Argument & Computation 13 (2022) 3–47
DOI 10.3233/AAC-200536
IOS Press

A logic of defeasible argumentation: Constructing arguments in justification logic

Stipe Pandžić
Department of Theoretical Philosophy, Faculty of Philosophy & Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Faculty of Science and Engineering, University of Groningen, The Netherlands
E-mail: stipepandzic@yahoo.com

Abstract. In the 1980s, Pollock's work on default reasons started the quest in the AI community for a formal system of defeasible argumentation. The main goal of this paper is to provide a logic of structured defeasible arguments using the language of justification logic. In this logic, we introduce defeasible justification assertions of the type $t : F$ that read as "t is a defeasible reason that justifies F". Such formulas are then interpreted as arguments and their acceptance semantics is given in analogy to Dung's abstract argumentation framework semantics. We show that a large subclass of Dung's frameworks that we call "warranted" frameworks is a special case of our logic in the sense that (1) Dung's frameworks can be obtained from justification logic-based theories by focusing on a single aspect of attacks among justification logic arguments and (2) Dung's warranted frameworks always have multiple justification logic instantiations called "realizations".

We first define a new justification logic that relies on operational semantics for default logic. One of the key features that is absent in standard justification logics is the possibility to weigh different epistemic reasons or pieces of evidence that might conflict with one another. To amend this, we develop a semantics for "defeaters": conflicting reasons forming a basis to doubt the original conclusion or to believe an opposite statement. This enables us to formalize non-monotonic justifications that prompt extension revision already for normal default theories.

Then we present our logic as a system for abstract argumentation with structured arguments. The format of conflicting reasons overlaps with the idea of attacks between arguments to the extent that it is possible to define all the standard notions of argumentation framework extensions. Using the definitions of extensions, we establish formal correspondence between Dung's original argumentation semantics and our operational semantics for default theories. One of the results shows that the notorious attack cycles from abstract argumentation cannot always be realized as justification logic default theories.

Keywords: Abstract argumentation, structured argumentation, Dung's framework, justification logic, default reasoning

1. Introduction

Defeasible reasoning is a key concept in the development of computational models of argument. Defeasible reasons became a topic of interest for AI researchers largely due to Pollock's work [63], which brought closer together the ideas of non-monotonic reasoning from AI and defeasible reasoning from philosophy. To highlight the importance of defeasibility for the study of reasoning, we use a variant of Pollock's "red-looking table" vignette [63], previously discussed by Chisholm [28]. Suppose you are standing in a room where you see red objects in front of you. This can lead you to infer that a red-looking table in front of you is in fact red. However, the reason that you have for your conclusion is defeasible. For a typical defeat scenario, suppose you learn that the room you are standing in is illuminated with red light. This gives you a reason to doubt your initial reason to conclude that the table is red, though it would not give you a reason to believe that it is not red. However, if you were to learn, instead, that the

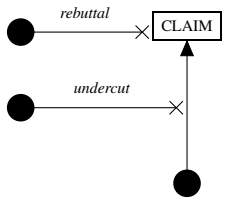

Fig. 1. The types of defeat: undercut and rebuttal.

original factory color of the table is white, then you would also have a reason to believe the denial of the claim that the table is red.

The example specifies two different ways in which reasons defeat other reasons: the former is known as *undercut* and the latter as *rebuttal*, according to Pollock's [63] terminology. Learning additional information about the light conditions incurs suspending the applicability of your initial reason to believe that the table is red. In contrast, learning that there is a separate reason to consider that the table is not red will not directly compromise your initial reason itself. The differences between undercutting and rebutting reasons are illustrated in Fig. 1.

An argument relying on default reasons is itself regarded as defeasible. The formal study of defeasible arguments is already well-developed, most prominently in the frameworks for structured argumentation represented in the 2014 special issue of this journal (vol. 5, issue 1): ABA [76], ASPIC+ [57], DeLP [42] and deductive argumentation [20].[1] These frameworks differ in the way they formalize argument structures and their defeasibility. Importantly, although all these frameworks use logic as a part of their formalization, none of them can be characterized as a *logic of defeasible arguments*. This is what will be done in the current paper. It presents a logic of defeasible arguments using the language of justification logic introduced by Artemov [5,6]. Among many advantages of formalizing arguments in a logical system, for now we will point out only a couple of the more obvious ones. First, our logic of arguments is a full-fledged normative system with definition(s) of logical consequence, which *ipso facto* enables satisfaction of structured argumentation postulates without needing to further constrain the system's behaviour. We will show this in Section 5. Secondly, our logic is not a framework for specifying other systems and it does not use any meta-level rules from an unspecified system. Instead, we formalize arguments using exclusively object level formulas and inference rules. From a computational perspective, such system is desirable as a way to manipulate arguments at a purely symbolic level.

The idea of finding a logical system with arguments as object-level formulas has already influenced the formal argumentation community. One especially interesting contribution in this direction is the logic of argumentation (LA) by Krause, Ambler, Gøransson and Fox [51]. These authors present a system in which inference rules manipulate labelled formulas interpreted as pairs of arguments and formulas[2]

$$arg : formula.$$

Our logic advances the search for the logic of arguments and builds on the take-away message from [51, p. 129] that we should take arguments "to be first class objects themselves". By refining the way

[1] The acronyms ABA, ASPIC and DeLP refer to "Assumption Based Argumentation", "Argumentation Service Platform with Integrated Components" and "Defeasible Logic Programming", respectively.

[2] The system has been used to develop applications that support medical diagnosis [31,40]. In LA, labels *arg* are interpreted as terms in the typed λ-calculus [14]. Thanks to Artemov [6, p. 7], we know that justification logic advances typed combinatory logic and typed λ-calculus allowing for e.g. iteration of type assignments and types that depend on terms.

in which we handle defeat among arguments, we make it possible to determine argument acceptance at a purely symbolic level and without using any measures of acceptability extraneous to the logic itself. This is one of the desiderata that the LA authors left open [51, § 6].

In order to formalize arguments, we embrace the strategy of using a formal language with labelled formulas. In justification logic, such labelled formulas represent pairs of reasons and claims. They are written as the so called "justification assertions" $t : F$ that read as "t is a reason that justifies formula F". The first justification logic was developed as a logic of proofs in arithmetic (logic of proofs, LP) by Artemov [6].[3] On the original reading of pairs $t : F$, the term t encodes some Peano arithmetic derivation for the statement F. Thus, the original logic of proofs does in fact give one particular formalization of arguments, namely a formalization of non-defeasible arguments. Accordingly, subsequent epistemic interpretations of justification logics provided a formal framework to deal with justifications and reasons, albeit non-defeasible ones. Even so, the underlying language of justification logic offers a powerful formal tool to model reasons as objects with operations. In this paper, the language of justifications is used to study defeasible reasons.

In Section 3, we will present the benefits of using this logical language when justification assertions are given with argumentation semantics. The language of justifications is expressive enough to combine desirable features of the mentioned structured argumentation frameworks in a single system. Here are some outcomes that a reader can expect from our approach:

- We show that default justification logic fulfills Pollock's project of defining a single formal system with strict and defeasible rules reified through deductive and default reasons. The four mentioned approaches dealing with structured argumentation are useful generalizations on how to understand arguments, but the problem we address here is how to unify their meta-analysis into a logical theory of undercut and rebuttal.
- Our system abstracts from the content of arguments, but, unlike ASPIC+ or ABA, represents arguments in the object language with default reasons. Compared to the level of abstraction in our logic, frameworks like ASPIC+ and ABA could be justly considered as meta-approaches to argumentation. As a most important contribution, our logic does not abstract from reasons. Reasons are represented as separate terms alongside the usual representation of statements and inference rules.
- Although ABA, ASPIC+ and deductive argumentation can generate Dung's frameworks, they cannot be said to provide a logical realization of Dung's frameworks because they do not define a specific logical system. In default justification logic, Dung's attack graphs, whose nodes can be interpreted as existential statements of the type "There is some argument", are realized with an explicit logical formula $t : F$ ascribed to each node of an attack graph. Determining acceptability of arguments through a normative system with logical consequence promises improvements in the area of computational argumentation.
- The logic we present here is capable of capturing all the components of Toulmin's six-fold argumentation scheme, with the exception of what he calls "qualifiers". The presence of elements like "warrant" and "backing" leads to a multi-layered understanding of an argument.[4] None of the mentioned structured argumentation frameworks gives a formalization of the added components

[3]The idea of explicit proof terms as a way to find the semantics for the provability calculus **S4** dates to 1938 and Gödel's lecture published in 1995 [43].

[4]Toulmin's book *The Uses of Argument* [77] is an acclaimed anti-formalistic argumentation monograph that separates logical methods and argumentation theory [81, p. 219]. Toulmin himself stated [77, p. vii] that the aim of his book was "to criticize the assumption, made by most Anglo-American academic philosophers, that any significant argument can be put in formal terms". One of the aims of this paper is to reunite logical methods and argumentation theory.

of arguments such as warrants and backings. In contrast, our logic represents three layers of arguments which are codified in reason terms t justifying formulas F that are not necessarily explicitly represented at every stage of manipulating the formula $t : F$ in the semantics. [5]

- Justification logic enables us to integrate default logic and argumentation theory. Our logic remedies an important limitation of constructing arguments as Reiter's defaults [81, p. 227]: Reiter's defaults are givens and it is not possible to provide reasons for why they hold. Introducing justification logic as the basic language of default rules supplies them with a formal version of Toulmin's warrants and provides a way to further reason about the acceptability of rules. In this way, default logic with warrants is able to subsume formal argumentation semantics.[6]

The rest of this article is structured as follows. The next section introduces the basic justification logic system for reasoning with certain information. Then we use this formal system to introduce default justifications based on default rules with justification formulas. The "red table" example will be used as a running example that illustrates the use of such default rules. A preliminary survey of this system was carried out in [60]. The system enables us to interpret formulas of the type $t : F$ as structured arguments with mutual attacks and to define the extension notions of Dung's framework in justification logic. We show that, by abstracting from the structure of arguments, we can obtain Dung's frameworks from the logic of default justifications and, *vice versa*, our logic provides realization procedures for Dung's frameworks that assign justification formulas to Dung's arguments. Finally, we discuss how our logic complies with the rationality postulates for structured argumentation frameworks proposed in [1].

2. Justification logic

Soon after Artemov developed the logic of proofs (LP) in [6], a possible worlds semantics for this logic was proposed by Fitting [33,34] in order to align justification logics within the family of modal logics. Syntactic objects that represent mathematical proofs in the logic of proofs LP are then more broadly interpreted as epistemic or doxastic reasons by Fitting [33,34] and Artemov [11]. A distinctive feature of justification logic taken as epistemic logic is replacing belief and knowledge modal operators that precede propositions ($\Box F$ for "F is known") by proof terms or, in a generalized epistemic context, justification terms. On top of the usual possible world condition for the truth of $t : F$ that F is true in all accessible alternatives, Fitting's semantics requires that the reason t is admissible for formula F.

Although justification logic introduced the notions of justification and reason into epistemic logic, it does not formally study the ways of *defeat* among reasons and it takes admissibility of reasons as a primitive notion. Given the pervasiveness of commonsense reasoning, we know that only a restricted group of epistemic reasons may be treated as completely immune to defeaters: mathematical proofs. But mathematical reasons form only a small part of possible reasons to accept a statement and, being a highly-idealized group of reasons, they have rarely been referred to as reasons. Fitting's possible worlds semantics for justification logics was meant to model not only mathematical and logical truths, but also

[5] With the help of these distinctions, we are able to verify apparently conflicting claims about the nature of defeat in the literature. For example, ASPIC+ correctly models undercut by referring to the exclusion of a rule that does not apply in a given context. However, at the "lower" level of the argument backing, undercut eliminates an assumption made in justifying that rule — which suggests that this type of attack might be reduced to an assumption attack, as claimed in e.g. ABA. Such meta-disagreements on the nature of defeasibility can be reconciled in a fine-grained account of arguments.

[6] Relations between Reiter's default logic [72] and argumentation are explored in Dung's seminal paper [30], but the idea of modelling arguments in default logic has been initiated earlier in the AI literature e.g. by [67].

facts of the world or "inputs from outside the structure" [36, p. 111]. Yet the original intent of the first justification logic LP to deal with mathematical proofs, together with the fact that mathematics is cumulative, is reflected in its epistemic generalizations. Accordingly, reasons that justify facts of the world were left encapsulated within a framework for non-defeasible mathematical proofs.[7]

Non-mathematical reasons and justifications are commonly held to depend on each other in acquiring their status of "good" reasons and justifications. Still, the questions related to non-ideal reasons have only recently been raised in the justification logic literature.[8] In the present paper we develop a non-monotonic justification logic with justification terms such that (1) their defeasibility can be tracked from the term structure and (2) other justifications can defeat them by means of an undercut or a rebuttal. Our logic combines techniques from default logic, justification logic and formal argumentation to represent conflicts of reasons produced in less-than-ideal ways.

2.1. *The logic of non-defeasible reasons* **JT**

Justification logics with modal semantics opened up a possibility to study formal systems for non-defeasible epistemic reasons. These systems include an explicit counterpart to the modal *Truth axiom*: $\Box F \rightarrow F$, read as "If F is known, then F".[9] In order to introduce our system of default reasons, we build upon the existing systems for non-defeasible reasons. In this respect, one can see our strategy as being analogous to the standard default logic approach [3,72] where agents reason from known or certain information. This section gives preliminaries on one of the logics of non-defeasible reasons.

What are the formal ingredients of justification logics? The language of justifications builds on the language of propositional logic, which is augmented by formulas labelled with reason terms $(t : F)$ and a grammar of operations on such terms. Reason terms are constructed from constants and variables, combined with the use of operations on terms. Intuitively, constants justify logical postulates and variables justify contingent facts or inputs outside the structure. The basic operation of standard justification logics is *application*. Intuitively, application produces a reason term $(u \cdot t)$ for a formula G which is a syntactic "imprint" of the *modus ponens* step from $F \rightarrow G$ and F to G for some labelled formulas $u : (F \rightarrow G)$ and $t : F$. We say that the term u has been applied to the term t to obtain the term $(u \cdot t)$. As a distinctive feature of justification terms, the history of reasoning steps taken in producing such terms is recorded in their structure.

Another common operation on justification terms is *sum*. Intuitively, if one takes that a reason term t justifies some formula F, then one is allowed to affix any other reason u term by the use of sum so that the new reason term $(t + u)$ still justifies F. On an epistemic interpretation, this operation can be informally motivated as follows [9, Section 2.2]: t and u might be thought of as two volumes of an encyclopedia that are used as evidence for some statement F. If one volume justifies F, then adding the other volume to the corpus of evidence does not compromise the justification for F. The axioms regulating the sum and application operations are formally described in this section, following the definition of the language.

[7] See [17, p. 620] for a discussion on the difference between mathematical proofs and persuasive arguments. For a more encompassing overview of standard justification logics see [10] or [53].

[8] The first proposed formalism that includes the idea of evidence elimination specific to a multi-agent setting is by Renne [73]. Baltag, Renne and Smets [12,13] bring together ideas from belief revision and dynamic epistemic logic and offer an account of good and conclusive evidence. Several approaches ([48,49,55,59]) start from the idea of merging probabilistic degrees of belief with justification logic, while [32] and [74] develop a possibilistic justification logic. In [39], Fitting introduces a paraconsistent formal system with justification assertions where contradictions can be interpreted as conflicting evidence.

[9] In fact, in [35, p. 156] we find three different truth axiom schemes. Varieties of systems with the truth axiom have been extensively studied and described in e.g. [35] and [52].

Since we assume in the next section that an agent starts to reason from indefeasible information, we want our underlying logic to represent "factive" or "truth-inducing" reasons. However, additional constraints on the system are not necessarily needed to introduce the system of default reasons. For the sake of formal clarity, we leave out standard axioms and operations that ensure positive or negative introspection, although these can be easily added. Accordingly, an adequate logical account of factive justifications is the logic **JT**, a justification logic with the axiom schemes that are explicit analogues of the axiom schemes for the modal logic **T**.[10] Intuitively, a reader can think of the **JT** logic as modelling an idealized arguer whose arguments fully exhaust all the possible information regarding claims and who, therefore, gives indisputable reasons for those claims. Note that there are also weaker variants of justification logic that do not assume factivity of reasons. These systems are not adequate for our purposes since we want to build defeasible arguments from a base of fact-inducing reasons — just as in standard default logic where reasoning starts from non-defeasible information or facts [3, p. 19]. After we define the underlying logic that represents non-defeasible argumentation, we develop our novel non-monotonic approach to reasons and provide this logic with the semantics for defeasible argumentation.

2.1.1. Syntax

Syntactically, knowledge operators take the form of justification terms preceding formulas: $t : F$. Given that "t" is a justification term and that "F" is a formula, we write "$t : F$", where t is informally interpreted as a reason or justification for F. We define the set Tm that consists of exactly all justification terms, constructed from variables $x_1, \ldots, x_n, \ldots$ and proof constants $c_1, \ldots, c_n, \ldots$ by means of operations $\cdot$ and $+$. The grammar of justification terms is as follows:

$$t ::= x \mid c \mid (t \cdot t) \mid (t + t)$$

where x is a variable denoting an unspecified justification and c is a proof constant. Proof constant c is atomic within the system. For a justification term t, a set of subterms $Sub(t)$ is defined by induction on the construction of t. Formulas of **JT** are defined by the following Backus-Naur form:

$$F ::= \top \mid P \mid (F \to F) \mid (F \vee F) \mid (F \wedge F) \mid \neg F \mid t : F$$

where $P \in \mathcal{P}$ and $\mathcal{P}$ is a countable set of atomic propositional formulas and $t \in Tm$. The set Fm consists of exactly all formulas.

