposals with the attached argumentation then enter the next step.

In the final step the participants prioritize the proposals. First, the participants select the proposals that they want to support. Then they order the supported proposals by their own priority (see Figure 2). The arguments attached to the proposals can be viewed and extended in this phase, but no new proposals can be created. The final result is then calculated using a truncated Borda count followed by a greedy collection of proposals which fit the budget.

[1] Corresponding Author: Björn Ebbinghaus, Heinrich-Heine-Universität, Universitätsstraße 1, 40225 Düsseldorf, Germany; E-Mail: ebbinghaus@hhu.de; The author is a member of the PhD-programme ‘Online Participation’, supported by the North Rhine-Westphalian funding scheme ‘Forschungskollegs’.



I want to talk about the position...

○ invest more in out IT infrastructure. (ca. 142 €)

● Neither of the above, I have a different idea.

I want to make the proposal, that

we should do something

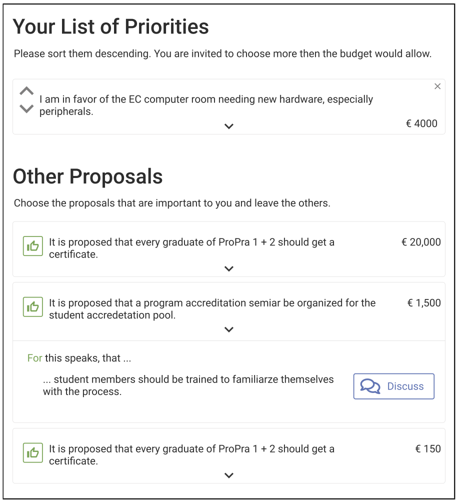
because

this is a reason for it

It has an estimated cost of

€ 2000

Figure 1. The interface that is used to enter new proposals into the dialog-based argumentation system. There also was a dedicated page for participants about what is an acceptable proposal.



Your List of Priorities

Please sort them descending. You are invited to choose more then the budget would allow.

I am in favor of the EC computer room needing new hardware, especially peripherals. € 4000

Other Proposals

Choose the proposals that are important to you and leave the others.

It is proposed that every graduate of ProPra 1 + 2 should get a certificate. € 20,000

It is proposed that a program accreditation semiar be organized for the student accredetation pool. € 1,500

For this speaks, that ...

... student members should be trained to familiarze themselves with the process.

It is proposed that every graduate of ProPra 1 + 2 should get a certificate. € 150

Figure 2. An extract from the *decide* interface. Participants accept proposals down below, and can prioritize the selected ones above. Also, the proposals can be extended to show arguments for and against it. It is also possible to jump back into the D-BAS argumentation at that point.

3. Future Work

We have received valuable feedback from the students who used *decide*. One main issue is that dialog-based argumentation tends to involve the participant in a lengthy exchange of pro and contra arguments. This is good to gain an in-depth understanding of all positions, but it makes it hard to gain a quick overview of the main points. One main issue is therefore to improve the argumentation step and also to test other approaches — such as nested pro and contra lists.

A second issue is the algorithm used to reach a decision. We would like to experiment with other voting schemes and see which of those are considered to be fair by the participants.

References

[1] Ebbinghaus, B.: Decision Making with Argumentation Graphs. Master's thesis, Department of Computer Science, Heinrich-Heine-University Düsseldorf (May 2019). https://doi.org/10.13140/RG.2.2.12515.09760/1

[2] Krauthoff, T., Meter, C., Betz, G., Baurmann, M., Mauve, M.: D-bas – a dialog-based online argumentation system. In: Modgil, S., Budzynska, K., Lawrence, J. (eds.) Computational Models of Argument. Proceedings of COMMA 2018. vol. 305, pp. 325–336. IOS Press, Amsterdam etc (2018). https://doi.org/10.3233/978-1-61499-906-5-325

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200536

Dialogical Fingerprinting of Debaters

Matt FOULIS [a,1], Jacky VISSER [a], Chris REED [a]
[a] *Centre for Argument Technology, University of Dundee, United Kingdom*

Keywords. BBC Moral Maze, Debate Analytics, Speaker profiling

In debates and dialogues, individual speakers exhibit characteristic communicative tendencies, such as using shorter or longer sentences, an idiosyncratic vocabulary, and speaking at particular times during the dialogue. Such and other characteristics underpin the emerging fields of Debating Technology and Debate Analytics [1]. As part of this goal of supporting and enhancing human debate and argument, we present *Dialogical Fingerprinting*, a system that uses these behavioural characteristics to build a unique dialogical fingerprint for the speakers within a debate. The dialogical fingerprints allow the individual identification of speakers and their roles, extending existing Debate Analytics.

The data used for the Dialogical Fingerprinting demonstrator comprises 21 transcribed episodes of The Moral Maze, a BBC Radio 4 programme about ethically divisive or controversial issues. The participants in each 43-minute episode are: the moderator, present in every episode; four panellists, drawn from a small pool of regulars; and three guest witnesses, who are experts on the topic debated in that episode. This leads to a total of 93 individual speakers within the dataset, with an unbalanced distribution. A subset of the transcripts is annotated on the basis of Inference Anchoring Theory, explicitly indicating discourse segments, communicative functions, and argumentative structure [2].