2.1.2. Axioms and rules of JT

We can now define the logic of non-defeasible reasons **JT**. The logic **JT** is the weakest logic with "truth inducing" justifications containing axiom schemes for the two basic operations $\cdot$ and $+$.[11] These are the axioms and rules of **JT**:

A0 *All the instances of propositional logic tautologies from* Fm
A1 $t : (F \to G) \to (u : F \to (t \cdot u) : G)$ (Application)
A2 $t : F \to (t + u) : F; u : F \to (t + u) : F$ (Sum)
A3 $t : F \to F$ (Factivity)

[10] Justification logic **JT** was first introduced by [21]. Justification logics with equivalent axiom schemes to the logic we define in this section are also defined and investigated in [52] and [35].

[11] As Fitting [34,35] shows, we can also technically consider dropping the operator $+$ from our language. In this way we obtain the logic that he calls $LP^{-}(T)$ [35, p. 162].

R0 *From* F *and* $F \to G$ *infer* G (Modus ponens)
R1 *If* F *is an axiom instance of A0-A3 and* $c_n, c_{n-1}, \ldots, c_1$ *proof constants, then infer* $c_n : c_{n-1} : \cdots : c_1 : F$ (Iterated axiom necessitation)

Proof constants are justifications of basic logic axioms. In justification logics, basic logic axioms are taken to be justified by virtue of their status within a system and their justifications are not further analyzed. Moreover, all the justification assertions of the format $c : F$ are themselves postulated to be justified by a constant, where proof constants can be nested at any depth.[12] A set of instances of all such canonical formulas in justification logic is called a *Constant Specification* ($\mathcal{CS}$) set:

Definition 1 (Constant specification)**.** The *Constant Specification* set is the set of instances of rule R1.

$$\mathcal{CS} = \{c_n : c_{n-1} : \cdots : c_1 : F \mid F \text{ is an axiom instance of A0–A3,}$$
$$c_n, c_{n-1}, \ldots, c_1 \text{ are proof constants and } n \in \mathbb{N}\}$$

The use of constants in R1 above is unrestricted. In such format, the rule generates a set of formulas where each axiom is justified by any constant at any depth. The set of formulas obtained in this way is called the *Total Constant Specification* ($\mathcal{TCS}$) set. A more appropriate name for the logic above would therefore be $\mathbf{JT}_{\mathbf{TCS}}$. It is possible to put restrictions on the use of constants in rule R1 in order to consider a limited class of $\mathcal{CS}$-sets. We restrict the constant specification set $\mathcal{CS}$ following a simple requirement that each axiom instance has its own proof constant.[13]

Restriction 2. $\mathcal{CS}$ is

- Axiomatically appropriate: for each axiom instance A, there is a constant c such that $c : A \in \mathcal{CS}$ and for each formula $c_n : c_{n-1} : \cdots : c_1 : A \in \mathcal{CS}$ such that $n \geqslant 1$, there is a constant c_{n+1} such that $c_{n+1} : c_n : c_{n-1} : \cdots : c_1 : A \in \mathcal{CS}$;
- Injective: each proof constant c justifies at most one formula.

The logic $\mathbf{JT}_{\mathbf{CS}}$ is defined by replacing the iterated axiom necessitation rule of $\mathbf{JT}_{\mathbf{TCS}}$ with the following rule dependent on Restriction 2:

R1* *If* F *is an axiom instance of A0-A3 and* $c_n, c_{n-1}, \ldots, c_1$ *proof constants such that* $c_n : c_{n-1} : \cdots : c_1 : F \in \mathcal{CS}$, *then infer* $c_n : c_{n-1} : \cdots : c_1 : F$

We say that the formula F is $\mathbf{JT}_{\mathbf{CS}}$-provable ($\mathbf{JT}_{\mathbf{CS}} \vdash F$) if F can be derived using the axioms A0-A3 and rules R0 and R1*.

2.1.3. Semantics

The semantics for $\mathbf{JT}_{\mathbf{CS}}$ is an adapted version of the semantics for the logic of proofs (**LP**) given by [56].[14] Intuitively, the semantics extends that of propositional logic with a function that ascribes reason

[12] This is required to ensure that standard properties as *Internalization* [6] hold.

[13] For example, one such constant specification is defined by Artemov [8, p. 31]: "$c_n : A \in \mathcal{CS}$ iff A is an axiom and n is the Gödel number of A". The choice of $\mathcal{CS}$ is not trivial. If we define an empty $\mathcal{CS}$, that is, $\mathbf{JT}_{\emptyset}$, we eliminate logical awareness for agents, while defining an infinite $\mathcal{CS}$ imposes logical omniscience. Moreover, different restrictions could affect complexity results, as discussed in e.g. [54].

[14] The condition for justifications of the type '$!t$' are not needed in the $\mathbf{JT}_{\mathbf{CS}}$ semantics. Mkrtychev's model can be thought of as a single world justification model. Since the notion of defeasibility introduced in the next section turns on the incompleteness of available reasons, our system eliminates worries about the trivialization of justification assertions that otherwise arise from considering justifications as modalities in a single-world model.

terms to formulas in such a way that it respects the sum and application axioms and some constant specification $\mathcal{CS}$ that satisfies Restriction 2.

Definition 3 ($\mathbf{JT_{CS}}$ model)**.** We define a function *reason assignment* based on $\mathcal{CS}$, $*(\cdot) : Tm \to 2^{Fm}$, a function mapping each term to a set of formulas from Fm. We assume that it satisfies the following conditions:

(1) If $F \to G \in *(t)$ and $F \in *(u)$, then $G \in *(t \cdot u)$
(2) $*(t) \cup *(u) \subseteq *(t + u)$
(3) If $c : F \in \mathcal{CS}$, then $F \in *(c)$

A *truth assignment* $v : \mathcal{P} \to \{True, False\}$ is a function assigning truth values to propositional formulas in $\mathcal{P}$. We define the interpretation $\mathcal{I}$ as a pair $(v, *)$. For an interpretation $\mathcal{I}$, $\models$ is a truth relation on the set of formulas of $\mathbf{JT_{CS}}$.

For any formula $F \in Fm$, $\mathcal{I} \models F$ iff

- For any $P \in \mathcal{P}$, $\mathcal{I} \models P$ iff $v(P) = True$
- $\mathcal{I} \models \neg F$ iff $\mathcal{I} \not\models F$
- $\mathcal{I} \models F \to G$ iff $\mathcal{I} \not\models F$ or $\mathcal{I} \models G$
- $\mathcal{I} \models F \vee G$ iff $\mathcal{I} \models F$ or $\mathcal{I} \models G$
- $\mathcal{I} \models F \wedge G$ iff $\mathcal{I} \models F$ and $\mathcal{I} \models G$
- $\mathcal{I} \models t : F$ iff $F \in *(t)$

An interpretation $\mathcal{I}$ is *reflexive* iff the truth relation for $\mathcal{I}$ fulfills the following condition:

- For any term t and any formula F, if $F \in *(t)$, then $\mathcal{I} \models F$.

In the absence of the reflexivity condition, it is possible that $\mathcal{I} \models t : F$ and $\mathcal{I} \models \neg F$. While reasons in reflexive models can be taken as *conclusive* or factive, without the reflexivity condition reasons are interpreted as being only *admissible*. In possible worlds semantics, the admissibility condition $F \in *(t)$ for the truth of $t : F$ is supplemented with the condition that F holds in all accessible alternatives [34, p. 4]. The consequence relation of the logic of factive reasons $\mathbf{JT_{CS}}$ is defined on reflexive interpretations:

Definition 4 ($\mathbf{JT_{CS}}$ consequence relation)**.** $\Sigma \models F$ iff for all reflexive interpretations $\mathcal{I}$, if $\mathcal{I} \models B$ for all $B \in \Sigma$, then $\mathcal{I} \models F$.

Due to Restriction 2, the consequence relation for $\mathbf{JT_{CS}}$ is weaker than the $\mathbf{JT_{TCS}}$ consequence relation.

Definition 5 ($\mathbf{JT_{CS}}$ closure)**.** $\mathbf{JT_{CS}}$ closure is given by $Th^{JT_{CS}}(\Gamma) = \{F | \Gamma \models F\}$, for a set of formulas $\Gamma \subseteq Fm$ and the $\mathbf{JT_{CS}}$ consequence relation $\models$ defined above.

For any closure $Th^{JT_{CS}}(\Gamma)$, it follows that $\mathcal{CS} \subseteq Th^{JT_{CS}}(\Gamma)$. Later, we also make use of a weaker closure Th^{-} for which a reason assignment function $*(\cdot)$ does not satisfy condition (2) and CS is relativized to its subset $\mathcal{CS}^{-}$ such that $\mathcal{CS}^{-}$ does not contain any formula $c_n : c_{n-1} : \cdots : c_1 : F$, where F is an instance of A2 (cf. [34]).

We can prove that the compactness theorem holds for the $\mathbf{JT_{CS}}$ semantics.[15] Compactness turns out to be a useful result in defining the operational semantics of default reason terms. We first say that a set

[15] A compactness proof for **LP** satisfiability in possible world semantics is given in [34]. A similar proof is given for $\mathbf{JT_{CS}}$ in the Appendix to provide a self-contained introduction to $\mathbf{JT_{CS}}$ in this paper.

of formulas Γ is $\mathbf{JT_{CS}}$ *satisfiable* iff there is an interpretation $\mathcal{I}$ that meets $\mathcal{CS}$ (via the third condition of Def. 3) for which all the members of Γ are true. A set Γ is $\mathbf{JT_{CS}}$-finitely satisfiable if every finite subset Γ' of Γ is $\mathbf{JT_{CS}}$ satisfiable.

Theorem 6 (Compactness)**.** *A set of formulas is* $\mathbf{JT_{CS}}$ *satisfiable iff it is* $\mathbf{JT_{CS}}$*-finitely satisfiable.*

Proof. See the Appendix. □

3. A logic of default justifications

In this section, we develop a system based on $\mathbf{JT_{CS}}$, in which an agent forms default justifications reasoning from incomplete information. Justification logic $\mathbf{JT_{CS}}$ is capable of representing the construction of a new piece of evidence out of existing ones by application ("$\cdot$") or sum ("$+$") operation. However, to extend an incomplete $\mathbf{JT_{CS}}$ theory, we need to import reasons that are defeasible. We come up with both a way in which such reasons are imported and a way in which they might get defeated. These possibilities are opened up by introducing concepts familiar from defeasible reasoning literature into justification logic.

We start from the above-defined language of the logic $\mathbf{JT_{CS}}$ and develop a new variant of justification logic $\mathbf{JT_{CS}}$ that enables us to formalize the import of reasons outside the structure as well as to formalize *defeaters* or reasons that question the plausibility of other reasons.

Our logical framework of defeasible reasons represents both factive reasons produced via the axioms and rules of $\mathbf{JT_{CS}}$ and plausible reasons based on default assumptions that "usually" or "typically" hold for a restricted context. We follow the standard way [72] of formalizing default reasoning through default theories to extend the logic of factive reasons with defeasible reasons. Building on the syntax of $\mathbf{JT_{CS}}$, we introduce the definition of the *default theory*:

Definition 7 (Default Theory)**.** A default theory T is defined as a pair (W, D), where the set W is a finite set of $\mathbf{JT_{CS}}$ formulas and D is a countable set of default rules.

Each default rule is of the following form:

$$\delta = \frac{t : F :: (u \cdot t) : G}{(u \cdot t) : G}.$$

The informal reading of the default δ is: "If t is a reason justifying F, and it is consistent to assume that $(u \cdot t)$ is a reason justifying G, then $(u \cdot t)$ is a defeasible reason justifying G". The formula $t : F$ is called the *prerequisite* and $(u \cdot t) : G$ is both the *consistency requirement*[16] and the *consequent* of the default rule δ. We refer to each of the respective formulas as $pre(\delta)$, $req(\delta)$ and $cons(\delta)$. For the set of all consequents from the entire set of defaults D, we use $cons(D) = \{cons(\delta) \mid \delta \in D\}$. The default rule δ introduces a unique reason term u, which means that, for a default theory T, the following holds:

(1) For any formula $v : H \in Th^{JT_{CS}}(W)$, $u \neq v$;
(2) For any formula $H \in W$, $u : (F \to G)$ is not a subformula of H and
(3) For any default rule $\delta' \in D$ such that $\delta' = \frac{t' : F' :: (u' \cdot t') : G'}{(u' \cdot t') : G'}$, if $u = u'$, then $F = F'$ and $G = G'$.

[16] In order to avoid any misunderstanding, we avoid the name *justification* for the formula $req(\delta)$ since justification logic terms are commonly known as justifications.

Note that the term u does not need to be fresh in the sense that it cannot appear in two different defaults' consequents.[17] Default reasons may refer to other default reasons and this possibility is crucial to represent interactions among defaults. The unique reason term u witnesses the defeasibility of the *prima facie* reason $(u \cdot t)$ for G. Whether a reason actually becomes defeated or not depends on other default-reason formulas from $cons(D)$. Other defaults might question both the plausibility of the reasoning that u codifies and the plausibility of the proposition G. Section 3.1 gives an example of a concrete $\mathbf{JT_{CS}}$ derivation that instantiates unique reason terms.

A formal way of looking at a default reason of this kind is that $(u \cdot t)$ codifies the default step we apply on the basis of the known reason t. A distinctive feature of such rules is generating justification terms as if it were the case that $cons(\delta)$ was inferred by using an instance of the application axiom: $u : (F \to G) \to (t : F \to (u \cdot t) : G)$. The difference is that an agent cannot ascertain that an available reason justifies applying the conditional $F \to G$ without restrictions. Still, sometimes a conclusion must be drawn without being able to remove all of the uncertainty as to whether the relevant conditional actually applies or not. In such cases, an agent turns to a plausible assumption of a justified "defeasible" conditional $F \to G$ that holds only in the absence of any information to the contrary. While the internal structure of the default reason $(u \cdot t)$ indicates that it is formed on the basis of the formula $u : (F \to G)$, the defeasibility of $(u \cdot t)$ lies in the fact that the formula $u : (F \to G)$ is not a part of the same evidence base as $(u \cdot t) : G$.

One can think of our use of the operation "$\cdot$" in default rules as the same operation that is used in the axiom A1, only being applied on an incomplete $\mathbf{JT_{CS}}$ theory. Similarly, we can follow Reiter [72, p. 82] and Antoniou [3, p. 21] in thinking of a standard default rule such as $\frac{A:B}{B}$ as merely saying that an implication $A \wedge \neg C \wedge \neg D \cdots \to B$ holds, provided that we can establish that a number of exceptions $C, D, \ldots$ does not hold. However, if the rule application context is defined sufficiently narrowly, the rule is classically represented as an implication $A \to B$. Generalizing on such interpretation of defeasibility, our defaults with justification assertions can be represented as instantiations of the axiom A1 applied in a sufficiently narrow application context.

Analogous to standard default theories, we take the set of facts W to be underspecified with respect to a number of facts that would otherwise be specified for a complete $\mathbf{JT_{CS}}$ interpretation. Besides simple facts, our underlying logic contains justification assertions. To deal with justification assertions, a complete $\mathbf{JT_{CS}}$ interpretation would also further specify whether a reason is acceptable as a justification for some formula. Therefore, except the usual incomplete specification of known propositions, default justification theories are also incomplete with respect to the actual specification of the reason assignment function. For our default theory, this means that, except the valuation v, default rules need to approximate an actual reason-assignment function $*(\cdot)$.

Let us again consider the red-looking-table example from the Introduction to see how *prima facie* reasons and their defeaters are imported through default rules.

Example 8. Let R be the proposition "the table is red-looking" and let T be the proposition "the table is red". Take t_a and u_a to be some specific individual justifications. The reasoning whereby one accepts the default reason $(u_a \cdot t_a)$ might be described by the following default rule:

$$\delta_a = \frac{t_a : R :: (u_a \cdot t_a) : T}{(u_a \cdot t_a) : T}.$$

[17] Compare Artemov's [8, p. 30] introduces "single-conclusion" (or "pointed") justifications that enable handling "justifications as objects rather than as justification assertions".

We can informally read the default as follows: "If t_a is a reason justifying that a table is red looking and it is consistent for you to assume that this gives you a reason $(u_a \cdot t_a)$ justifying that the table is red, then you have a defeasible reason $(u_a \cdot t_a)$ justifying that the table is red". Suppose you then get to a belief that "the room you are standing in is illuminated with red light", a proposition denoted by L. For some specific justifications t_b and u_b, the following rule gives you an undercutting reason for $(u_a \cdot t_a)$:

$$\delta_b = \frac{t_b : L :: (u_b \cdot t_b) : \neg[u_a : (R \to T)]}{(u_b \cdot t_b) : \neg[u_a : (R \to T)]},$$

where the rule is read as "If t_b is a reason justifying that the lighting is red and it is consistent for you to assume that this gives you a reason $(u_b \cdot t_b)$ denying that the reason u_a justifies that if the table is red-looking, then it is red, then you have a defeasible reason $(u_b \cdot t_b)$ denying that the reason u_a justifies that if the table is red-looking, then it is red". The formula $cons(\delta_b)$ denies your reason to conclude $cons(\delta_a)$, although note that it is not directly inconsistent with $cons(\delta_a)$. In Section 3.2, we define what undercutting defeaters are semantically.

Suppose that instead of learning about the light conditions in the room as in δ_b, you learn that the original factory color of the table is white. This would also prompt a *rebutting defeater* – a separate reason to believe the contradicting proposition $\neg T$. Let W denote the proposition "the table is originally white" and let t_c and u_c be some specific justifications. We have the following rule:

$$\delta_c = \frac{t_c : W :: (u_c \cdot t_c) : \neg T}{(u_c \cdot t_c) : \neg T}.$$

The rule reads as "If t_c is a reason justifying that the table is originally white and it is consistent for you to assume that this gives you a reason $(u_c \cdot t_c)$ justifying that the table is not red, then you have a defeasible reason $(u_c \cdot t_c)$ justifying that the table is not red". Note that the formula $cons(\delta_c)$ does not directly mention any of the subterms of $(u_a \cdot t_a)$. The defeat among the reasons $(u_a \cdot t_a)$ and $(u_c \cdot t_c)$ comes from the fact that they cannot together consistently extend an incomplete $\mathbf{JT_{CS}}$ theory.

The entire example can be described by the following default theory $T_0 = (W_0, D_0)$, where $W_0 = \{t_a : R, t_b : L, t_c : W\}$ and $D_0 = \{\delta_a, \delta_b, \delta_c\}$.

Each defeater above is itself defeasible and considered to be a *prima facie* reason. The way in which *prima facie* reasons interact is further specified through their role in the operational semantics. By the end of this section, we explain the workings of the operational semantics that determines the acceptable reasons given a definition of a default theory.

3.1. Operational semantics of default justifications

The logic of default justifications we develop here relies on the idea of operational semantics for standard default logics presented in [3]. Let us informally describe the role of the steps of operational semantics in determining acceptable reasons. First, in the operational part of the semantics, default reasons are taken into consideration at face value. Then we check dependencies among default reasons in order to find out what are the non-defeated reasons. Finally, a rational agent includes in its knowledge base only acceptable pieces of information that are based on those reasons that are ultimately non-defeated. An important part of the latter step is an acceptance semantics analogous to the argument acceptance semantics of formal argumentation frameworks.

The basis of operational semantics for a default theory $T = (W, D)$ is the procedure of collecting new, reason-based information from the available defaults. A *sequence* of default rules $\Pi = (\delta_0, \delta_1, \ldots)$ is a possible order in which a list of default rules without multiple occurrences from D is applied (Π is possibly empty). Applicability of defaults is determined in the following way:

Definition 9 (Applicability of Default Rules)**.** For a set of $\mathbf{JT_{CS}}$-closed formulas Γ we say that a default rule $\delta = \frac{t:F::(u\cdot t):G}{(u\cdot t):G}$ is applicable to Γ iff

- $t : F \in \Gamma$ and
- $\neg(u \cdot t) : G \notin \Gamma$.[18]

Reasons are brought together in the set of $\mathbf{JT_{CS}}$ formulas that represents the current *evidence base*:

Definition 10. $In(\Pi) = Th^{JT_{CS}}(W \cup \{cons(\delta) \mid \delta \text{ occurs in } \Pi\})$.

The set $In(\Pi)$ collects reason-based information that is yet to be determined as acceptable or unacceptable depending on the acceptability of reasons and counter-reasons for formulas.

We need to further specify sequences of defaults that are significant for a default theory T: default processes. For a sequence Π, the initial segment of the sequence is denoted as $\Pi[k]$, where k stands for the number of elements contained in that segment of the sequence and where k is a minimal number of defaults for the sequence Π. Any segment $\Pi[k]$ is also a sequence. Intuitively, the set of formulas $In(\Pi)$ represents an update of the incomplete evidence base W where the new information is not yet taken to be granted. Using the notions defined above, we can now get clear on what a default process is:

Definition 11 (Process)**.** A sequence of default rules Π is a *process* of a default theory $T = (W, D)$ iff every k such that $\delta_k \in \Pi$ is *applicable* to the set $In(\Pi[k])$, where $\Pi[k] = (\delta_0, \ldots \delta_{k-1})$.

The kind of process that we are focusing on here is called *closed* process and we say that a process Π is closed iff every $\delta \in D$ that is applicable to $In(\Pi)$ is already in Π. For default theories with a finite number of defaults, closure for any process Π is obviously guaranteed by the applicability conditions. However, if a set of defaults is infinite, then this is less-obvious.

Lemma 12 (Infinite Closed Process)**.** *For a theory $T = (W, D)$ and infinitely many k's, an infinite process Π is closed iff for every default rule δ_k* applicable *to the set $In(\Pi[k])$, $\delta_k \in \Pi$.*

Proof. From the compactness of $\mathbf{JT_{CS}}$ semantics we have that if a set $In(\Pi[k]) \cup \{req(\delta)\}$ is satisfiable for all the finite k's, it is also satisfiable for infinitely many k's. Therefore the applicability conditions for a rule δ are equivalent to the finite case. □

To illustrate how the basic notions of the operational semantics work, Fig. 2 shows the process tree for the default theory T_0 from our running Example 8.

The figure shows that T_0 has four closed processes: $\Pi_1 = (\delta_a, \delta_b)$, $\Pi_2 = (\delta_b, \delta_a)$, $\Pi_3 = (\delta_b, \delta_c)$ and $\Pi_4 = (\delta_c, \delta_b)$. The *In*-sets $In(\Pi_1)$ and $In(\Pi_2)$ are equal and $\mathbf{JT_{CS}}$-inconsistent with $In(\Pi_3)$ and $In(\Pi_4)$, which are also equal. Whenever two sets $In(\Pi)$ and $In(\Pi')$ are not equal, they are $\mathbf{JT_{CS}}$-inconsistent. We can already see that $\mathbf{JT_{CS}}$-inconsistent *In*-sets capture the idea of rebuttal in our semantics, as introduced informally in Example 8. For example, $\mathbf{JT_{CS}}$-inconsistent *In*-sets $In(\Pi_1)$ and $In(\Pi_4)$ reflect the

[18] We follow the convention of omitting parentheses around the expression $(u \cdot t) : G$ and interpret the negation as binding the entire expression $(u \cdot t) : G$. The convention is also familiar from modal logics.

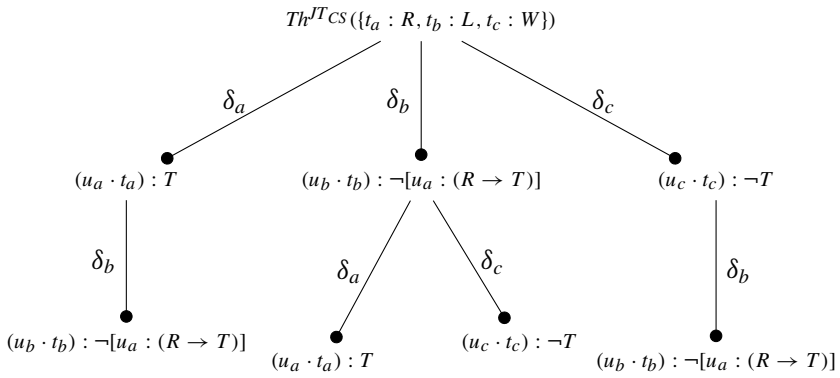

Fig. 2. The process tree of T_0 from Example 8.

opposition between the reasons $(u_a \cdot t_a)$ and $(u_c \cdot t_c)$. At the level of process trees, however, we are not yet able to explain the attack on $(u_a \cdot t_a)$ by the undercutting reason $(u_b \cdot t_b)$. To do so, we need to move further from the semantics of collecting new information.

We have already discussed the key components of our operational semantics that bear some similarity to standard default theories. Now we develop our new argument semantics that builds on the expressivity of the justification logic language. We show that the default variant of the application operation is essential to the way in which we represent arguments and their mutual attacks in justification logic.

3.2. Argumentative schemes and argumentative attacks in justification logic

In a complete specification of $\mathcal{I}$, acceptability of reasons for a formula is determined *ex officio* by assigning formulas to reasons through the function $*(\cdot)$. In contrast, in reasoning from an incomplete evidence base W, a closure $Th^{JTCS}(W)$ is typically underspecified as to whether a reason t is acceptable for a formula F. In "guessing" what a true interpretation is, every default rule introduces a reason term whose structure codifies an application operation step from an unknown justified conditional. For example, in rule δ above, we rely on the justified conditional $u : (F \to G)$. Even though this justified conditional is not a part of the rule δ itself, it is the underlying assumption on the basis of which we are able to extend an incomplete evidence base. The propositions of this kind are in one sense taken as rules allowing for default steps, but they are also specific justification logic formulas. They will be referred to as "warrants", because their twofold role in our system corresponds to Toulmin's concept of argument warrants.[19] Justification logic defaults give a formal meaning to Toulmin's philosophical idea that warrants are formulated as statements, even though they function as rules of inference within arguments. Each underlying formula of this kind can be made explicit by means of a function *warrant assignment*: $\#(\cdot) : D \to Fm$. The function maps each default rule to a specific justified conditional as follows:

$$\#(\delta) = u : (F \to G),$$

[19]Toulmin explains [77, p. 91] inference-licensing warrants as follows: "...taking these data as a starting point, the step to the original claim or conclusion is an appropriate and legitimate one. At this point, therefore, what are needed are general, hypothetical statements, which can act as bridges, and authorise the sort of step to which our particular argument commits us."

where $\delta \in D$ and

$$\delta = \frac{t : F :: (u \cdot t) : G}{(u \cdot t) : G},$$

for some reason term t, a unique reason term u and some formulas F and G.

A set of all such underlying warrants of default rules is called *Warrant Specification* ($\mathcal{WS}$) set.

Definition 13 (Warrant specification)**.** For a default theory $T = (W, D)$, justified defeasible conditionals are given by the *Warrant Specification* set:

$$\mathcal{WS}^T = \#(D) = \{\#(\delta) \mid \delta \in D\}.$$

We will use warrant specification sets that are relativized to default processes:

$$\mathcal{WS}^{\Pi} = \{\#(\delta) \mid \delta \in \Pi\}.$$

In reasoning from incomplete information, defeasible justification assertions from $\mathcal{WS}^T$ are the only available resource to approximate a reason assignment function that actually holds. Moreover, the use of underlying assumptions from $\mathcal{WS}^T$ is responsible for the non-monotonic character of default reasons. Thus our default rules are in contrast with the standard application operation represented by the axiom A1. The extended meaning of the application operation via default rules will be referred to as **default application**. Importantly, default application extends the standard idea of "proof terms" in justification logic so as to include reason terms that codify inference steps from assumptions to warrant formulas as conclusions dependent on those assumptions. We briefly explain this idea after we specify how warrants and default application are decisive for the semantics of attacks between arguments.

The extension of the application operation to its defeasible variant opens new possibilities for a semantics of justifications. In particular, it enables reasoning that is not regimented by the standard axioms A1 and A2 of basic justification logic [7, p. 482]. For instance, if a set of $\mathbf{JT_{CS}}$ formulas contains both a *prima facie* reason t and its defeater u, then the set containing a conflict of justifications does not support concatenation of reasons by which $t : F \rightarrow (t + u) : F$ holds for any two terms t and u. In other words, the possibility of a conflict between reasons eliminates the monotonicity property of justifications assumed in the sum axioms (A2).

In explaining the basics of the operational semantics, we qualified the semantics of rebutting attacks as being straightforward. Rebuttal is already captured in the mechanism of multiple extensions known from standard default theories. What requires additional explanation is the semantics of undercutting defeaters. Notice that each formula $\#(\delta)$ has the format of a justified material conditional. This formula is not a part of a default inference δ itself, but the default application described by δ depends on a conjecture that the conditional holds and the justification assertion $cons(\delta)$ encodes this conjecture in the internal structure of the resulting reason term. This brings to attention the following possibility: an evidence base may at the same time contain justified formulas of the type $t : F$, $(u \cdot t) : G$ and $v : \neg[u : (F \rightarrow G)]$, without the evidence base being $\mathbf{JT_{CS}}$-inconsistent.

Although the application axiom A1 does not say that $t : F$ and $(u \cdot t) : G$ together entail the formula $u : (F \rightarrow G)$, there is, intuitively, something wrong with the reason $(u \cdot t)$ justifying the formula G, taken together with t justifying F and v justifying $\neg u : (F \rightarrow G)$. This new type of opposition among reasons explains why we need to refer to warrant formulas. The co-occurrence of the formulas $t : F$, $(u \cdot t) : G$

and $v : \neg[u : (F \rightarrow G)]$ together is not significant in standard justification logic where reasoning is exclusively regulated by the standard axioms for idealized reasons, such as the axioms of the basic $\mathbf{JT_{CS}}$ logic. It only becomes significant with default application.[20] We will now use the presented "reverse engineering" of axiom A1 to model undercut.[21]

We have already discussed why the semantics of undercut cannot be reduced to the existence of multiple inconsistent extensions. Nevertheless, $\mathbf{JT_{CS}}$ inconsistency is important for undercutting attacks.[22] Notice that adding arbitrary warrants from $\mathcal{WS}^T$ to an evidence base $In(\Pi)$ could lead to an inconsistent set of $\mathbf{JT_{CS}}$ formulas. In Example 8, if we start from any evidence base of T_0 and add the warrant $u_a : (R \rightarrow T)$ of δ_a to it, the union becomes $\mathbf{JT_{CS}}$-inconsistent with both the warrant $u_b : (L \rightarrow \neg[u_a : (R \rightarrow T)])$ of δ_b and the warrant $u_c : (W \rightarrow \neg T)$ of δ_c. This means that the three warrants are jointly incompatible in the context of default reasoning defined by T_0. An agent needs to find out which warrants and, thereby, which reasons prevail in a conflicting set of warrants. This procedure relies on the following definition that captures the above-discussed intuition behind undercut:

Definition 14 (Undercut)**.** A reason u undercuts reason t being a reason for a formula F in a set of $\mathbf{JT_{CS}}$ formulas $\Gamma \subseteq In(\Pi[k])$ iff $\bigvee_{(v)\in Sub(t)} u : \neg[v : (G \rightarrow H)] \in Th^{JT_{CS}}(\Gamma)$ and there is a process Π' of T such that $v : (G \rightarrow H) \in \mathcal{WS}^{\Pi'}$.

We will also specify the way in which sets of $\mathbf{JT_{CS}}$ formulas undercut some default reason. This definition will be used in defining different variants of default theory extensions. Sets of justification logic formulas are said to undercut reasons according to the following definition:

Definition 15. A set of $\mathbf{JT_{CS}}$ formulas $\Gamma \subseteq In(\Pi[k])$ undercuts reason t being a reason for a formula F iff $\bigvee_{(v)\in Sub(t)} \neg[v : (G \rightarrow H)] \in Th^{JT_{CS}}(\Gamma)$ and there is a process Π' of T such that $v : (G \rightarrow H) \in \mathcal{WS}^{\Pi'}$.

One can think of Γ as a set of reasons against which the reason t is tested as a reason that justifies the formula F. This is further elaborated in the semantics of acceptability of reasons. By introducing default reasons through default application and considering rebuttal and undercut among such reasons, it is possible to take an argumentation perspective to justification logic formulas. For example, Fig. 3 provides an intuitive Toulminian interpretation of the default reasoning steps with justification formulas in Example 8, where each step can be associated with a corresponding step in the Toulminian argument scheme.[23]

Note that the formula $(u_c \cdot t_c) : \neg T$ is captioned as a rebuttal of the formula $(u_a \cdot t_a) : T$, but $(u_a \cdot t_a) : T$ also rebuts $(u_c \cdot t_c) : \neg T$. Their rebuttal relation is symmetric because the two conclusions

[20] Notice that a ($\mathbf{JT_{CS}}$-closed) evidence base that contains the formulas $t : F$ and $(u \cdot t) : G$, also contains the formula $((c \cdot t) \cdot (u \cdot t)) : (F \rightarrow G)$, assuming that the constant c justifies the axiom $F \rightarrow (G \rightarrow (F \rightarrow G))$. This is so regardless of whether $u : (F \rightarrow G)$ is also in the evidence base or not.

[21] One way to model exclusionary reasons and undercutters in default logic is to use non-normal defaults. However, with the use of non-normal defaults, many desirable features of default logics are lost, and this holds already for semi-normal defaults [3, Chapter 6]. Besides that, the use of justification logic warrants provides an elegant way to subsume argumentation semantics in default logic. For a more extensive discussion on the benefits of warrants over non-normal defaults see [61].

[22] Later, in Lemma 24, we characterize the relation between rebuttal and undercut formally.

[23] A reader should take the following two provisos into account here. Firstly, Toulmin does not use the term "undercutter". Instead, Toulmin uses rebuttal as an ambiguous concept that, among other kinds of defeat, covers for circumstances in which the general authority of the warrant would have to be set aside [81, p. 235]. Secondly, our scheme does not include "qualifiers" [77, p. 94] that indicate the strength of the step from grounds to claim.

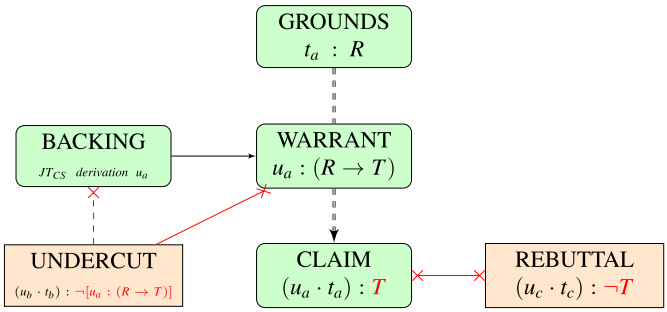

Fig. 3. Toulminian layout of arguments in Example 8.