Using scitkit-learn and Keras (with TensorFlow back-end), Dialogical Fingerprinting models the behavioural characteristics of a speaker as features in a machine learning approach. Within a Moral Maze episode, each participant fulfils a distinct debating role (moderator, panellist, or witness). Each role is associated with a characteristic dialogical fingerprint common to all individual speakers in that particular role. We use this common dialogical fingerprint to automatically classify the participants' debating roles. Similarly, by using behavioural characteristics to build a unique dialogical fingerprint for each individual speaker, we are able to automatically classify a participant's identity.

Upon testing various machine learning models, a Support Vector Machine (SVM) outperformed the other techniques. When training on 20 episodes and testing on one full episode, the initial role classification resulted in a macro F1 of 0.75. After implementing a post-processing rule reflecting the fact that any speaker only ever performed one particular role within a single Moral Maze episode, the macro F1 reached 1.0. The results for the much harder task of identifying the individual speaker on the basis of their contributions to the debate yielded a macro F1 of 0.52, again using an SVM model. While this appears to be a modest performance, we have to take into account that this result was achieved when testing on a previously unseen episode, in which three of the speakers are unique to that episode and therefore absent from the training data.

[1] Corresponding author; e-mail: mzfoulis@dundee.ac.uk

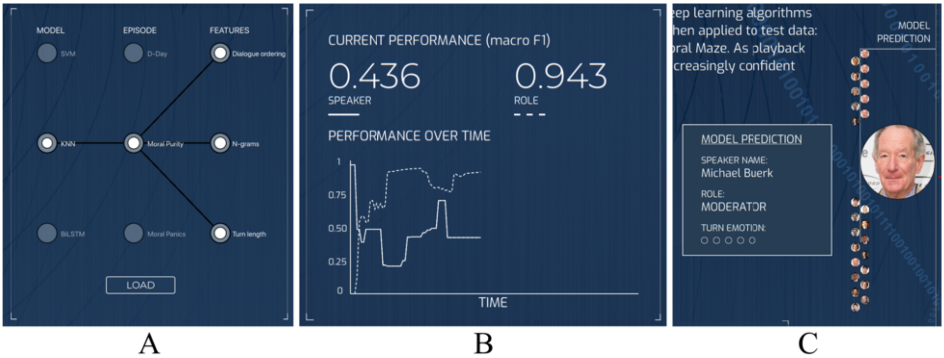


Figure 1. Elements of the interactive demonstrator: (A) selection of model, data, and features; (B) time-based graph indicating model performance, the speaker and role classification are shown as 0.436 and 0.943 respectively at the current point in the episode; and (C) model predictions for selected speaker.

In addition to classifying individual speakers and their roles, we automatically analysed the relative emotionality and ideological scaling of debaters. Using the subjectivity lexicon developed by Wilson et al. [3], we counted occurrences of lexemes that carry a positive or negative sentiment, and cast these values to a 0 to 1 interval. To group debaters in a Moral Maze episode on the basis of ideological scaling, we adopted the unsupervised methods developed by Glavas et al. [4], comparing the language used by the speakers to data for which the ideological orientation is known, such as political manifestos.

To demonstrate the software, we developed a touch-based graphical user interface (GUI), giving non-experts an intuitive way to interact with the underlying machine learning models and Debate Analytics. The GUI prompts users to choose a machine learning algorithm, an episode to test the performance on, and the dialogical features to be included (Figure 1A). Using this selection, the software demonstrates the gradual improvement of the speaker and role classifiers as the episode progresses (Figure 1B), the emotionality of the speakers' turns (Figure 1C), and their ideological scaling.

Dialogical fingerprinting demonstrates that it is possible to identify participants of a debate based not only on *what* they say, but *how* they say it, opening up new areas of research in person identification within dialogue and debates.

Acknowledgements. This work was supported by Dstl under DASA grant ACC6005123.

References

[1] J. Lawrence, M. Snaith, B. Konat, K. Budzynska, and C. Reed, "Debating technology for dialogical argument: sensemaking, engagement, and analytics," *ACM Trans. Internet Technol.*, vol. 17, no. 3, 2017.

[2] C. Reed, S. Wells, K. Budzynska, and J. Devereux, "Building arguments with argumentation: the role of illocutionary force in computational models of argument," in *Computational Models of Argument. Proceedings of COMMA 2010* (P. Baroni, F. Cerutti, M. Giacomin, and G. R. Simari, eds.), (Amsterdam), pp. 415–426, IOS Press, 2010.

[3] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of HLT and EMNLP*, p. 347–354, ACL, 2005.

[4] G. Glavas, F. Nanni, and S. P. Ponzetto, "Unsupervised cross-lingual scaling of political texts," in *Proceedings of the 15th EACL*, pp. 688–693, ACL, apr 2017.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200537

PEOPLES: From Private Responses to Messages to Depolarisation Nudges in Two-Party Adversarial Online Talk

Iwan ITTERMANN [b] and Brian PLÜSS [a]
[a] *Digital Peace Talks gUG (h.b.), Germany*
[b] *Centre for Argument Technology, University of Dundee, UK*