T and $\neg T$ of the default reasons $(u_a \cdot t_a)$ and $(u_c \cdot t_c)$ are contradictory, which means that applying either of the default rules δ_a and δ_c blocks the application of the other default rule. Moreover, in Toulmin's scheme of argumentation, backing is understood as a certification or evidence for the use of a warrant to introduce some conclusion. In justification logic, backing naturally translates into a $\mathbf{JT_{CS}}$ derivation (with undischarged assumptions) of a default conditional, and the steps of that derivation are codified in the reason term u_a that justifies the conditional $R \to T$. Consider a simple backing for δ_a from Example 8:

1	$x : (\neg T \to \neg R)$	(Assumption)
2	$(\neg T \to \neg R) \to (R \to T)$	(A0)
3	$c : \big[(\neg T \to \neg R) \to (R \to T)\big]$	(R1)
4	$(c \cdot x) : (R \to T)$	(1,3 A1)

By taking that $u_a = (c \cdot x)$, one can recover the underlying structure of reasoning for the warrant $u_a : (R \to T)$, which corresponds to the idea of backing. Informally, the backing $(c \cdot x)$ describes reasoning when one assumes that if the table was not actually red, then it would not look red. This is a simple backing example, but, in general, such reasoning structures can become more complex. For example, assumptions made in deriving a warrant formula may include literals that are not subformulas of the warrant itself, as Example 26 later illustrates. In general, representative cases of warrants cannot be derived from a knowledge base W, without using (undischarged) assumptions. This is also the case in our Example 8, where the warrant formula $u_a : (R \to T)$ is not contained in the knowledge base closure $Th^{JT_{CS}}(W)$. Clearly, proof terms are thus interpreted more broadly than in standard justification logics.

A reader may notice here that the self-referential mechanism in which the language of justification logic treats its own reasoning steps within the language gives a three-layered understanding of arguments. The first layer is an argument seen as a pair of reason terms and formulas, e.g. the formula $(u_a \cdot t_a) : T$, resulting from the default $\delta_a = \frac{t_a : R :: (u_a \cdot t_a) : T}{(u_a \cdot t_a) : T}$. In argumentative terms, this layer includes the formula $t_a : R$ that represents Toulminian grounds or data. Since the term $(u_a \cdot t_a)$ formally realizes the default application step of δ_a, the formula $(u_a \cdot t_a) : T$ will always be explicitly featured in the semantic treatment of the acceptability of the reasoning steps codified by the term $(u_a \cdot t_a)$. Argumentation semantics for such formulas will be presented in the next section. The second layer gives a wider understanding of the argument. It includes the rule δ_a together with its warrant formula $u_a : (R \to T)$.

This layer explains the reasoning step from the grounds $t_a : R$ to the claim T. It provides an answer to Toulmin's question [77, p. 90] "How did you get there?", that is, how to justify that some claim follows from the available data or grounds. Finally, the third layer of the argument for T additionally includes the backing or the unfolded formal structure of the reasoning steps represented by u_a that are given in support of the use of the warrant $u_a : (R \to T)$. Analogously to Toulmin's argument scheme [77, p. 92], the warrant makes explicit the connection between the grounds and the claim, while the backing explains why the warrant counts as a justified one. Argument warrant can themselves become a part of the reasoning process, especially upon questioning their authority. This is illustrated by the default rule δ_b in the running example.

3.3. Argument acceptance in justification logic

By introducing default reasons in justification logic it becomes possible not only to use argumentation terminology in talking about formulas of the type $t : F$ but also to give standard abstract argumentation theory conditions of argument acceptance of such formulas. The idea of conflicting default reasons overlaps with abstract argumentation frameworks that treat conflicts between arguments. This section shows that all the formal conditions of argument acceptance as defined in Dung's framework [30] can be defined for default justifications introduced here. In Section 4, this is used to prove that the logic of default justifications generalizes Dung's frameworks.

The semantics of reason acceptance starts from characterizing conflict-free sets of $\mathbf{JT_{CS}}$ formulas. Note that by introducing default justification, conflicts are not only defined in terms of $\mathbf{JT_{CS}}$ inconsistency, but also in terms of undercut from Def 14. The following definition gives conditions for conflict-free sets with respect to undercut:

Definition 16 (Conflict-free sets)**.** A set of $\mathbf{JT_{CS}}$ formulas Γ is conflict-free iff $Th^{JT_{CS}}(\Gamma)$ does not undercut a formula $t : F$ such that $t : F \in Th^{-}(\Gamma)$.

Note that, if a set of formulas $In(\Pi)$ for any process Π is conflict-free according to Def. 16, then it is also free from rebuttal for a consistent set of formulas W. To see why, first consider that rebuttal occurs between formulas that are contained in inconsistent evidence bases. Since we know that the conditions under which a default can be applied to an evidence base preserve consistency of each segment $In(\Pi[k])$ of $In(\Pi)$, we also know that $In(\Pi)$ is rebuttal-free. Consistency preservation of extended evidence bases is established in the following theorem:

Theorem 17. *For a theory $T = (W, D)$ and a process Π of T, if the set of formulas W is $\mathbf{JT_{CS}}$ consistent, then any conflict-free set of formulas $In(\Pi)$ is also $\mathbf{JT_{CS}}$ consistent.*

Proof. The property of $\mathbf{JT_{CS}}$ consistency for a set of formulas $In(\Pi)$ follows from the applicability conditions for any default rule $\delta \in \Pi$ of the form $\frac{t:F::(u\cdot t):G}{(u\cdot t):G}$ and the fact that W is $\mathbf{JT_{CS}}$ consistent. □

The theorem ensures that, for any non-empty process Π, a set of conflict-free formulas $In(\Pi)$ that an agent could eventually accept is free from any possible conflict.

As stated before, the set W contains certain information and this means that any information from W is always acceptable regardless of what has been collected later on. Therefore, any set of formulas Γ that extends the initial information contains W. To decide whether a consequent of a default δ is acceptable, an agent looks at those sets of reasons that can be defended against all the available counter-reasons. For any set of $\mathbf{JT_{CS}}$ formulas Γ, we define the notion of acceptability of a justified formula $t : F$:

Definition 18 (Acceptability)**.** For a default theory $T = (W, D)$, a formula $t : F \in cons(\Pi)$ is acceptable w.r.t. a set of $\mathbf{JT_{CS}}$ formulas $\Gamma \subset In(\Pi[k])$ iff for each undercutting reason u for t being a reason for F such that $u : G \in In(\Pi[k])$, Γ undercuts u being a reason for G.

An agent looks at finding a defensible set of arguments in the space of all possible arguments defined by all certain information taken together with the consequents of applicable defaults. Accordingly, for a default theory $T = (W, D)$, an agent considers potential extension sets of $\mathbf{JT_{CS}}$ formulas that meet the following conditions:

(1) $W \subseteq \Gamma$ and
(2) $\Gamma \subseteq W \cup \{cons(\Pi) \mid \Pi \text{ is some process of } T\}$.

Informally, an agent has yet to test any potential extension against all the other available reasons before it can be considered as an admissible extension of the evidence base.

Definition 19 ($\mathbf{JT_{CS}}$-Admissible Extension)**.** A potential extension set of $\mathbf{JT_{CS}}$ formulas $\Gamma \subset In(\Pi)$ is a $\mathbf{JT_{CS}}$-admissible extension of a default theory $T = (W, D)$ iff Γ is conflict-free, each formula $t : F \in \Gamma$ is acceptable w.r.t. Γ and Π is closed.

After considering all the available reasons, an agent accepts only those defeasible statements that can be defended against all the available reasons against these statements.

The two latter definitions introduce the idea of "external stability" of knowledge bases [30, p. 323] into default logic by taking into account that only those reasons that are able to defend themselves against the reasons that question their plausibility eventually become accepted. In addition to that, our operational semantics prompts an implicit revision procedure. Any new default rule that is applicable to the set of formulas $In(\Pi[k])$ potentially makes changes to what an agent considered to be acceptable relying on the set of formulas $In(\Pi[k-1])$. Before we show this on the formalized example from the beginning of this section, we introduce the idea of default extension for a default theory T. Extension is the fundamental concept in defining logical consequence in standard default theories. We think of preferred extensions as maximal plausible world views based on the acceptability of reasons:

Definition 20 ($\mathbf{JT_{CS}}$-Preferred Extension)**.** For a default theory $T = (W, D)$, a closure $Th^{JT_{CS}}(\Gamma)$ of a $\mathbf{JT_{CS}}$-admissible extension Γ is a $\mathbf{JT_{CS}}$-preferred extension of T iff for any other $\mathbf{JT_{CS}}$-admissible extension Γ', $\Gamma \not\subset \Gamma'$.

In other words, $\mathbf{JT_{CS}}$-preferred extensions are maximal $\mathbf{JT_{CS}}$-admissible extensions with respect to set inclusion. The existence of $\mathbf{JT_{CS}}$-preferred extensions is universally defined for default theories. To ensure that this result also holds for the case of an infinite number of default rules and infinite closed processes, we make use of Zorn's lemma and restate it as follows:

Lemma 21 (Zorn [83])**.** *For every partially ordered set A, if every chain of (totally ordered subset of) B has an upper bound, then A has a maximal element.*

Theorem 22 (Existence of $\mathbf{JT_{CS}}$-Preferred Extension)**.** *Every default theory $T = (W, D)$ has at least one $\mathbf{JT_{CS}}$-preferred extension.*

Proof. If W is inconsistent, then for any default δ, negation of the consistency requirement $req(\delta)$ is contained in $Th^{JTCS}(W)$ and the only closed process Π is the empty sequence. Therefore, the only potential and $\mathbf{JT_{CS}}$-admissible extension is W itself and T has a unique $\mathbf{JT_{CS}}$-preferred extension $Th^{JTCS}(W)$ containing all the formulas of $\mathbf{JT_{CS}}$.

Assume that W is consistent. In general, if there is a finite number of default rules in D, any closed process Π of T is also finite. $\mathbf{JT_{CS}}$-admissible extensions obtained from closed processes form a complete partial order with respect to $\subseteq$. Since there are only finitely many $\mathbf{JT_{CS}}$-admissible sets, any $\mathbf{JT_{CS}}$-admissible set Γ has a maximum Γ' within a totally ordered subset of a set of all $\mathbf{JT_{CS}}$-admissible sets. Therefore, $\Gamma \subseteq \Gamma'$ and $Th^{JTCS}(\Gamma')$ is a $\mathbf{JT_{CS}}$-preferred extension of T.

For the case where D is infinite and closed processes $\Pi_1, \Pi_2, \ldots$ are infinite, there is again a complete partial order formed from a set of all $\mathbf{JT_{CS}}$-admissible sets. The argument for finite processes does not account for the case where Γ', the union of $\mathbf{JT_{CS}}$-admissible sets $\Gamma_1, \Gamma_2, \ldots$, could be contained in some Γ'' for an ever increasing sequence $\Gamma_1, \Gamma_2, \ldots$. We first state that Γ', the union of an ever increasing sequence of $\mathbf{JT_{CS}}$-admissible sets $\Gamma_1, \Gamma_2, \ldots$, is also a $\mathbf{JT_{CS}}$-admissible set. To ensure this, we turn to its subsets. That is, if Γ' was not admissible, then some of its subsets Γ_n for $n \geqslant 1$ would not be conflict-free or would contain a formula that is not acceptable, but this contradicts the assumption that Γ_n is $\mathbf{JT_{CS}}$-admissible. Now, for the set of all $\mathbf{JT_{CS}}$-admissible sets ordered by $\subseteq$, any chain (totally ordered subset) has an upper bound, that is, the union of its members $\Gamma' = \bigcup_{n=1}^{\infty} \Gamma_n$. According to Lemma 21, there exists a maximal element and, therefore a $\mathbf{JT_{CS}}$-preferred extension of T. □

The semantics of defeasible reasons enables us to define additional types of extensions that are not necessarily based on the admissibility of reasons. One of them is the stable extension familiar from formal argumentation theory [30]:

Definition 23 ($\mathbf{JT_{CS}}$-Stable Extension)**.** For a default theory $T = (W, D)$, a conflict-free closure $Th^{JTCS}(\Gamma)$ of a potential extension Γ is a $\mathbf{JT_{CS}}$-stable extension of T iff for any process Π of T, Γ undercuts all the formulas $t : F \in cons(\Pi)$ outside $Th^{JTCS}(\Gamma)$.

The intuition behind the definition is that every reason left outside the accepted set of reasons is attacked. To understand the process semantics workings of the stable extension definition, we can parse this definition into two components. First, it is clear that a stable extension $Th^{JTCS}(\Gamma)$ undercuts each default reason t for every $cons(\delta) = t : F$ such that $t : F$ is not contained in Γ, but δ occurs in a closed process Π of T for which it holds that Γ is a subset of the evidence base $In(\Pi)$. Intuitively, from those reasons that are applicable within a closed default process, only the reasons that are undercut by $Th^{JTCS}(\Gamma)$ are left outside. But notice that, secondly, for each default reason u and a formula $cons(\delta') = u : G$ such that Γ and $u : G$ do not co-occur in any potential extension of T, but $u : G$ is included in some potential extension of T, it holds that u also has to be undercut. This means that if δ' cannot be applied to the default process Π and δ' occurs in some other closed process Π', then Γ undercuts u. To see why, take for example the justification assertions $t : F = cons(\delta)$ and $v : \neg F = cons(\delta'')$. For any potential extension $\Gamma \subset In(\Pi)$ such that $\delta \in \Gamma$ and δ'' is not applicable to Π due to the inconsistency of the formula $req(\delta'')$, $Th^{JTCS}(\Gamma)$ contains an undercutter for the reason v. In fact, if $t : F \in \Gamma$, then $Th^{JTCS}(\Gamma)$ entails a formula $\neg r : (J \rightarrow \neg F)$ for any formula $J \in In(\Pi)$ and any reason r. Therefore, it also contains some reason term s that undercuts the warrant of the default rule

$cons(\delta'')$. This means that inconsistent justification assertions responsible for rebuttal indirectly undercut rebutted reasons. This undercut is further inherited by all the potential default reasons that are inferred from inconsistent default reasons, even if these are not involved in any rebuttal induced by $\mathbf{JT_{CS}}$ inconsistency. The following lemma generalizes this observation on the dependence between rebuttal and undercut:

Lemma 24. *For a default theory $T = (W, D)$ and its closed processes Π and Π', if some rule $\delta = \frac{t:F::(u\cdot t):G}{(u\cdot t):G}$ from Π' is inapplicable to $In(\Pi)$ and $t : F \in In(\Pi)$, then there is a potential extension $\Gamma \subset In(\Pi)$ that undercuts $(u \cdot t)$ being a reason for G.*

Proof. By Theorem 6, we know that there is some segment $In(\Pi[k])$ that contains the formula $t : F$ and, by assumption, that δ is inapplicable to $In(\Pi[k])$. Therefore, $In(\Pi[k])$ contains the formula $\neg(u \cdot t) : G$. According to axiom A1 and propositional reasoning, if the $\mathbf{JT_{CS}}$ closure $In(\Pi[k])$ contains $t : F$ and $\neg(u \cdot t) : G$, then it also contains the formula $\neg[u : (F \to G)]$. By the definition of an *In*-set (Def. 10) and the way in which potential extensions are built for T, there is some potential extension $\Gamma \subset In(\Pi[k])$ such that $Th^{JT_{CS}}(\Gamma)$ contains $\neg[u : (F \to G)]$. Since $\#(\delta) = u : (F \to G)$ and $u : (F \to G) \in \mathcal{WS}^{\Pi'}$, Γ undercuts $(u \cdot t)$ being a reason for G by Definition 15. □

If a potential extension Γ of T undercuts all the formulas left outside, then Γ also has to maximize admissibility with respect to set inclusion. This straightforwardly leads to the following lemma:

Lemma 25. *Every $\mathbf{JT_{CS}}$-stable extension of a default theory $T = (W, D)$ is also a $\mathbf{JT_{CS}}$-preferred extension of T.*

We can check that in the red-looking-table example, $\mathbf{JT_{CS}}$-stable and $\mathbf{JT_{CS}}$-preferred extension coincide. Formally, theory T_0 has a unique $\mathbf{JT_{CS}}$-stable and $\mathbf{JT_{CS}}$-preferred extension $Th^{JT_{CS}}(W_0 \cup \{cons(\delta_b), cons(\delta_c)\})$. Moreover, note that the process $\Pi_1 = (\delta_a, \delta_b)$ includes revising the resulting set of acceptable reasons, since the reason $(u_b \cdot t_b)$ undercuts $(u_a \cdot t_a)$ being a reason for formula T.

However, $\mathbf{JT_{CS}}$-stable extensions are not universally defined for any default theory T. To show this, we will formalize Pollock's "pink elephant" example [64, pp. 119–120, 66, pp. 181–182]. This example is an instance of defeasible reasoning with a self-defeating argument. The concept of self-defeat is notorious in argumentation theory. Firstly, suppose that Robert says that the elephant beside him looks pink. Normally, we would take Robert's testimony to support the conclusion that the elephant is pink. However, Robert suffers from what is known as "pink-elephant phobia". People in this condition "become strangely disoriented so that their statements about their surroundings cease to be reliable" [66, p. 181]. Therefore, it seems that "if it were true that the elephant beside Robert is pink, we could not rely upon his report to conclude that it is" [66, p. 181].

Example 26. Let P be the proposition "The elephant looks pink", let E be the proposition "The elephant is pink", and let H be the proposition "Robert suffers from pink-elephant phobia". The pink elephant example is then described by the default theory $T_1 = (W_1, D_1)$, where $W_1 = \{k : H, l : P\}$ and D_1

consists of the default rules[24]

$$\delta_1 = \frac{l : P :: (m \cdot l) : E}{(m \cdot l) : E} \quad \text{and}$$

$$\delta_2 = \frac{(m \cdot l) : E :: (n \cdot (m \cdot l)) : \neg[m : (P \to E)]}{(n \cdot (m \cdot l)) : \neg[m : (P \to E)]}.$$

While the structure of the backing for δ_1 resembles that of δ_a from Example 8, the backing for the default rule δ_2 has a more intricate structure:

1	$x : \big[m : (P \to E) \to \neg(E \wedge H)\big]$	(Assumption)
2	$k : H$	(Assumption)
3	$\big[m : (P \to E) \to \neg(E \wedge H)\big] \to \big[(H \to \big(E \to \neg\big[m : (P \to E)\big]\big)\big]$	(A0)
4	$c : (\big[m : (P \to E) \to \neg(E \wedge H)\big] \to \big[\big(H \to \big(E \to \neg\big[m : (P \to E)\big]\big)\big]\big)$	(R1)
5	$(c \cdot x) : \big[(H \to \big(E \to \neg\big[m : (P \to E)\big]\big)\big]$	(1,4 A1)
6	$\big((c \cdot x) \cdot k\big) : \big(E \to \neg\big[m : (P \to E)\big]\big)$	(2,5 A1)

Let $n = ((c \cdot x) \cdot k)$. The above inference steps in $\mathbf{JT_{CS}}$ formalize the backing for the warrant $n : (E \to \neg[m : (P \to E)])$ of δ_2. Notice that, in the formalization of its backing, the warrant of δ_2 is supported by appeal to the presupposed information about the phobia that Robert suffers from, that is, to the justification assertion $k : H$.

The theory T_1 has a $\mathbf{JT_{CS}}$-preferred extension $Th^{JT_{CS}}(W_1)$. However, it has no $\mathbf{JT_{CS}}$-stable extension, because the available reasons cannot form a conflict-free set that attacks all the reasons outside that set. This result conforms to similar results about preferred and stable semantics in abstract argumentation frameworks [30, p. 328]. By the end of the section, we define the theory T_3 that shows the same type of a self-defeating argument alongside other arguments. In our default theories, self-defeating arguments do not influence other independent arguments, except in the above-illustrated sense of affecting the existence of stable semantics.

In addition, we can easily define other significant notions of extensions in formal argumentation. In particular, we can define variants of Dung's [30, p. 329] *complete* and *grounded* extension:

[24]Notice that in the original formulation of his pink elephant example, Pollock introduces [64, p. 120] an intermediate inference between the rules δ_1 and δ_2. Namely, he thinks that there is an inference from Robert's saying (reason term l) that the elephant looks pink to him, to the conclusion that it *does* look pink. We follow a version of the example that does not take the intermediate step as a separate inference, taken from [50, §4.1]. There are two reasons for this decision. Firstly, Pollock's red table example that we formalized in Example 8 has the same structure of inference that starts from seeing a red-looking table to conclude that the table is red. There is no mention of the table looking red independently of an agent's report that it does. It is not clear why to think that Robert's unreliability in the presence of pink elephants would question the fact that the elephant does look pink, even if Robert himself realizes that he suffers from the phobia. It is also not clear what would it mean for an object to look pink, regardless of being perceived as pink by some agent. Secondly, a report of another agent to whom the elephant does not look pink would be treated differently in justification logic. Such report would undermine Robert's own report and the subject matter of undermining attacks is dealt with in another paper [62], together with the topic of how to model testimonies. In any case, an intermediate default rule could formally be added without affecting the significance of the example for the discussion.

Definition 27 (**JT$_{\mathbf{CS}}$**-Complete Extension)**.** For a default theory $T = (W, D)$, a closure $Th^{JT_{CS}}(\Gamma)$ of a **JT$_{\mathbf{CS}}$**-admissible extension Γ is a **JT$_{\mathbf{CS}}$**-complete extension of T iff for each closed process Π of T such that there is a **JT$_{\mathbf{CS}}$**-admissible extension Γ' in $In(\Pi)$ and $\Gamma \subset \Gamma'$, if a formula $t : F \in cons(D)$ is acceptable w.r.t. Γ in $In(\Pi)$, then $t : F$ belongs to Γ.

Definition 28 (**JT$_{\mathbf{CS}}$**-Grounded Extension)**.** For a default theory $T = (W, D)$, a **JT$_{\mathbf{CS}}$**-complete extension $Th^{JT_{CS}}(\Gamma)$ is the unique **JT$_{\mathbf{CS}}$**-grounded extension if Γ is the smallest potential extension with respect to set inclusion such that $Th^{JT_{CS}}(\Gamma)$ is a **JT$_{\mathbf{CS}}$**-complete extension of T.[25]

Unsurprisingly, the results for different types of extensions from [30] are valid for our default theory extensions.

Lemma 29. *Every* **JT$_{\mathbf{CS}}$***-preferred extension of a default theory* $T = (W, D)$ *is also a* **JT$_{\mathbf{CS}}$***-complete extension of* T.

Proof. Assume that $Th^{JT_{CS}}(\Gamma)$ is a **JT$_{\mathbf{CS}}$**-preferred extension of T for some potential extension Γ. Assume towards contradiction that for some closed process Π such that $\Gamma \subset In(\Pi)$ and Γ is **JT$_{\mathbf{CS}}$**-admissible there exists a formula $cons(\delta)$, where $\delta \in \Pi$, acceptable with respect to Γ, but not included in Γ. According to Def. 19, there is a **JT$_{\mathbf{CS}}$**-admissible extension Γ' for which it holds that $\Gamma \subset \Gamma'$. But this contradicts the assumption that $Th^{JT_{CS}}(\Gamma)$ is a **JT$_{\mathbf{CS}}$**-preferred extension. Therefore, for any closed process Π' for which Γ is **JT$_{\mathbf{CS}}$**-admissible and for any formula $cons(\delta')$ such that $\delta' \in \Pi'$, if $cons(\delta')$ is acceptable with respect to Γ, then $cons(\delta')$ is included in Γ. □

It does not hold, however, that every **JT$_{\mathbf{CS}}$**-complete extension is also **JT$_{\mathbf{CS}}$**-preferred. The following theory T_2 is a counterexample. Let the theory be defined as $T_2 = (W_2, D_2)$, where $W_2 = \{p : K, q : L\}$ and D_2 consists of the default rules

$$\delta_3 = \frac{p : K :: (r \cdot p) : M}{(r \cdot p) : M} \quad \text{and}$$

$$\delta_4 = \frac{q : L :: (s \cdot q) : \neg M}{(s \cdot q) : \neg M}.$$

One of the **JT$_{\mathbf{CS}}$**-complete extensions of T_2 is $Th^{JT_{CS}}(W_2)$, as a result of the fact that none of the available default reasons is acceptable with respect to the potential extension W_2. However, $Th^{JT_{CS}}(W_2)$ is not one of **JT$_{\mathbf{CS}}$**-preferred extensions for T_2. The theory has two **JT$_{\mathbf{CS}}$**-preferred extensions such that one of them contains $cons(\delta_3)$, while the other contains $cons(\delta_4)$.

Considering some proposition as justified might be seen as a function of interacting reasons. Each of the presented **JT$_{\mathbf{CS}}$** extensions is a method to compute extensions with justified formulas. Moreover, each of the **JT$_{\mathbf{CS}}$** extension definitions can be used as a way to define a corresponding characterization of logical consequence. Given a particular **JT$_{\mathbf{CS}}$** extension of a theory T, the formulas contained in that extension are valid formulas for T under that specific **JT$_{\mathbf{CS}}$** semantics. There are some analogies with

[25] Note here that the we know that there is the smallest potential extension which is **JT$_{\mathbf{CS}}$**-complete since we can represent **JT$_{\mathbf{CS}}$**-admissible extensions as forming a complete partial order w.r.t. set inclusion. Ordered extensions lend themselves to a fixed-point reformulation of all admissibility-based extensions and a possibility of guaranteeing the existence of the smallest potential extension by the application of the Knaster-Tarski theorem [75].

the traditional notions of non-monotonic consequence relations. For example, $\mathbf{JT_{CS}}$-grounded extensions correspond to *cautious* consequence relations describing what a skeptical reasoner would accept for some default theory. In a similar way, $\mathbf{JT_{CS}}$-preferred semantics describes a *credulous* inference relation. The consequence relation defined by $\mathbf{JT_{CS}}$-stable extension is an interesting case in this context. Although for many default theories $\mathbf{JT_{CS}}$-stable and $\mathbf{JT_{CS}}$-preferred semantics coincides, there are some intuitive grounds to consider $\mathbf{JT_{CS}}$-stable extensions as skeptical in nature. This specifically relates to the demand that the existence of $\mathbf{JT_{CS}}$-stable extensions depends on whether a set of $\mathbf{JT_{CS}}$ formulas is able to defeat all other reasons outside that set or not. Such excessive demands on the validity of formulas do not comply to our ordinary intuitions about credulous consequence relations.

To illustrate the differences among the above defined semantics, we will elaborate on an example of a single default theory whose $\mathbf{JT_{CS}}$-grounded, $\mathbf{JT_{CS}}$-complete, $\mathbf{JT_{CS}}$-preferred and $\mathbf{JT_{CS}}$-stable extensions do not coincide, although each of them exists. We define the default theory $T_3 = (W_3, D_3)$ with $W_3 = \{t_1 : F, t_2 : H, t_3 : I\}$ and $D_3 = \{\delta_5, \delta_6, \delta_7, \delta_8\}$, where δ_5, δ_6, δ_7 and δ_8 are defined as follows:

$$\delta_5 = \frac{t_1 : F :: (u_1 \cdot t_1) : G}{(u_1 \cdot t_1) : G},$$

$$\delta_6 = \frac{(u_1 \cdot t_1) : G :: (u_2 \cdot (u_1 \cdot t_1)) : \neg[u_1 : (F \rightarrow G)]}{(u_2 \cdot (u_1 \cdot t_1)) : \neg[u_1 : (F \rightarrow G)]},$$

$$\delta_7 = \frac{t_2 : H :: (u_3 \cdot t_2) : (J \wedge \neg[u_2 : (G \rightarrow \neg[u_1 : (F \rightarrow G)])])}{(u_3 \cdot t_2) : (J \wedge \neg[u_2 : (G \rightarrow \neg[u_1 : (F \rightarrow G)])])} \quad \text{and}$$

$$\delta_8 = \frac{t_3 : I :: (u_4 \cdot t_3) : \neg J}{(u_4 \cdot t_3) : \neg J}.$$

Any evidence base $In(\Pi)$ of T_3 containing the formula $cons(\delta_7)$ will also contain the formula $(c_1 \cdot (u_3 \cdot t_2)) : \neg[u_2 : (G \rightarrow \neg[u_1 : (F \rightarrow G)])]$, which represents the reasoning behind an argument that questions the warrant of the self-defeating argument given in δ_6 by undercutting $(u_2 \cdot (u_1 \cdot t_1))$. The undercutter $(c_1 \cdot (u_3 \cdot t_2))$ can be derived from $cons(\delta_7)$ with some propositional reasoning combined with the use of axiom A1 and rule R1*. Moreover, default δ_7 provides an argument that rebuts the reason $(u_4 \cdot t_3)$ for $\neg J$, for any extension that contains $cons(\delta_7)$. This argument is codified within the term $(c_2 \cdot (u_3 \cdot t_2))$ justifying the formula J, again assuming some propositional reasoning, axiom A1 and rule R1*. Accordingly, the rules δ_7 and δ_8 cannot occur together in any default process of T_3.

In total, the theory T_3 has six closed processes, as shown in the process tree of T_3 displayed in Fig. 4. Building a process tree for our default theories proceeds in the following way: each node of the process tree is labeled with an *In*-set after a default rule (connecting edges) has been applied. Note that, for each node of the process tree in Fig. 4 and a closed process Π of T_3, if a node corresponds to some segment $\Pi[k]$ of Π we indicate only the formula that has been added to $In(\Pi[k])$ as a result of applying an available default rule to $In(\Pi([k-1])$. The process tree helps us to check the status of $\mathbf{JT_{CS}}$ extensions for T_3. The theory has two preferred extensions, namely $Th^{JT_{CS}}(W_3 \cup \{cons(\delta_5), cons(\delta_7)\})$ and $Th^{JT_{CS}}(W_3 \cup \{cons(\delta_8)\}$. Of the two $\mathbf{JT_{CS}}$-preferred extensions, only $Th^{JT_{CS}}(W_3 \cup \{cons(\delta_5), cons(\delta_7)\})$ is also $\mathbf{JT_{CS}}$-stable. A skeptical reasoner will only accept $Th^{JT_{CS}}(W_3)$, the unique $\mathbf{JT_{CS}}$-grounded extension of T_3. Finally, all the three mentioned $\mathbf{JT_{CS}}$ closures are $\mathbf{JT_{CS}}$-complete for T_3.

It is possible to specify conditions under which different $\mathbf{JT_{CS}}$ extension notions above coincide. Sufficient conditions need to eliminate the possibility of attack cycles. We first define the cycle of asymmetrical attacks:

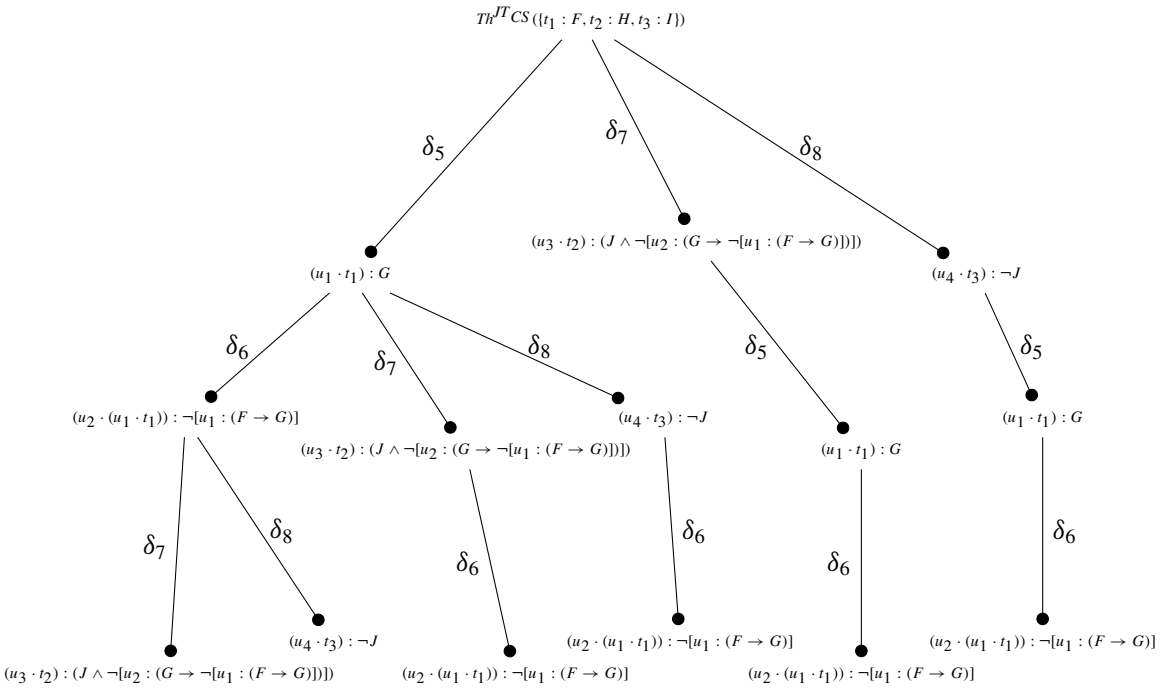

Fig. 4. The process tree of T_3.

Definition 30 (Undercut Cycle)**.** A cycle of undercuts is an infinite periodic sequence of $\mathbf{JT_{CS}}$ formulas $t_1 : F_1, \ldots, t_n : F_n, t_1 : F_1, \ldots, t_n : F_n, t_1 : F_1, \ldots$, for some number of formulas $n \geqslant 1$, such that each reason t_i undercuts t_k being a reason for the formula F_k according to Def. 14 and $t_i : F_i$ is the predecessor of the formula $t_k : F_k$ in the sequence.

Rebuttals among formulas ultimately derive from the property of $\mathbf{JT_{CS}}$ inconsistency. They are thus symmetric and can be traced through the process semantics and existence of different evidence bases $In(\Pi')$ and $In(\Pi'')$ for some closed processes Π and Π'. Therefore, we do not need to define rebuttal separately, but only provide a condition that excludes attacks induced by $\mathbf{JT_{CS}}$ inconsistency.

We are ready now to give the conditions for the coincidence of $\mathbf{JT_{CS}}$ extensions in *well-founded* default theories. A default theory $T = (W, D)$ is called well-founded if for all closed processes Π and Π' of T it holds that:

(1) $In(\Pi) = In(\Pi')$ and
(2) There are no sets of $\mathbf{JT_{CS}}$ formulas $\Gamma \in In(\Pi)$ forming a cycle of undercuts.

The following theorem shows that $\mathbf{JT_{CS}}$ extensions of well-founded default theories coincide.[26]

Theorem 31. *Every well-founded default theory $T = (W, D)$ has a unique $\mathbf{JT_{CS}}$-complete extension $Th^{JTCS}(\Gamma)$ which is $\mathbf{JT_{CS}}$-grounded, $\mathbf{JT_{CS}}$-preferred and $\mathbf{JT_{CS}}$-stable.*

[26]Compare [30, p. 331] for well-foundedness of abstract argumentation frameworks. Here we adapt the proof idea for the coincidence of extensions of well-founded abstract argumentation frameworks that can be found there.

Proof. Firstly, if a $\mathbf{JT_{CS}}$-grounded extension is also a $\mathbf{JT_{CS}}$-stable extension of a default theory $T = (W, D)$, then it is also $\mathbf{JT_{CS}}$-preferred and the unique $\mathbf{JT_{CS}}$-complete extension of T. Therefore, it is sufficient to focus on the proof that each $\mathbf{JT_{CS}}$-grounded extension is $\mathbf{JT_{CS}}$-stable for a well-founded theory.