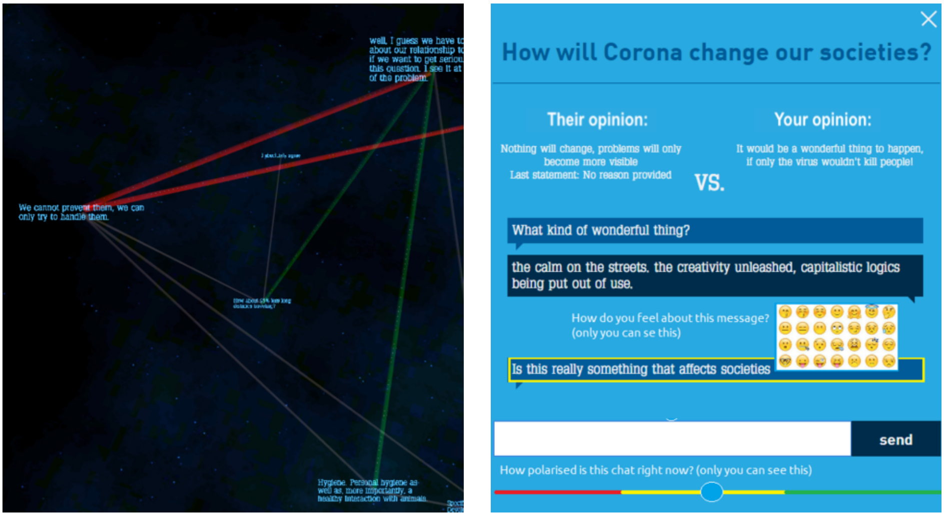
Keywords. Affective Polarisation, Dialogue Modelling, Nudging

The PEOPLES (Private Expression of Polarisation Leveraged to Expand Sociability) Project envisages a fine grained, language-independent measure of affective polarisation between participants in two-party chats over controversial topics. The ultimate goal of the project is to channel the analytical power of the measure to enable automated real-time interventions, nudging participants towards healthier conversational behaviours.

We hypothesise that this measure can be derived solely from the unique profiles of each conversational participant's private reactions (akin to emoji responses on mainstream social media) to the messages they receive in two-party chats. Aided by the language-independence of the approach, we intend to base and evaluate the measure on empirical evidence, by studying polarised users from several cultural contexts, both Western and non-Western.

So far, much emphasis has been on text classification to detect hate speech [1,2], profanity [3] and incivility [4], or on sentiment analysis and psychometric measuring to identify influential factors on political polarisation in deliberative spaces and networks [5,6,7]. Both approaches have limitations when it comes to developing helpful automated interventions at scale. The former assumes uniform reactions across all participants and is thereby prone to have discriminatory effects on minorities, while depending on substantial, costly training datasets. The latter is descriptive, assuming polarisation to be the effect of actions (e.g. news consumption, media use) or connectivity (network popularity, group contact), thus offering little insight for effective automated interventions.

To the best of our knowledge, researchers have not previously employed opinion polarisation analysis based on two-party private communication such as chats online. One of the main reasons for this is the scarcity of natural data publicly available, due to privacy constraints. DPT (`demo.dpt.world`) offers an uncommon opportunity to access such data: it publishes and structures two-party discussions between opinion postings in a signed graph (see Figure 1a). Conceptually, it is comparable to ChangeAView (`changeaview.com`), with the difference that users are required to post their opinions regarding a given topic before they can take part in one-to-one discussions. Chat messages are published after three days. Users can continuously rate the chat's degree of polarisation. The averaged ratings of both users determine the weight of the edge connecting both postings in the graph. In a new feature, participants of a chat will be able


well, I guess we have to
about our relationship to
if we want to get seriou
this question. I see it at
of the problem.
We cannot prevent them, we can
only try to handle them.
Hygiene. Personal hygiene as
well as, more importantly, a
healthy interaction with animals.
×
send

How will Corona change our societies?

Their opinion:

Nothing will change, problems will only become more visible
Last statement: No reason provided

VS.

Your opinion:

It would be a wonderful thing to happen, if only the virus wouldn't kill people!

What kind of wonderful thing?

the calm on the streets, the creativity unleashed, capitalistic logics being put out of use.

How do you feel about this message?
(only you can se this)

Is this really something that affects societies

How polarised is this chat right now? (only you can see this)

(a) Graph of opinions (nodes) and chats between posters (edges).

(b) DPT chat interface with emoji reaction to messages and polarisation rating.

Figure 1. The PEOPLES-DPT system

to click on icons (comparable to emoji reactions in Messenger, only they are not visible to the other party during the conversation) to privately record their reaction to a specific message (see Figure 1b).

The exploration of a sender-receiver aware polarisation measure, as well as receiver aware nudge-style interventions, is aimed at advancing the understanding of the role of messengers in affective opinion polarisation, and at laying ground for depolarisation technologies to gain momentum.

References

[1] S. Akhtar, V. Basile and V. Patti, A New Measure of Polarization in the Annotation of Hate Speech, in: *AI*IA 2019 – Advances in Artificial Intelligence*, M. Alviano, G. Greco and F. Scarcello, eds, Springer International Publishing, Cham, 2019, pp. 588–603. ISBN ISBN 978-3-030-35166-3.

[2] P. Fortuna and S. Nunes, A survey on automatic detection of hate speech in text, *ACM Computing Surveys (CSUR)* **51**(4) (2018), 1–30.

[3] P.L. Teh, C.-B. Cheng and W.M. Chee, Identifying and Categorising Profane Words in Hate Speech, in: *Proceedings of the 2nd International Conference on Compute and Data Analysis*, ICCDA 2018, Association for Computing Machinery, New York, NY, USA, 2018, pp. 65–69. https://doi.org/10.1145/3193077.3193078.

[4] T. Hopp, C.J. Vargo, L. Dixon and N. Thain, Correlating self-report and trace data measures of incivility: A proof of concept, *Social Science Computer Review* (2018), 0894439318814241.