Assume that a well-founded theory T has a $\mathbf{JT_{CS}}$-grounded extension $Th^{JT_{CS}}(\Gamma)$ that is not $\mathbf{JT_{CS}}$-stable. The set $\Gamma \subset In(\Pi)$ is the smallest potential extension such that $Th^{JT_{CS}}(\Gamma)$ is a $\mathbf{JT_{CS}}$-complete extension of T. Moreover, there is at least one formula $t : F \in cons(\delta)$ from the set of consequents $cons(D)$ such that $t : F \notin Th^{JT_{CS}}(\Gamma)$, but, since $Th^{JT_{CS}}(\Gamma)$ is not $\mathbf{JT_{CS}}$-stable, $Th^{JT_{CS}}(\Gamma)$ does not undercut t being a reason for F. Now we have to show that unless $Th^{JT_{CS}}(\Gamma)$ undercuts t being a reason for F, at least one of the following statements has to hold about T:

(1) $t : F$ is acceptable w.r.t. Γ in $In(\Pi)$, but Γ is a subset of $In(\Pi')$ for some other closed process Π' and $t : F$ is not acceptable w.r.t. Γ in $In(\Pi')$. But this means that the sets $In(\Pi)$ and $In(\Pi')$, which, in turn, means that T is not well-founded according to condition (1) on well-founded default theories;

(2) $t : F$ is not acceptable w.r.t. Γ in $In(\Pi)$ and there is some formula $v : G \in In(\Pi)$ such that v undercuts t being a reason for F, but $Th^{JT_{CS}}(\Gamma)$ does not undercut v being a reason for G. However, $v : G$ is not contained in Γ, since we assumed that Γ is not $\mathbf{JT_{CS}}$-stable and that Γ does not undercut t being a reason for F. But this means that there exists an infinite periodic sequence of $\mathbf{JT_{CS}}$ formulas $t_1 : F_1, \ldots, t_n : F_n, t_1 : F_1, \ldots, t_n : F_n, t_1 : F_1, \ldots$ forming an undercut cycle according to Def. 30. This means that T is not well-founded according to condition (2).

Therefore, since T is well-founded, it has a unique $\mathbf{JT_{CS}}$-complete extension $Th^{JT_{CS}}(\Gamma)$ which is $\mathbf{JT_{CS}}$-grounded, $\mathbf{JT_{CS}}$-preferred and $\mathbf{JT_{CS}}$-stable. □

4. Relations of the logic of default justifications to abstract argumentation frameworks: Realizing Dung's frameworks in justification logic

Abstract argumentation frameworks (AF) inquire into the problem of the acceptability of arguments based on their mutual conflicts. More precisely, an argumentation framework is a pair of a set of arguments, and a binary relation representing the attack-relationship (defeat) between arguments. These frameworks are abstract in at least two ways: they neither represent the structure of arguments nor do they specify the exact nature of attacks between them. The study of abstract arguments was initiated in [30]. From then on, there have been many attempts to develop frameworks where the structure of arguments is included, most notably in the ASPIC+ framework [68].

In this section we examine connections between abstract argumentation frameworks and our logic. The semantics of justification formulas $t : F$ we introduced can be naturally related to the concepts of argumentation semantics. Any justification formula can be plausibly regarded as an argument where t codifies premises and F is a conclusion of an argument.[27] However, the expressiveness of the language $\mathbf{JT_{CS}}$ enables us to construct complex argument structures resulting from logical operations on formulas. As expected, abstract argumentation frameworks are not able to capture all the subtleties of more complex default reasons. Interestingly, it turns out that there are also AF structures that cannot be translated into default theories.

[27] We can say this also about the formula $c : F$, where c is a proof constant, but in this case the attack relation will be empty.

We first focus on the possibility of mapping from default theories to AFs. To establish the connection between default reasons semantics and AF semantics, we need to restrict our attention to a subclass of our default theories. Since our logic is more expressive with respect to attack relations, we focus on non-complex default theories where attack relations are defined only by looking at the union of logical consequences of each consequent of a default rule. In this way, each default rule is taken separately as a self-contained argument. To achieve this, we first specify what it means for two default rules to *block* each other's applicability. For a process Π of $T = (W, D)$, the rules δ and δ' from D block each other in Π iff for some segment $\Pi[k]$ such that both δ and δ' are applicable to $In(\Pi[k])$, if either of the two defaults has been applied, the other default becomes inapplicable to $In(\Pi[k+1])$. A default theory $T = (W, D)$ is non-complex if it fulfills the following two conditions:

(1) If two defaults δ and δ' from D block each other in a process Π of T, then for each process Π' with a segment $\Pi'[k]$ such that either δ or δ' has been applied to $In(\Pi'[k])$ it holds that the default that has not been applied to $In(\Pi'[k])$ is inapplicable to $In(\Pi'[k+n])$ for any segment $\Pi'[k+n]$ of Π';
(2) For a process Π of T, a reason t such that $t : F \in In(\Pi)$ and any undercutter u for t such that $u : \neg[v : (G \rightarrow H)] \in In(\Pi)$ for some $v \in Sub(t)$, there exists a reason $w \in Sub(u)$ such that $w : \neg[v : (G \rightarrow H)] \in Th^{JT_{CS}}(cons(\delta))$ for a default rule $\delta \in \Pi$.

In other words, we require for any defeat that occurs in a theory T to be derivable only from a consequent of a default rule because joint attacks cannot be represented in Dung's [30] framework.

Using default justifications, one can look into the details of arguments' structure, including grounds, warrants, backings and different ways of attack, while Dung's framework treats arguments abstracting from their contents. This means that any translation from default theories with justification terms to Dung's framework has to "forget" information about arguments' structure. Having restricted our target theories to non-complex theories, we can now describe a mapping "$\Longrightarrow$" called *Forgetful projection*. Forgetful projection converts each formula $cons(\delta)$ such that δ occurs in some process of a given default theory into a corresponding argument of a Dung's framework and it converts each attack among default reasons into a corresponding attack relation between Dung's arguments. A mapping $\Longrightarrow$ from a non-complex default theory $T = (W, D)$ to an abstract argumentation framework $Af = (Arg, Att)$, where Arg is a set of arguments $A_1, A_2, \ldots$ and Att is a a binary attack relation, is defined as follows:

- $\delta_n \in \Pi$ for a process $\Pi \Longrightarrow A_n \in Arg$
- $\delta_m \in \Pi'$ & $\delta_n \in \Pi''$ for some processes Π' & Π'' such that δ_m & δ_n do not occur together in any process $\Pi \Longrightarrow (A_m, A_n) \in Att$ & $(A_n, A_m) \in Att$
- $t : \neg[u : (F \rightarrow G)] \in Th^{JT_{CS}}(cons(\delta_m))$, $v : H = cons(\delta_n)$ such that $u \in Sub(v)$ & $u : (F \rightarrow G) \in \mathcal{WS}^{\Pi}$ and $\delta_m \in \Pi$ & $\delta_n \in \Pi \Longrightarrow (A_m, A_n) \in Att$

Recall the theory T_0 described in Example 8. The theory T_0 has its forgetful projection AF_0 that preserves the direction of the attacks from the original example. Consider that each of the rules δ_a, δ_b and δ_c is applicable to at least one process. This means that we can map all three defaults to the arguments A_a, A_b and A_c in Arg_0. Given that δ_a and δ_c cannot be applied to the same process of T_0 and given the fact that they are applicable to some processes, both $(A_a, A_c) \in Att_0$ and (A_c, A_a) are in Att_0. Finally, notice that the rules δ_b and δ_a can be applied together in a default process and that the reason $(u_b \cdot t_b)$ undercuts $(u_a \cdot t_a)$ via justifying the denial of the warrant $u_a : (R \rightarrow T)$ of δ_a. Forgetful projection maps this relation between $cons(\delta_b)$ and $cons(\delta_a)$ into an additional attack (A_b, A_a) in Att_0.

Since forgetful projection does preserve the structure of conflicts among groups of arguments, it is possible to compare $\mathbf{JT_{CS}}$ extensions of default theories with extensions of the obtained Dung's frameworks. It is not difficult to check that the following extension-correspondence statement holds:

Proposition 32. *For a formula $t : F = cons(\delta_n)$ such that $\delta_n \in D$ for a non-complex default theory $T = (W, D)$ and its* $\mathbf{JT_{CS}}$*-complete,* $\mathbf{JT_{CS}}$*-grounded,* $\mathbf{JT_{CS}}$*-preferred or* $\mathbf{JT_{CS}}$*-stable extension $Th^{JT_{CS}}(\Gamma)$, it holds that $t : F \in Th^{JT_{CS}}(\Gamma)$ iff an argument A_n is contained in the corresponding complete, grounded, preferred or stable extension sets for a forgetful projection $Af = (Arg, Att)$ of T.*

Proof. See the Appendix for a proof sketch. □

Intuitively, forgetful projections of justification logic arguments outline a single perspective on argumentation, namely that of opposition among arguments. Note that there are extensions of Dung's framework that formalize joint attacks from sets of arguments such as [58]. In a framework with joint attacks, Proposition 32 can be generalized to any default theory with justification formulas.

One may also ask whether the other direction of translating from argumentation frameworks to default theories always works. Since the content of arguments is not specified in Dung's framework, it is only possible to retrieve incomplete information about justification logic counterparts of Dung's frameworks. For any argument in Dung's framework, there are many justification logic realizations. Starting from a directed graph obtained from a framework $Af = (Arg, Att)$, each node $A_1, \ldots, A_n$ is paired with a corresponding formula $t_1 : F_1, \ldots, t_n : F_n$, where each $t_i : F_i$ is a consequent of some rule δ_i such that δ_i occurs in at least one process of a theory $T = (W, D)$ that realizes Af. Moreover, each node A_i is paired with a warrant $u_i : (G_i \to F_i)$.

The algorithm treats every single arrow in Dung's graph as a specification of $\mathbf{JT_{CS}}$ entailments that hold for default consequents paired with the nodes of a graph. Accordingly, we determine the structure of attacks among the obtained formulas. More specifically, a pointed arrow without an inverted arrow specifies that a default consequent formula, which realizes a direct predecessor for the arrow, entails an undercut formula for the consequent formula via entailing the negation of a warrant that realizes the successor node. An arrow with an inverted arrow specifies inconsistency for consequent formulas paired with the connected nodes, that is, a rebuttal between the two formulas.[28] Using this algorithm, we would get information on which formulas should a default consequent formula entail with respect to other default consequents, provided the definition of attack relations among arguments in Arg. However, the algorithm fails as the following example shows. Take a simple framework $Af^* = (Arg, Att)$ with A as its only argument and $(A, A) \in Att$. It turns out that it is not possible to realize A as a single consequent of a default rule.

The problem can be generalized to a class of *unwarranted* argumentation frameworks. An argumentation framework $Af = (Arg, Att)$ is said to be unwarranted iff:

(1) There is an infinite sequence $A_1, A_2, \ldots, A_n, \ldots$ such that for each i, A_{i+1} attacks A_i;
(2) For any two distinct arguments $A = A_k$ and $B = A_{k+1}$ such that A_k and A_{k+1} are adjacent members of the $A_1, A_2, \ldots, A_n, \ldots$ sequence, it does not hold that $(A, B) \in Att$ and $(B, A) \in Att$;
(3) There exists no argument C outside the sequence such that:
   (a) for some A from the sequence $A_1, A_2, \ldots, A_n, \ldots$ it holds that $(A, C) \in Att$;
   (b) C is not a member of an infinite sequence $B_1, B_2, \ldots, B_n, \ldots$ such that for each i, B_{i+1} attacks B_i;
   (c) for no two distinct arguments D and E from Arg it holds that $(D, C) \in Att$ and $(E, C) \in Att$.

[28] It is possible that an obtained formula has a complex structure and, for example, entails both a rebutting and an undercutting reason for some formulas.

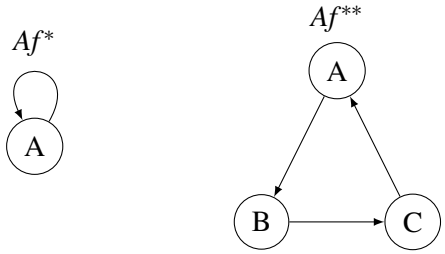

Fig. 5. Unwarranted argumentation framework examples.

The conditions above eliminate realizations of a small subclass of graphs with "floating" cycles, but they do not eliminate the possibility to realize cycles of attacks in general. In the abstract argumentation [16] and defeasible reasoning [65] literature, only the semantics of odd-length cycles of attacks (or of defeats) is notorious for undesirable properties that odd-length cycles entail for different types of extensions. In our default reason theory, both odd- and even-length "floating-attack" cycles have no direct counterparts. This will be explained below in details.

Informally, we can say that such unwarranted frameworks violate the following postulate for structured argumentation frameworks:

- Prior to any challenge there must be at least one reasoned claim.

From the perspective of our default theory, the frameworks Af^* and Af^{**} represented in Fig. 5 are impossible. If precisely assessed, their status of *argumentation* frameworks can be attributed to the possibility to abstract from argument structure in Dung's model.

Once additional argument features are considered, and in particular arguments' warrants, the structures from Fig. 5 can be proved to be impossible. The following theorem shows that, in our default theories, floating-attack cycles without at least one outgoing edge to an argument outside the cycle are not possible.

Theorem 33. *For a sequence $In(\Pi)[k]$ of a default theory $T = (W, D)$ and a set of formulas $\{t_1 : F_1, \ldots, t_n : F_n\} \in In(\Pi)[k]$, a cycle of undercuts among the reasons $t_1, \ldots, t_n$ is possible only if (1) there is a reason t_i for a formula F_i, where $1 \leqslant i \leqslant n$, such that one of its subterms $p \in Sub(t_i)$ for a warrant $p : (B \to C) \in \mathcal{WS}^{\Pi[k]}$ is not undercut by any of the reasons from the cycle $t_1, \ldots t_n$ and (2) there is a warrant $r : (D \to E) \in \mathcal{WS}^{\Pi[k]}$, such that $r \in Sub(t_i)$ and r is undercut by some reason from the cycle $t_1, \ldots t_n$, but none of the other warrants from $\mathcal{WS}^{\Pi[k]}$ is a subformula of E.*

Proof. Assume that there is a cycle of undercuts in a set of formulas $In(\Pi)[k]$ among reasons $t_1, \ldots, t_n$, such that each t_i, where $2 \leqslant i \leqslant n$, is undercut by t_{i-1} as a reason for F_i and that t_1 is undercut by t_n as a reason for F_1. By Definition 14, for each reason t_i and each formula $t_i : F_i$ from a set of formulas $\{t_2 : F_1, \ldots, t_n : F_n\} \in In(\Pi)[k]$, there is a subterm $s \in Sub(t_i)$ such that $t_{i-1} : \neg[s : (G \to H)]$ and for the formula $t_1 : F_1 \in In(\Pi)[k]$ and a subterm $u \in Sub(t_1)$, it holds that $t_n : \neg[u : (I \to J)]$. Then assume that each reason term from the set $\{v \mid v \in \bigcup_{j=1}^{n} Sub(t_j)$ and $v : (K \to L) \in \mathcal{WS}^{\Pi[k]}\}$, is undercut in the cycle of undercuts $t_1, \ldots t_n$. This means that each warrant $v : (K \to L)$ for $v \in Sub(t_k)$ and $2 \leqslant k \leqslant n$ would have to be a proper subformula of a formula $t_{k-1} : F_{k-1}$ from the cycle such that $t_{k-1} : \neg[v : (K \to L)]$ and, thereby, t_{k-1} undercuts t_k being a reason for formula F_k. Additionally, the warrant $w : (M \to N)$ for $w \in Sub(t_1)$ would have to be a proper subformula of the formula $t_n : F_n$

such that $t_n : \neg[w : (M \to N)]$ and, thereby, t_n undercuts t_1 being a reason for formula F_1. But this is not possible since no formula is a proper subformula of itself. Therefore, at least one reason t_k from the cycle of undercuts $t_1, \ldots, t_n$ has to attack a warrant $r : (O \to P) \in \mathcal{WS}^{\Pi[k]}$, where $r \in Sub(t_k)$ and $1 \leqslant k \leqslant n$, such that none of the other warrants from $\mathcal{WS}^{\Pi[k]}$ is a subformula of P. □

The theorem ensures that cycles of asymmetrical attacks among arguments are possible only if there is an outlying argument and this argument is attacked by an argument in the cycle. Although our justification logic cannot realize the subclass of unwarranted frameworks, this result does not exclude circular argumentation from it in general. However, the result does show that there are constraints on interpreting directed graphs as argumentation frameworks and these constraints are due to the inclusion of additional argument features into our system.

In the literature about abstract argumentation frameworks, there are attempts to provide frameworks Af^* and Af^{**} with intuitive interpretations. For example, [79, p. 630] give the following sports situation as an informal interpretation of Af^{**}. Imagine that Ajax has recently won matches against Feyenoord. We have a reason to think that Ajax is the best Dutch football club (argument A). But assume that it is also the case that Feyenoord has won recent matches against PSV and that PSV has won recent matches against Ajax. Then we have a reason to think that Feyenoord is the best Dutch club (argument B) and that PSV is the best Dutch club (argument C). The available arguments leave us with no answer to the question which football club is the best.

By fleshing out the content of these arguments in our default theory, it becomes clear that there is more to this example than the cycle of three attacks is able to show. There are two kinds of arguments involved in resolving the conflict among the claims to the status of the best club. First, the fact that Ajax has won recent matches against Feyenoord, provides a reason to claim that Ajax is the best club. Secondly, the same fact provides grounds to question the claim that Feyenoord is the best club. The first argument can be an attacker only as a rebuttal, while the second argument is an undercutter. Analogously, arguments can be provided with reference to Feyenoord and PSV, as we will formalize below.