[5] J.N. Druckman and M.S. Levendusky, What do we measure when we measure affective polarization?, *Public Opinion Quarterly* **83**(1) (2019), 114–122.

[6] J. Yang, H. Rojas, M. Wojcieszak, T. Aalberg, S. Coen, J. Curran, K. Hayashi, S. Iyengar, P.K. Jones, G. Mazzoleni et al., Why are "others" so polarized? Perceived political polarization and media use in 10 countries, *Journal of Computer-Mediated Communication* **21**(5) (2016), 349–367.

[7] C.A. Bail, L.P. Argyle, T.W. Brown, J.P. Bumpus, H. Chen, M.F. Hunzaker, J. Lee, M. Mann, F. Merhout and A. Volfovsky, Exposure to opposing views on social media can increase political polarization, *Proceedings of the National Academy of Sciences* **115**(37) (2018), 9216–9221.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200538

ArgAgent: A Simulator of Goal Processing for Argumentative Agents

Henrique M. R. JASINSKI [a], Mariela MORVELI-ESPINOZA [a] and Cesar A. TACLA [a]
[a] *Graduate Program in Electrical and Computer Engineering (CPGEI), Federal University of Technology - Paraná (UTFPR), Curitiba, Brazil*

1. Introduction

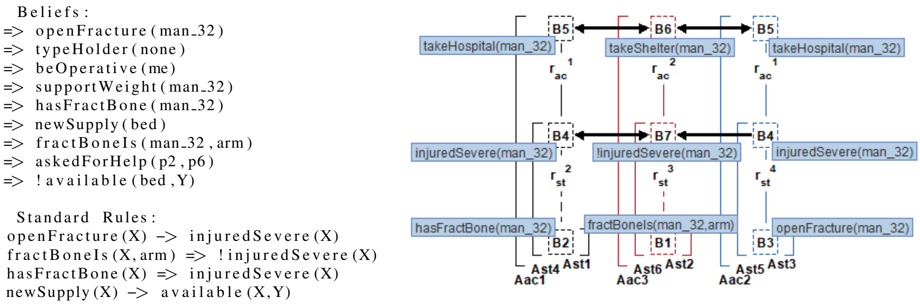
Since multiagent systems are intrinsically distributed, debugging and explaining their behaviour poses a especial challenge. In [1] a new abstract model for intelligent agents is presented, called Belief-Based Goal Processing (BBGP), which is different from the Belief-Desire-Intention (BDI) model [2] mainly due to the "goal processing" responsible for choosing which goals should be pursued. In [3] and [4] a computational formalization of the BBGP is presented which uses argumentation for the goal processing reasoning. We call this model Argumentative-BBGP. The developed simulator is a tool that allows an Argumentative-BBGP agent to be executed and to inspect its decision process.

ArgArgent[1] was developed using Java. Two libraries, and its dependencies, from the TweetyProject [5] were used. The first-order logic module was used to represent the basic elements of the model, such as the beliefs and goals. The ASPIC argumentation module was used in the goal processing for non-monotonic reasoning. The focus of the simulator is primarily the "goal processing". Each goal may be in one of the following states: **active**, **pursuable**, **chosen**, **executive**, **completed**, or **canceled**, in that order, where necessarily a goal must have attained the previous states, with the exception of canceled state, which can be achieved from any state. The goal processing comprises four well defined stages: I) **activation**, which instantiates goals based on the agents current beliefs; II) **evaluation**, which identifies and evaluates obstacles for pursuing active goals; III) **deliberation**, which identifies the associated plans for each pursuable goal, evaluates conflicts among pursuable, chosen, and executive goals, and determines which pursuable goals should become chosen; and IV) **checking**, which evaluates whether the conditions to execute the plan for every chosen goal hold.

To start the simulation, a file containing the agent's initial beliefs, rules, the set of plans, and the preference total order on the goals is required. Each rule must be either strict or defeasible. It is also possible to load a perception file containing the perception itself and the simulation cycle in which they occur. Once the simulation begins, it is possible to inspect the current agent beliefs and the perceptions that it receives at a given cycle. It is also possible to inspect the goals memory, which describes when a given

[1] ArgAgent can be found at www.github.com/henriquermonteiro/BBGP-Agent-Simulator.

Acknowledgement This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 - and the Centro Nacional de Pesquisa - Brasil (CNPq).



```
  Beliefs:
=> openFracture(man_32)
=> typeHolder(none)
=> beOperative(me)
=> supportWeight(man_32)
=> hasFractBone(man_32)
=> newSupply(bed)
=> fractBoneIs(man_32,arm)
=> askedForHelp(p2,p6)
=> !available(bed,Y)

  Standard Rules:
openFracture(X) -> injuredSevere(X)
fractBoneIs(X,arm) => !injuredSevere(X)
hasFractBone(X) => injuredSevere(X)
newSupply(X) -> available(X,Y)
```

(a) Belief inspector (b) Goal memory inspector

Figure 1. (a) shows the beliefs and some rules of the rescue agent, where '–>' and '=>' represent strict and defeasible rules, respectively. (b) shows the interactive diagram of a goal memory entry. Beliefs, goals, and rules receive an identifier to make the diagram more readable, but a tooltip with the full description is available. Arguments receive their own identifier as well, and each bracket indicates an argument. The arrows represent attacks among arguments. The argument in blue is the selected one. Arguments in red are the rejected ones, and the ones in black are accepted arguments that defend the selected one.

goal attained a status and the beliefs that supported such decision. Figure 1 shows an example of a rescue agent, which must decide whether to send *man_32* to the hospital or to the shelter. Figure 1(b) shows the reasoning process which led to the decision of taking *man_32* to the hospital ('Aac2'), since he had an open fracture, which in turn is a severe injure. Such decision took place because the rule $openFracture(x) \rightarrow injuredSevere(x)$ is strict.

We plan to improve the simulator by implementing the mechanism for changing a goal state towards a previous state and cancellation. Our aim is to create an agent model capable of explaining in more details his decision process compared with similar approaches.

References

[1] Castelfranchi, C., Paglieri, F. The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. Synthese, v. 155, n. 2, p. 237–263, 2007.

[2] Rao, A., Georgeff, M. BDI Agents: From Theory to Practice. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), p. 312-319, 1995.

[3] Morveli-Espinoza, M., Possebom, A. T., Puyol-Gruart, J., Tacla, C. A. Argumentation-based intention formation process. Journal of the National University of Colombia (DYNA), 86(208), pp. 82-91, January - March, 2019.

[4] Morveli-Espinoza, M., Nieves, J. C., Possebom, A., Puyol-Gruart, J. and Tacla, C. A. An argumentation-based approach for identifying and dealing with incompatibilities among procedural goals. International Journal of Approximate Reasoning, v. 105, p. 1–26. Elsevier Inc, 2019.

[5] Thimm, M. Tweety: A comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation. Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning, p. 528–537, 2014.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200539

Implementing Argument and Explanation Schemes in Dialogue

Isabel SASSOON [a,1], Nadin KOKCIYAN [b], Martin CHAPMAN [c], Elizabeth SKLAR [e], Vasa CURCIN [c], Sanjay MODGIL [c], and Simon PARSONS [d]

[a] *Department of Computer Science, Brunel University London*
[b] *School of Informatics, University of Edinburgh*
[c] *Department of Informatics, King's College London*
[d] *School of Computer Science, University of Lincoln*
[e] *Lincoln Institute for Agri-Food Technology,University of Lincoln*

Keywords. Argumentation Schemes, Explanation, Dialogue

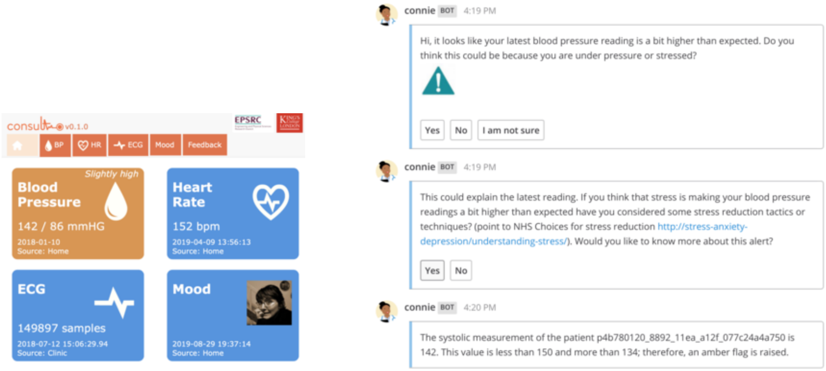
In this demo paper we outline the implementation of argumentation schemes within the CONSULT mobile application [1]. We illustrate it through a specialised argumentation scheme that supports the generation of Blood Pressure (BP) alerts within the CONSULT self management process. The scheme not only creates alerts when required but also supports the explanation of the alert to the user. The thresholds that dictate whether a user should be alerted about their BP reading are outlined in NICE guidelines CG127 [3]. The approach to structuring the explanation templates is based on our previous work [2,5]. This was part of the CONSULT mobile application version that was piloted in January 2020 in a 7 day pilot study involving 6 healthy volunteers.

Table 1. Argument scheme for blood pressure measurements

AS for BP
premise - If mean blood pressure M is higher than 140, High Blood Pressure can be inferred
premise - M is higher than 140
therefore : High blood pressure (*hbp*) is inferred

The argument scheme and dialogue implementation. This demo shows how a new BP measurement taken by the user is processed, and an alert is triggered depending on the value. This processing involves the instantiation of an argumentation scheme, ASBP [4], as outlined in Table 1. Depending on the instantiation, an alert may or may not be generated. For example, an explanation for an amber alert is constructed according to the explanation template e_1, represented as $e_1 = \langle$*ASBP, "The systolic measurement of the patient* $\{P\}$ *is* $\{S\}$*, this value is less than 150 and more than 134 and therefore an Amber flag is raised."*$\rangle$. The textual explanation includes variables (P and S) shown in brackets, which are the patient id and the systolic BP respectively. These variables will be replaced by actual values as a result of the instantiation of the ASBP scheme.

[1] Corresponding Author: Department of Computer Science, Brunel University London, United Kingdom.; E-mail: isabel.sassoon@brunel.ac.uk



connie BOT 4:19 PM

Hi, it looks like your latest blood pressure reading is a bit higher than expected. Do you think this could be because you are under pressure or stressed?

Yes | No | I am not sure

connie BOT 4:19 PM

This could explain the latest reading. If you think that stress is making your blood pressure readings a bit higher than expected have you considered some stress reduction tactics or techniques? (point to NHS Choices for stress reduction http://stress-anxiety-depression/understanding-stress/). Would you like to know more about this alert?