Example 34. Let $T_4 = (W_4, D_4)$, be the default theory describing the conflict of football clubs. The set of facts is defined by $W_4 = \{t_1 : A_1, t_2 : F_1, t_3 : P_1, t_4 : [\neg(A_2 \wedge F_2) \wedge \neg(A_2 \wedge P_2) \wedge \neg(F_2 \wedge P_2)]\}$, where A_1, F_1 and P_1 are the propositions "Ajax/Feyenoord/PSV has won recent matches against Feyenoord/PSV/Ajax" and A_2, F_2 and P_2 are the propositions "Ajax/Feyenoord/PSV is the best Dutch football club". Notice that the set of facts contains a formula which corresponds to the background knowledge that only one club can be the best club. Finally, $D_4 = \{\delta_9, \delta_{10}, \delta_{11}, \delta_{12}, \delta_{13}, \delta_{14}\}$ is the set of defaults, where

$$\delta_9 = \frac{t_1 : A_1 :: (u_1 \cdot t_1) : A_2}{(u_1 \cdot t_1) : A_2}, \qquad \delta_{10} = \frac{t_2 : F_1 :: (u_2 \cdot t_2) : F_2}{(u_2 \cdot t_2) : F_2},$$

$$\delta_{11} = \frac{t_3 : P_1 :: (u_3 \cdot t_3) : P_2}{(u_3 \cdot t_3) : P_2}, \qquad \delta_{12} = \frac{t_1 : A_1 :: (u_4 \cdot t_1) : \neg[u_2 : (F_1 \to F_2)]}{(u_4 \cdot t_1) : \neg[u_2 : (F_1 \to F_2)]},$$

$$\delta_{13} = \frac{t_2 : F_1 :: (u_5 \cdot t_2) : \neg[u_3 : (P_1 \to P_2)]}{(u_5 \cdot t_2) : \neg[u_3 : (P_1 \to P_2)]} \quad \text{and}$$

$$\delta_{14} = \frac{t_3 : P_1 :: (u_6 \cdot t_3) : \neg[u_1 : (A_1 \to A_2)]}{(u_6 \cdot t_3) : \neg[u_1 : (A_1 \to A_2)]}.$$

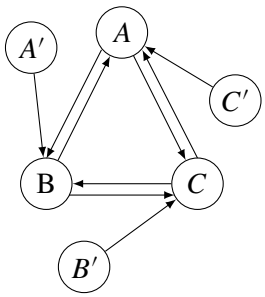

Fig. 6. Abstract attack structure of Example 34.

It is easy to check that theory T_4 has a unique $\mathbf{JT_{CS}}$-stable and $\mathbf{JT_{CS}}$-preferred extension $Th^{JT_{CS}}(W_4 \cup \{cons(\delta_{12}), cons(\delta_{13}), cons(\delta_{14})\})$. Therefore, the conflict between Dutch football clubs results in accepting that the available reasons do not sanction any of the three Dutch football clubs to claim the title of the best club.

Theory T_4 shows that Af^{**} misrepresents the conflict of Dutch football clubs. A more faithful abstract argumentation framework should include additional arguments and attack relations as Fig. 6 shows. The only accepted arguments are the additional arguments A', B' and C' that are not featured in Af^{**}. These three arguments ensure that none of the unjustified claims to the title of the best Dutch football club goes through. Note how the arguments that are eventually accepted as winning are those indicating the inability of reasons and warrants to justify claims – this layer of argumentation has been so far elusive to a strict logical formalization.

By excluding unwarranted Dung's frameworks, it is possible to formalize the *Realization* procedure ("$\longrightarrow$") of warranted Dung's frameworks in justification logic. For a warranted abstract argumentation framework $Af = (Arg, Att)$, there is a default theory $T = (W, D)$ such that:

- $A \in Arg \longrightarrow t : F = cons(\delta)$ and $u : (G \to F) \in \mathcal{WS}^{\Pi}$ s. t. $\delta \in \Pi$ for a process Π of T
- $(A_m, A_n) \in Att$ & $(A_n, A_m) \in Att \longrightarrow t : P \in Th^{JT_{CS}}(cons(\delta_m))$ and $u : \neg P \in Th^{JT_{CS}}(cons(\delta_n))$ s. t. P is a fresh propositional variable and, for some processes Π' and Π'' of T, $\delta_m \in \Pi'$ and $\delta_n \in \Pi''$
- $(A_m, A_n) \in Att$ & $(A_n, A_m) \notin Att \longrightarrow \neg[u : (G \to F)] \in Th^{JT_{CS}}(cons(\delta_m))$ and $u : (G \to F) \in \mathcal{WS}^{\Pi}$ for a term $u \in Sub(t)$ s. t. $t : F = cons(\delta_n)$ and, for a process Π of T, $\delta_m \in \Pi$ and $\delta_n \in \Pi$

The following proposition characterizes realizations of warranted Af's:

Proposition 35. *An argument A_n is contained in a complete, grounded, preferred or stable extension of a warranted Dung's framework $Af = (Arg, Att)$ iff a formula $t : F = cons(\delta_n)$ such that $\delta_n \in D$ for a default theory $T = (W, D)$ is contained in the corresponding* $\mathbf{JT_{CS}}$*-complete,* $\mathbf{JT_{CS}}$*-grounded,* $\mathbf{JT_{CS}}$*-preferred or* $\mathbf{JT_{CS}}$*-stable extension $Th^{JT_{CS}}(\Gamma)$ for a realization T of Af.*

Proof. See the Appendix for a proof sketch. □

In Dung's framework arguments are only implicit and one can consider each argument A as a statement of the following type "There is an argument A". When realized in justification logic, each of these existential statements can be instantiated with an explicit argument structure $t : F$.

One may wonder what is the significance of (un)warranted abstract argumentation frameworks for formal argumentation in general. We will conclude this section by pointing out what could the realization results from justification logic contribute to our understanding of arguments. The most important insight given by the justification logic realization of AFs is that once we include reason terms into our formal language, we bring forward the requirements on the logical language that are only implicit in representing arguments as graph nodes. One of these requirements is that the reasoning structure that we call "backing" in this paper has to be built according to the axioms and rules of the underlying calculus of reason terms. According to it, it is impossible to build a "proof term" or a reason term that would support and undercut one and the same conclusion, which is the result obtained in Theorem 33. However, an isolated AF cycle requires a possibility to have a default reason term, without other default reasons as its subterms, that supports and undercuts a conclusion introduced in a single consequent formula of a default rule.

Such loops and cycles that correspond to attack relations in AFs cannot easily be exemplified in natural language either. Even self-defeating argument that require more than a single default inference are difficult to exemplify, as Pollock's pink elephant example witnesses. Starting with the work in [22], it has been argued that attack loops could be exemplified with the statement "I am unreliable" or, using the third-person perspective, "An agent says that the agent is unreliable". In the same vein, the above discussed three-node cycle of attacks has been exemplified [24] by a scenario featuring agents who question one another's reliability in the following way.[29] Suppose that there are three agents, namely Bert, Ernie and Elmo. If Bert says that Ernie is unreliable, then everything that Ernie says cannot be relied on. If Ernie says that Elmo is unreliable, then everything that Elmo says cannot be relied on. Finally, If Elmo says that Bert is unreliable, then everything that Bert says cannot be relied on. This creates a cycle of attacks among Bert, Ernie and Elmo.

It is in such borderline examples of arguments that we can value the precision of the justification logic language. Natural language allows the type of self-referentiality featured in the sentence "I am unreliable". With the use of the justification logic language, we can see that such examples belong to a special group of statements that require the logical machinery of propositional quantification or that of quantification into sentential position [78, §3.5]. In an extended language with propositonal quantifiers, we could represent the statement "I am unreliable" with the following formula

$$t : \forall p(\neg t : p),$$

where p is a propositional variable. The Bert-Ernie-Elmo attack cycle would be just an extended version of the loop example. Take

$$t_1 : \forall p(\neg t_2 : p), \qquad t_2 : \forall q(\neg t_3 : q) \quad \text{and} \quad t_3 : \forall r(\neg t_1 : r)$$

to realize the attack structure of the conflicting testimonies of unreliability among the three agents, where p, q and r are propositional variables. Recall from the Introduction that the basic informal reading of justificiation assertions $t : F$ is that "t is a reason justifying F". This means that the following informal reading applies to the formula $t_1 : \forall p(\neg t_2 : p)$: "Bert's testimony (or statement) is a reason justifying that, for everything that Ernie says, Ernie's testimony (or statement) is not a reason justifying what is being said". Bert's testimony is thus understood as a source of information undermining another source of information, namely, Ernie's testimony.

[29] Similar examples are discussed in [66] and [69].

There are two findings related to the above examples that deserve our attention in the context of modelling arguments. Firstly, if the above examples are to be taken as arguments on a par with arguments that do not require such strong logical machinery, they should not be considered as a part of the *default* reasoning paradigm of argumentation. Such examples of argumentative attack belong to the *plausible* reasoning paradigm. In the default reasoning paradigm, which is the paradigm we investigate in this paper, argumentative attacks result from attacking defeasible inferences, as illustrated by rebutting and undercutting attacks form Example 8. In the plausible reasoning paradigm, argumentative attacks result from adding new information that questions old information and, thereby, it might question old conclusions. Notice that undercutting and rebutting attacks do not question the reliability of old information. For example, concluding that the table is white, rather than red, cannot question the fact that the table looks red under the red lighting. On the other hand, if you question the old information that the table is red looking, then you compromise both old information and any default conclusions that may follow from old information. This type of attack is called "undermining" and it is defined as an attack on premises of an argument [79, p. 626]. In the Bert-Ernie-Elmo attack cycle, the three sources of information are undermined in such a way that the testimonies of the three agents question one another in the proposed order. This differs from the attacks induced by default inferences, where some default step is being questioned, rather than the credibility of information sources.

Secondly, default justification logic shows that argumentation frameworks that include such arguments on a par with other defeasible arguments do not consider the paradoxical nature of the mentioned example. In an important sense, ASPIC+ is still too abstract to capture the intensional paradox created by adding propositional quantification in the justification logic representation of attack cycles.[30] Notice that the reason term t in the statement $t : \forall p(\neg t : p)$ justifies that it cannot justify any proposition. To assess if such reason terms could ever be acceptable, we would first need to resolve what it is that t justifies. Following, for example, Prior's explanation [71] of the intensional version of the Liar paradox, there have to be at least two statements justified by the operator t. The issues that $t : \forall p(\neg t : p)$ raises are fundamental to our understanding of arguments, but they cannot be further developed here. For now, we are able to conclude that AFs, as well as structured argumentation frameworks, are not capable of capturing the paradoxical nature and the exact logical structure of the examples discussed above. Structured argumentation frameworks may be more expressive in the sense that they allow such arguments with their definitions of arguments. However, their expressivity rests on the fact that they do not specify a logical system expressive enough neither to logically represent reasons nor to logically represent arguments. Once we have a precise language with reason terms, we are able to talk about the issues of whether to include paradoxical propositions of the type $t : \forall p(\neg t : p)$ on a par with other arguments or not. More importantly, we are in a position to discuss what is required to logically represent notorious cycles of attacks in formal argumentation.

5. Rationality postulates for structured argumentation

Section 3.1 shows that default rules with justification formulas are expressive enough to model elements of arguments that are traditionally seen as extra-logical, such as warrants and backings. The results from Section 4 establish the logic of default justifications as a system that explicitly features the structure of arguments and uses Dung's methods for argument evaluation. The $\mathbf{JT}_{\mathbf{CS}}$ variants of admissible,

[30] See [70] for a discussion about intensional paradoxes. Intensional paradoxes belong to a "class of paradoxes of self-reference whose members involve intensional notions such as *knowing that*, *saying that*, etc." [70, p. 193].

complete, grounded, preferred and stable extensions preserve reasonable outputs of the corresponding Dung's extensions. An additional question that may be asked is whether our logic also behaves reasonably with respect to "rationality postulates" that are set for structured argumentation frameworks in the literature [1,25].

According to [1], the exact formulation of rationality postulates for structured argumentation frameworks depends on the family of a logical language that they use: rule-based or classical. In frameworks with rule-based languages, a distinction is made between strict rules (rules without exceptions) and defeasible rules (rules that may have exceptions). Arguments are built according to the available strict and defeasible rules. Examples of such systems are ASPIC+ [68] and DeLP [41]. In frameworks with classical languages, arguments are built from a knowledge base using an underlying monotonic logic. Examples of frameworks that use classical languages are [18] and [19]. The framework described in [18] is based on a propositional language, while that of [19] is based on a first-order language.

Following [1], we will first consider five postulates, originally formulated for argumentation frameworks built on classical languages. In general, classical-logic based argumentation frameworks start from the idea that there is some knowledge base with classical logic formulas. We define arguments from that (possibly inconsistent) knowledge base as pairs of sets of formulas and conclusion formulas such that a conclusion formula is classically entailed by a set of formulas. We will here present the five postulates without committing to Amgoud's definition of an argument. We do so deliberately, because the definition of an argument for *classical-logic based* frameworks cannot be applied to our logic. The reasons will be given shortly after we present the postulates. We give their "framework-neutral" formulation, leaving the exact definitions of framework extensions, arguments, sub-arguments, strict rules, premises and conclusions unspecified:

Closure The set of conclusions for each extension is closed under strict rules.

Sub-arguments If an argument is contained in an extension, then all the sub-arguments of the argument are contained in the extension.

Consistency The set of conclusions for each extension is consistent.

Exhaustiveness If each premise and the conclusions of an argument are conclusions of an extension, then the argument is contained in the extension.

Free precedence If an argument is not involved in any conflict, then the argument is contained in each extension.

Although we mentioned that the five postulates provide criteria to evaluate classical-logic based argumentation frameworks, they are also relevant for rule-based argumentation frameworks. In fact, their rule-based framework variants can be found in [2]. To conclude the discussion about the rationality postulates at the end of the section, we briefly turn to the recent debate over satisfying the *non-interference* postulate. This postulate turned be problematic for those approaches that make a distinction between strict and defeasible rules.

5.1. Delimiting the notion of argument in default justification logic

To discuss whether rationality postulates hold for a system, it is required to have a precise definition of an argument. Note that default justification logic offers both a narrower and broader understandings of an argument, which may include implicit components. The narrower understanding takes every formula of the type $t : F$ to be a structured argument such that t represents premises of an argument and F represents its conclusion. However, as Fig. 3 shows, t codifies a more complex structure that involves implicit

features of an argument, such as an argument's warrant and its backing. This offers a broader perspective whereby an argument can rather be seen as an argument schema, which is inclusive of its implicit elements. For the discussion on the rationality postulates Closure, Consistency and Free-precedence, it suffices to focus on explicit features of arguments in the sense of the narrower understanding. To discuss Sub-arguments and Exhaustiveness, we will use additional elements from the broader understanding of arguments.

Although the idea of a classical-logic based system is closer to our default logic, the postulates given by [1] are not directly applicable to our logic. While the logic of default justifications uses $\mathbf{JT_{CS}}$ consequence to build arguments, it also allows for defeasible rules by means of extending the application operation $\cdot$ for default rules. It is not the case that all arguments are built from a knowledge base using only the monotonic consequence for the underlying language, as required in [1, p. 2030]. Default reasons are built using warrants of the type $u : (F \rightarrow G)$ and warrants function as defeasible rules, but they are not initially known and they do not need to become a part of the knowledge base, although they potentially could. Finally, and most importantly, arguments in the narrower sense are featured in the $\mathbf{JT_{CS}}$ language itself, which means that a pair (*premises*, *conclusion*) is also an object-level formula, unlike, for example, in [18].

On the other hand, rule-based languages introduce the differentiation between strict and defeasible rules, but these rules are not a part of the base language. In contrast with, for example, [68], arguments in justification logic are built via the operations in the $\mathbf{JT_{CS}}$ language, where the strict rules are simply the rules of the logic $\mathbf{JT_{CS}}$ and defeasible rules are in the object language due to the fact that both warrants are a part of the $\mathbf{JT_{CS}}$ and the operation $\cdot$ is a part of the language. This makes any argument logically dependent on other strict and defeasible conclusions within the system. For example, since warrants are formulas of the $\mathbf{JT_{CS}}$ language, a consequent of a default rule may refer to the underlying warrant of another default rule in the way of an undercut attack.

Our logic takes the middle way between rule-based systems and classical-logic based systems by combining the distinction between strict rules and defeasible rules with logical dependency of arguments via $\mathbf{JT_{CS}}$ consequence. This middle way is epitomized by the two roles that warrants have in the system: they function as both implicit rules as well as statements.[31] Warrants in the role of rules enable default conclusions and warrants in the role of logical statements enable other formulas to refer to warrants within the logical system. However, this also means that our logic cannot be aligned with only one of the two families of logic-based argumentation systems identified in [1].

5.2. *Postulates for default justification logic*

Even without directly applying the postulates for classical-logic based argumentation, we can check whether the desiderata on which [1] builds the rationality postulates hold for our logic. We first examine three postulates from [1, pp. 2032–2035] that are easily adaptable for our logic. For any $\mathbf{JT_{CS}}$-complete, $\mathbf{JT_{CS}}$-grounded, $\mathbf{JT_{CS}}$-preferred or $\mathbf{JT_{CS}}$-stable extension Γ of a default theory $T = (W, D)$, the following postulates are required to hold:

$\mathbf{JT_{CS}}$ closure The set of conclusions for each $\mathbf{JT_{CS}}$ extension Γ is closed under strict rules;

[31] Note that this corresponds to Toulmin's ambiguous use of the term "warrant". For example, [77, p. 91] refers to warrants as both rules and statements in a single paragraph.

$\mathbf{JT_{CS}}$ consistency The set of conclusions for each $\mathbf{JT_{CS}}$ extension Γ is $\mathbf{JT_{CS}}$-consistent;[32]

$\mathbf{JT_{CS}}$ free precedence If some argument $t : F$ is not involved in any conflict, then $t : F \in \Gamma$ for each $\mathbf{JT_{CS}}$ extension Γ.