Yes | No

connie BOT 4:20 PM

The systolic measurement of the patient p4b780120_8892_11ea_a12f_077c24a4a750 is 142. This value is less than 150 and more than 134; therefore, an amber flag is raised.

(a) Dashboard Alert (b) Chatbot interaction

Figure 1. Alert in the CONSULT Dashboard and the Alert Dialogue in the Chatbot

If an argument in support of an alert is generated, this is seen by the user as an 'Amber' or 'Red' shading of the BP box as depicted in Figure 1a. Furthermore, this alert is written out in detail in the BP specific tab of the CONSULT mobile application as a graph. This alert also triggers a new dialogue in the CONSULT chatbot, where a textual explanation about the alerts is provided (see Figure 1b).

Scenario. The screenshots illustrate a scenario in which a user's latest systolic BP measurement is 142. This is considered as an Amber alert for Stage I Hypertension. In this case, the instantiation of ASBP results in an argument inferring high blood pressure. Then the argumentation engine instantiates the corresponding explanation template e_1 and constructs the following explanation: "The systolic measurement of the patient is 142, this value is less than 150 and more than 134 and therefore an Amber flag is raised". This explanation is displayed as part of the dialogue when the user enquires as to why this alert has been raised by interacting with the CONSULT chatbot (Figure 1b).

Acknowledgements. This research was supported by the UK Engineering & Physical Sciences Research Council (EPSRC) under grant #EP/P010105/1.

References

[1] N. Kökciyan, M. Chapman, P. Balatsoukas, I. Sassoon, K. Essers, M. Ashworth, V. Curcin, S. Modgil, S. Parsons, and E. Sklar. A collaborative decision support tool for managing chronic conditions. In *MEDINFO 2019: Health and Wellbeing e-Networks for All*, volume 264, pages 644–648, 2019.

[2] N. Kökciyan, S. Parsons, I. Sassoon, E. Sklar, and S. Modgil. An argumentation-based approach to generate domain-specific explanations. In *European Conference on Multiagent Systems*, 2020. in press.

[3] National Institute for Health and Care Excellence (NICE). Hypertension in adults: diagnosis and management cg127, 2011. https://www.nice.org.uk/guidance/cg127.

[4] I. Sassoon, N. Kokciyan, S. Parsons, and E. Sklar. Towards the use of commitments in multi-agent decision support systems. In *The International Workshop on Dialogue, Explanation and Argumentation in Human-Agent Interaction (DEXAHAI)*, 12 2018.

[5] I. Sassoon, N. Kökciyan, E. Sklar, and S. Parsons. Explainable argumentation for wellness consultation. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 186–202. Springer, 2019.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200540

A Modular Platform for Argument and Dialogue

Mark SNAITH [1], John LAWRENCE, Alison PEASE and Chris REED
Centre for Argument Technology, University of Dundee, UK

Keywords. dialogue, dialogue games, dialogue execution

1. Introduction

The Dialogue Game Execution Platform (DGEP) [1] allows human and virtual participants to engage in a structured dialogue following a specified protocol. In this short abstract, we present PAD: an open-source platform for argument and dialogue that builds on DGEP by wrapping it in a modular architecture that allows new functionality to be easily added. We introduce one such module, the *Dialogue Utterance Generator* (DUG), which finds propositional content to populate the abstract move types provided by DGEP.

2. Platform description

The **Dialogue Game Execution Platform (DGEP)** forms the core of PAD. Its function is to keep a record of a dialogue, accepting played moves and generating the resultant dialogue state including the current speaker and next available moves.

Game descriptions are written in a revised and updated version of the Dialogue Game Description Language (DGDL) [2]. Output from DGEP is the available legal move *types* based on the protocol being followed, without any consideration for the propositional content of those moves. DGEP provides a template that should subsequently be filled if the move type is selected; for instance, the template for an “argue” move may be:

```
{"reply":{"p":"go_to_cinema", "q":"$q"}}
```

After DGEP has generated a set of available move types, they can be passed to the **Dialogue Utterance Generator (DUG)**, which takes the available move types and attempts to find propositional content to instantiate them into concrete moves. The DUG itself follows a modular design that allows different sources of content to be used, even within the same dialogue.

Core to the DUG are a set of *content descriptors* and associated *content locators*. A content descriptor is linked to the move type and template provided by DGEP and

[1] Corresponding author. E-mail: m.snaith@dundee.ac.uk.

describes how variables in the "reply" object should be populated. A content locator provides an implementation of an algorithm that actually finds the content. As a concrete example, a content locator that queries a MySQL database will have the following content descriptor for an "argue" move type:

```
argue{
 @mysql("SELECT premise FROM arguments WHERE conclusion='$p';");
}
```

Continuing the previous example, this query would be instantiated with the content of "p" (i.e. "...WHERE conclusion='go_to_cinema';"). The query is then passed to the content locator for querying MySQL databases, with the result being assigned to "q" in the reply. The reply is then made available to a user or agent as a concrete move that can be played in the dialogue. If the content locator returns multiple values, a concrete move is created for each piece of content.

Other potential sources of content include AIFdb [3] (which in turn can allow a participant to contribute to a past argument or debate), argument mining [4], or a logical representation such as ASPIC+ [5]. It is however possible, in principle, for content to be obtained from any queriable source.

3. Using the platform

The source code for DGEP and the DUG is available at `https://github.com/arg-tech`. Both are also available as web services at `https://ws.arg.tech`.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. The authors are grateful to colleagues on the project for their feedback during the design and implementation.

References

[1] Bex F, Lawrence J, Reed C. Generalising argument dialogue with the Dialogue Game Execution Platform. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Fifth International Conference on Computational Models of Argument (COMMA 2014). IOS Press; 2014. p. 141–152.

[2] Wells S, Reed CA. A domain specific language for describing diverse systems of dialogue. Journal of Applied Logic. 2012;10(4):309–329.

[3] Lawrence J, Bex F, Reed C, Snaith M. AIFdb: infrastructure for the argument web. In: Verheij B, Szeider S, Woltran S, editors. Fourth International Conference on Computational Models of Argument (COMMA 2012). IOS Press; 2012. p. 515–516.

[4] Lawrence J, Reed C. Argument Mining: A Survey. Computational Linguistics. 2019;45(4):765–818.

[5] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93–124.

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200541

The Open Argumentation PLatform (OAPL)

Simon WELLS [a]

[a] *School of Computing, Edinburgh Napier University, Scotland, U.K.*

Keywords. Argument Analysis, Open Source Tools, Argument Visualisation

1. Introduction

The Open Argumentation Platform (OAPL[1]) pronounced "*opal*" is a suite of argumentation software that includes APIs, libraries, and user interfaces that work together to support a range of argument-oriented computational tasks and associated pipelines. OAPL is an open platform that is built around a suite of open-development, free-software tools, released under a permissive license. By developing and promoting open standards, the goal is to develop sustainable argumentation software, that finds real world uptake beyond the argumentation theory community, and which can act as a flexible framework for investigating new, and extending existing, techniques in argument analysis, processing, visualisation, and reuse.

The tools that make up the platform are designed to support a range of argument-centric activities such as reasoning over argument resources, dialogically-oriented interaction, manual argument analysis, and automated argument analysis. The suite currently comprises seven software tools which can be combined and configured to form a variety of argument pipelines. These tools all aim to have the following:

- A simple but extensible underlying data model.
- Clear extension points for domain specific analysis & representation tasks.
- Tooling to support import from other formats, e.g. AML, AIF, &c.
- An open source canonical implementation.
- Supporting Documentation.
- Liberal (GPL3) licensing.
- A completely open development model including public GIT repository & public issue/bug tracking.

2. Tools

OAPL currently comprises the following software tools:

[1] `http://www.openargumentation.org`

SADFAce[2] is a simple JSON based Argument description format, software library, and supporting tools to enable developers and researchers to describe arguments and to easily reuse their data. The goal is to make it as easy as possible to incorporate argument data into modern software. SADFace has a simple but extensible model that is compatible with AIF and can serialise other formats such as AML. There are both Python 3 and JavaScript implementations of the core SADFace format as well as supporting tools for SADFace document creation, editing, and manipulation. SADFace currently forms the core of OAPL and is the *lingua franca* that underpins the integration of the other tools.

ArgDB[3] is the main Argument Database that provides persistence of SADFace documents. This is a CouchDB based datastore which natively stores SADFace JSON documents. A couch-app is used to provide a web search interface. ArgDB is designed to run in either private/local mode or as a public argument data server as part of the argument web.

MonkeyPuzzle[4] [1] is a browser-based user interface for manual argument analysis. The interface is centered around a resource pane that holds the resource being analysed and a visual workspace in which a graph based argument visualisation is constructed.

ALIAS[5] [2] is “A Library for Implementing Argumentation Systems” a Python implementation of a library for working with Dung frameworks. This is currently used to provide a mechanism for automated reasoning over argument resources, for example, from SADFace documents.

DGDL[6] & ADAMANT[7] are the Dialogue Game Description Language [3] and its associated Python-based dialogue game execution platform. These two technologies work together to enable dialogue games to be specified, run, and managed.

Canary[8] is the most recent addition to OAPL, an argument mining library that is currently under heavy development. Canary is a Python library that builds on existing natural language toolkits

Many of these tools were developed independently so the path to closer integration is ongoing and non-exclusive. Contributors, developers, bug-fixers, and users are all welcomed to the OAPL community.

References

[1] J. Douglas and S. Wells. Monkeypuzzle. In *Proceedings of the 17th International Workshop on Computational Models of Natural Argument (CMNA17)*, 2017.

[2] S. Wells and R. La Greca. Introducing alias. In *Proceedings of the 15th International Workshop on Computational Models of Natural Argument (CMNA15)*, 2015.

[3] S. Wells and C. Reed. A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic*, 10(4):309–329, 2012.

[2] `https://github.com/Open-Argumentation/SADFace`

[3] `https://github.com/Open-Argumentation/ArgDB`

[4] `https://github.com/Open-Argumentation/MonkeyPuzzle`

[5] `https://github.com/Open-Argumentation/ALIAS`

[6] `https://github.com/Open-Argumentation/DGDL`

[7] `https://github.com/Open-Argumentation/ADAMANT`

[8] `https://github.com/Open-Argumentation/Canary`

Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200542

Neva – Extension Visualization for Argumentation Frameworks

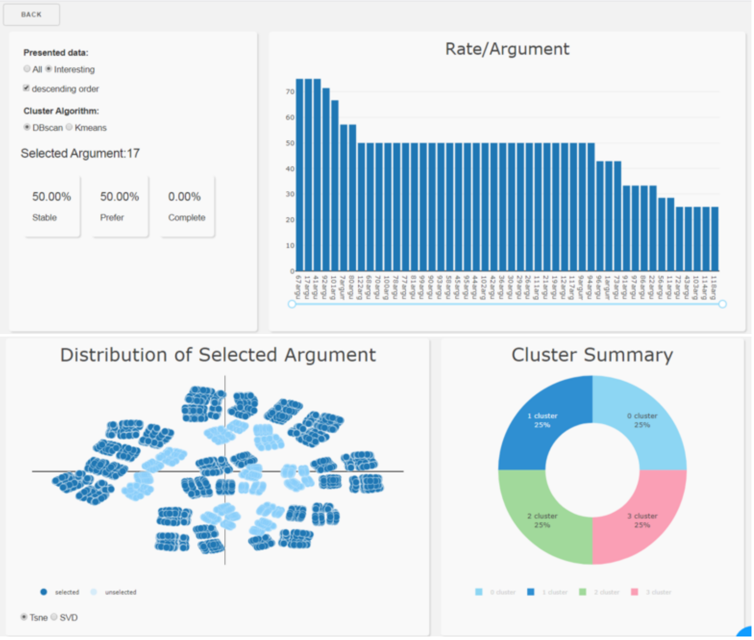
Mei YANG, Sarah Alice GAGGL and Sebastian RUDOLPH
Computational Logic Group, Technische Universität Dresden, Germany

Many combinatorial search problems, such as finding all extensions of an argumentation framework (AF) for a semantics, result in a large solution space. Nowadays, systems compute these solutions very efficiently [1]. However, the enormous number of answers is very difficult to cope with by users. Recently, Rudolph et al. [2] proposed a framework for *faceted answer set navigation* where, given an answer set program, atoms can be interactively selected or excluded in order to navigate towards desired answer sets.

A standard way to visualize argumentation extensions is to highlight accepted arguments in the argumentation framework, but this method only allows to represent one solution at a time. Interesting insights of a solution set, such as which arguments usually are accepted together, or which never appear in the same extension (under a given semantics), can only be answered by further processing the solution sets.

The system `Neva` follows a novel approach in the visualization and analysis of argumentation extensions. Based on data mining algorithms, `Neva` identifies inner patterns in the solution space, and helps users to find the interesting attributes for further investigation. Within `Neva`, answer sets are conceived as data points in a high-dimensional space, which are projected to a plane for visualizing their distribution. The input for `Neva` is a set of answer sets as produced by the system `aspartix` [3] with the ASP solver `clingo` [4], i.e. sets of answer sets with predicates $\texttt{in}(a_i)$ for all arguments a_i to be in the extension of a given semantics. Additionally, the AF in `.apx` format is required. In the data process, datasets are transformed into numerical representations by a one-hot-encoding. Then, using Euclidean distance, the system provides the options to cluster via DBscan [5] or Kmeans. `Neva` has a variety of functions for different analysis requirements. The main interface of the system shows the data distribution in the whole space and the feature attributes w.r.t. different clusters and semantics separately. In addition, there are two buttons that can trigger argument-centered analysis and argument correlation analysis (i.e. correlation matrix and its clustering). If users want to analyze their own data, an upload component is provided at bottom on this page.

For first tests on the system `Neva` we used the benchmarks from ICCMA-2017 [1]. Figure 1 presents an interactive interface that illustrates the argument occurrences in the whole answer set space and analyzes answer sets that contain the selected argument. On the upper panel, options appear on the left and can be used to define the form of the bar plot on the right. These occurrence rates mean the percentages of answer sets in the whole dataset that contain the selected argument in question. Secondly, the radio items can decide if all the arguments will be included in the bar chart. Here, the option ”interesting” focuses on those arguments whose occurrence rates are neither 100 % nor 0%. Below this, there is a check box that can control the order of attribute bars in the right graph. After finishing these decisions, the bar chart is created and users can select a specific argument by clicking on the bars. On the lower panel, the left picture shows the distribution of those answer sets containing the selected argument in two-dimensional space, while the right pie plot shows how they distribute over the clusters.



Presented data:
○ All ● Interesting
☑ descending order

Cluster Algorithm:
● DBscan ○ Kmeans

Selected Argument:17

50.00%	50.00%	0.00%
Stable	Prefer	Complete

● Tsne ○ SVD

Figure 1. Attribute Analysis

The source code of our system is freely available at `https://github.com/Lexise/ASP-Analysis` and the online `Neva` is provided at `https://asp-analysis.herokuapp.com/`. It might be updated in the future as our research continues.

Acknowledgements. This research has been funded by DFG grant 389792660 as part of TRR 248 (see `https://perspicuous-computing.science`).

References

[1] Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. *Artificial Intelligence*, 279, February 2020.

[2] Christian Al-Rabaa, Sebastian Rudolph, and Lukas Schweizer. Faceted answer-set navigation. In *Proceedings of the Second International Joint Conference on Rules and Reasoning - RuleML+RR2018,*, LNCS, pages 211–225. Springer, 2018.

[3] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[4] Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Thomas Schneider. Potassco: The Potsdam answer set solving collection. *AI Commun.*, 24(2):107–124, 2011.

[5] Krzysztof J Cios, Witold Pedrycz, and Roman W Swiniarski. Data mining and knowledge discovery. In *Data mining methods for knowledge discovery*, pages 1–26. Springer, 1998.

Subject Index

Computational Models of Argument
H. Prakken et al. (Eds.)

Author Index