In our logic, strict rules are simply the rules of $\mathbf{JT_{CS}}$ logic. By Definitions 20, 23, 27, 28, extensions are closed under $\mathbf{JT_{CS}}$ consequence and, therefore, closed under strict rules.

The satisfaction of the consistency postulate is guaranteed for each default theory $T = (W, D)$ with a consistent set of facts W. For such default theories, it can be easily shown that $\mathbf{JT_{CS}}$ consistency of each extension is preserved by the conditions of application for each default rule. This follows from Theorem 17 and the fact that each $\mathbf{JT_{CS}}$ extension is conflict-free. Exceptions to the consistency postulate are theories with an inconsistent set of facts W. This reflects the way in which our logic deals with inconsistent information. Firstly, an agent starts with known facts represented by justified formulas that do not conflict with one another. Conflicts arise only after an agent needs to extend an incomplete knowledge base by default assumptions. Resolving such meaningful conflicts always leads to $\mathbf{JT_{CS}}$-consistent extensions.

The free precedence postulate requires that the system infers all the arguments and, in general, formulas that do not conflict with any other argument. As stated above, we take arguments in the narrower sense of formulas $t : F$ and these arguments may be based either on strict or on defeasible rules. This postulate follows trivially for all $\mathbf{JT_{CS}}$ extensions, except for $\mathbf{JT_{CS}}$-admissible extensions that do not maximize inclusion of arguments by their definition. Notice that for other $\mathbf{JT_{CS}}$ extensions, no formula $t : F = cons(\delta)$ is excluded from a $\mathbf{JT_{CS}}$ extension Γ, unless δ is inapplicable to the respective process containing Γ or one of the subterms of t is undercut by Γ. The inclusion of all free formulas and arguments built on conflict-free grounds is then ensured by the closure under $\mathbf{JT_{CS}}$ consequence.

For the two additional postulates from [1], the notion of a sub-argument of an argument needs to be defined. We will start again from the narrower understanding of an argument in the sense of any formula $t : F$. The concept of a *sub-argument* for default application will be taken to mean the following:

- If a formula $(u \cdot t) : G$ is obtained by means of application (axiom A1) or default application from the formulas $t : F$ and $u : (F \to G)$, then $t : F$ and $u : (F \to G)$ are *sub-arguments* of $(u \cdot t) : G$;
- If a formula $(t + u) : F$ is obtained by means of sum (axiom A2) from either the formula $t : F$ or the formula $u : F$, then at least one of the formulas $t : F$ and $u : F$ is a *sub-argument* of $(t + u) : F$.

If an argument $t : F$ is a sub-argument of $(u \cdot t) : G$ and a sub-argument of $(t + u) : F$, then any sub-argument of $t : F$ is also a sub-argument of $(u \cdot t) : G$ and $(t + u) : F$. Notice that if an argument $t : F$ is a sub-argument of $(t + u) : F$, it is not necessary that there is some formula $u : G$ which is also a sub-argument of $(t + u) : F$. It is possible that some justification term u does not justify any formula G. For a simple example, take some justification constant c and any formula F. Then $(c \cdot c)$ is not a justification for F in $\mathbf{JT_{CS}}$ logic because the application operation that gives $(c \cdot c)$ is not meaningful for an injective constant specification $\mathcal{CS}$.

The following two postulates require rational acceptance of an argument with respect to its substructure:

$\mathbf{JT_{CS}}$ sub-arguments If an argument $t : F$ is in a $\mathbf{JT_{CS}}$ extension Γ, then any sub-argument of $t : F$ is also in Γ;

[32] It is not needed to distinguish between the *direct* and *indirect* version of this postulate. This distinction rests on the assumption that, by their definition, extensions of an argumentation framework are not closed under strict rules. However, in our logic, this follows from the definitions of $\mathbf{JT_{CS}}$ extensions.

$\mathbf{JT}_{\mathbf{CS}}$ exhaustiveness If each sub-argument and the formula F for some argument $t : F$ are conclusions of $\mathbf{JT}_{\mathbf{CS}}$ extension Γ, then $t : F$ is in Γ.

In contrast to the Exhaustiveness postulate on page 35, notice that the $\mathbf{JT}_{\mathbf{CS}}$ variant of exhaustiveness does not mention premises of an argument as conclusions of $\mathbf{JT}_{\mathbf{CS}}$ extensions. On the narrower understanding of the arguments as justification assertions, the premises of an argument are reason terms, but not well-formed formulas. The postulate is reinterpreted to track conclusion formulas that are sub-arguments for an argument, because reason terms codify reasoning steps from those formulas such as, for example, the warrants of arguments. The $\mathbf{JT}_{\mathbf{CS}}$ exhaustiveness postulate obviously holds for all $\mathbf{JT}_{\mathbf{CS}}$ extensions closed under $\mathbf{JT}_{\mathbf{CS}}$ consequence by axioms A1 and A2. Thus, informally, if the steps of an argument are contained in an extension, then the argument itself is.

The sub-arguments postulate can be seen as a dual version of exhaustiveness, in the sense that it requires that all the steps of an accepted argument should also be accepted [1, p. 2029]. This postulate is not directly satisfied by our logic. Take, for example, an argument $(u \cdot t) : G$ obtained by default application. According to default application, one of the sub-arguments of $(u \cdot t) : G$ is some formula $u : (F \to G)$ which is neither a part of a knowledge base W for a default theory T nor is it required for that formula to become a part of an extended knowledge base, which results from applying the available defaults.

Does that mean that arguments introduced by default rules are based on unjustified reasoning steps? We can show that this is not the case. Although the sub-arguments postulate is not directly satisfied, the basic idea behind the postulate is: "an argument cannot be accepted if at least one of its sub-parts is bad" [1, p. 2033]. This desideratum holds because, even if the sub-argument $u : (F \to G)$ of an argument $(u \cdot t) : G$ does not become a part of a knowledge base, the system ensures that the warrant $u : (F \to G)$ has not been compromised by other available arguments in the knowledge base. For any argument $(u \cdot t) : G$ and its warrant $u : (F \to G)$, if $(u \cdot t) : G$ is in a $\mathbf{JT}_{\mathbf{CS}}$ extension, then that extension contains the formula $((c \cdot t) \cdot (u \cdot t)) : (F \to G)$, assuming that the constant c justifies the axiom $F \to (G \to (F \to G))$ and that the sub-argument $t : F$ of $(u \cdot t) : G$ is also contained in the extension. Therefore, it is possible to ascertain that none of the steps in building the argument $(u \cdot t) : G$ has turned out to be bad, if the argument $(u \cdot t) : G$ is actually accepted in a $\mathbf{JT}_{\mathbf{CS}}$ extension.

5.3. The non-interference postulate

Recently, a number of authors including [23,44,82] have discussed the way in which the *Ex Falso Sequitur Quodlibet* principle in the underlying logic of an argumentation system may threaten plausibility of acceptability semantics outputs for the system. We explain the intuition behind the problem of "trivialisation" [44, p. 199] that the *Ex Falso* principle causes in the case of rebutting attacks, but a reader can refer to the mentioned sources for a more technical elaboration of the problem. In a nutshell, argumentation systems with strict and defeasible rules allow that arguments using defeasible rules have contradictory conclusions, say φ and $\neg\varphi$, rebutting each other. However, if strict rules are based on a deductive logic with the *Ex Falso* principle, then the conclusions φ and $\neg\varphi$ of the arguments can form an additional argument for any proposition ψ, where ψ is a conclusion of some argument in the system. This produces undesirable effects, because the conclusion ψ may be a conclusion of an argument that is unrelated to the rebuttal between φ and $\neg\varphi$.

The effects of trivialisation are recognized, for example, in the context of the ASPIC+ framework. To test whether a system is susceptible to the problem, Caminada, Carnielli and Dunne [26] propose the rationality postulate called "non-interference" for generalized defeasible theories based on propositional

logic. We take DT to be defined as a pair (P, R), where P is a (consistent) set of propositional formulas and R a set of defeasible rules. Two theories DT_1 and DT_2 are said to be syntactically disjoint iff $Atoms(DT_1) \cap Atoms(DT_2) = \emptyset$, where $Atoms(DT)$ is the set of all atomic propositional formulas occurring in DT. The postulate requires the following:

Non-interference For two syntactically disjoint defeasible theories DT_1 and DT_2 such that P_1 and P_2 are consistent, conclusions that are acceptable under some argumentation semantics for each of the defeasible theories DT_1 and DT_2 should remain acceptable under the same argumentation semantics for the merged defeasible theory defined as the union $DT_1 \cup DT_2 = (P_1 \cup P_2, R_1 \cup R_2)$.

Violating this postulate would imply "that a defeasible theory somehow influences the entailment of a completely unrelated (syntactically disjoint) defeasible theory when being merged to it" [23, p. 2728].

In default justification logic, the basic $\mathbf{JT_{CS}}$ logic language extends that of propositional logic with justification assertions. In the $\mathbf{JT_{CS}}$ logic, we assume the disputed *Ex Falso Sequitur Quodlibet* principle in axiom $A0$. However, in our operational semantics, rebutting reasons are necessarily kept apart by the way in which we build process trees. According to Definition 9, default rules are applied to an evidence base in such a way that it it is not possible that $\mathbf{JT_{CS}}$-inconsistent conclusions of an argument together form new arguments based on the *Ex Falso* principle.

Let $Atoms(T)$ be the set of all atomic propositional formulas occurring in a default theory $T = (W, D)$ and let $Subterms(T)$ be the set of all subterms occurring in a default theory theory $T = (W, D)$. We say that the default theories T_1 and T_2 are syntactically disjoint iff $Atoms(T_1) \cap Atoms(T_2) = \emptyset$ and $Subterms(T_1) \cap Subterms(T_2) = \emptyset$. We can then ask whether our logic satisfies the following $\mathbf{JT_{CS}}$ variant of the non-interference postulate or not:

$\mathbf{JT_{CS}}$ non-interference For two syntactically disjoint default theories T_1 and T_2 such that W_1 and W_2 are $\mathbf{JT_{CS}}$-consistent, conclusions that are acceptable under some argumentation semantics for each of the defeasible theories T_1 and T_2 should remain acceptable under the same argumentation semantics for the merged defeasible theory defined as the union $T_1 \cup T_2 = (W_1 \cup W_2, D_1 \cup D_2)$.

It is clear that our default theories do not satisfy the postulate with respect to the $\mathbf{JT_{CS}}$-stable semantics. For example, consider joining some syntactically disjoint default theories T_1 and T_2, where T_1 is the theory defined in Example 26. Recall that the example represents a self-defeating argument featuring Robert who suffers from pink-elephant phobia. Regardless of whether the default theory T_2 has a $\mathbf{JT_{CS}}$-stable extension or not, the union $T_1 \cup T_2$ will not have a $\mathbf{JT_{CS}}$-stable extension, given the assumption that the two theories are syntactically disjoint.

Assuming consistent and syntactically disjoint default theories, we can check that our default justification logic satisfies the $\mathbf{JT_{CS}}$ non-interference postulate for all the semantics based on $\mathbf{JT_{CS}}$ admissibility. The proof for this statement would follow from the above observation that our process tree semantics prevents constructing new arguments from conclusions of rebutting arguments, and the assumption that we consider consistent default theories.

To conclude the discussion about rationality postulates, we will indicate several limitations on giving a systemic evaluation of default justification logic in this paper. Firstly, default justification logic as presented in this paper does not yet model all the varieties of argumentative attacks as, for example, enabled in ASPIC+.[33] In particular, this paper does not include a justification logic variant of *undermining*

[33]There are also further differences in the basic languages underlying our logic of arguments and the examples that initiated the non-interference postulate debate. Consider, for example, the statement "John says the cup of coffee contains sugar". In

attacks or attacks on premises of an argument. Undermining in justification logic is first introduced in [62] and it is also further developed as a part of an ongoing work. It can be plausibly assumed that satisfying non-interference will not be affected by undermining, because we use the mechanism of multiple extensions of an evidence base to model undermining.

6. Related work and discussion

The logic of default justification has a similar connection to abstract argumentation frameworks as standard justification logic systems have to their modal logic counterparts. Artemov [6] provided a proof of the *Realization Theorem* that connects the logic of arithmetic proofs LP with the modal logic S4. The result has been followed up by similar theorems for many other modal logics with known "explicit" justification counterparts.[34] In our paper we show that our logic can be considered as an explicit justification logic counterpart to a substantial subclass of abstract argumentation frameworks called warranted frameworks.

In the general context of default logics, our logic introduces some new technical properties for normal default theories that are still to be thoroughly investigated. Among them are revision of extensions and interaction of different defaults that does not rely on their preference orderings, as commonly done in default logic [29]. An extensive account of default reasons that makes use of preference orderings on defaults is developed by [47]. Horty's logic is based on a propositional language and develops from a different notion of reasons, which are not explicitly featured in the language itself. He uses the idea of preferences to represent undercutters or exclusionary reasons.

Our work provides a complementary addition to the study of less-than-ideal reasons in justification logic. Among related approaches, the logic of conditional probabilities developed by [59] introduces a way to model non-monotonic reasoning with justification assertions. Their proposal is based on defining operators for approximate probabilities of a justified formula given some condition formula. Using conditional probabilities, the logic models certain aspects of defeasible inferences with justification terms. Yet the system can neither encode the defeasibility of justification terms in their internal structure nor model defeat among reasons, to mention only some differences from our initial desiderata.

Baltag, Renne and Smets [12] define a justification logic in which an agent may hold a justified belief that can be compromised in the face of newly received information. The logic builds on the ideas from belief revision and dynamic epistemic logic to model examples where epistemic actions cause changes to an agent's evidence. Concerning the possibility of modelling defeaters, the logic offers two dynamic operations that change the availability of evidence in a model, namely "updates" and "upgrades" [12, p. 183]. Evidence obtained by updates counts as "hard" or infallible, while upgrades bring about "soft" or fallible evidence. With the use of these actions, epistemic models can represent justified beliefs being defeated, for example, by means of an epistemic action of update with hard evidence. In this way,

standard structured argumentation frameworks, the underlying language is usually assumed to be a classical logic language. If we take propositional logic, the statement is formalized using a propositional formula, e.g., 's'. Justification assertions are richer expressions and this opens up a possibility to go beyond the propositional language and indicate that a certain reason supports a formula. As we do in the examples in this paper, the statement would translate into the justification assertion '$j : S$', thus specifying the source of information as a reason justifying the statement. This is the starting point of the modelling testimonies and undermining attacks in [62] that can be used to provide an alternative formalization of the benchmark examples from [23]. We already indicated at the end of Section 4 how this affects the way in which we can model self-referential claims of unreliability such as "John says that John is unreliable".

[34] See [38] for a good overview of realization theorems.

however, the mechanism by which reasons may conflict with one another is simply being "outsourced" to an extra-logical notion of fallibility and, therefore, the logic does not directly address the ways of defeat that we formalize in this paper.

Several interesting paths could be followed in connecting the logic of default justifications with formal argumentation frameworks. Among frameworks with abstract arguments, the AFRA framework [15] with recursive attacks offers a possibility of representing attacks to attacks. This conceptual advance is useful in connecting default reasons to abstract arguments. More obviously, our logic is closely related to the frameworks with structured arguments, which is why connections with systems such as ASPIC+ [68], DeLP [41], SG [46] and the logic-based argumentation framework by [18] are interesting to explore. Since each of these frameworks elaborates on the notion of defeat, a thorough comparison to our logic would shed light on their formal connections. A different logic-based perspective on argumentation frameworks is given by [27] and [45]. Both papers start from the idea of studying attack graphs and formalizing notions of extensions from abstract argumentation theory using modal logic, with the former approach being proof-theoretical and the latter model-theoretical. A further interesting research venue in the field of argumentation theory is the one about the logical interpretation of *prima facie* justified assumptions in [80]. The DefLog system which is developed there is closely related to ours in motivation, but it develops from a perspective of a sentence-based theory of defeasible reasoning instead of a rule-based or argument-based approach.

Further developments are possible starting from the basic form of default rules with justified formulas. We indicate some of the possibilities to extend the basic logic. On the technical side of the logic, we used only the expressiveness of normal default rules and we still need to investigate how to add non-normal default rules. Since all processes are successful for normal default theories, it is interesting to see whether the logic has some further desirable properties such as, for example, goal-driven query evaluation.

It is also possible to use the first-order variant of justification logic [37], instead of the propositional justification logic used here. This is an intriguing direction because of the possibilities it opens. To mention one of them, a first-order warrant of a default rule would fully correspond to the Toulminian warrant, being also generally applicable to all the objects as a rule schema. Defining default rules on such rich language would be one step closer to a full account of structured arguments.

One of the ongoing projects started in [62] is to add the dynamic aspect of default theories with justification terms. Besides the existing mechanism of extension revisions, we also consider changes to a default theory and adding belief-revision-style operations to deal with such theory changes. A similar proposal is given in [4] for standard default theories. This completes the logic proposed here because it enables modelling an additional kind of defeat that was only briefly mentioned in this paper, namely *undermining* defeat. This form of defeat is understood as an attack on the premises or assumptions of an argument [79, p. 626] and premises can be interpreted as the information contained in the set W for a theory $T = (W, D)$.

At this point, our logic is presented as single-agent. Since argumentation is distinctively dialogical and multi-agent practice, developing a multi-agent generalization of the default justification logic stands as one of the main future goals. The problem that needs to be initially addressed is how does inclusion of multiple agents essentially differ from the already existing argument exchange through default reasons.

Finally, the logic of default justifications has a potential to link the formal analysis of knowledge with mainstream epistemology. Ever since the concept of justification entered into epistemic logics, there has been a tendency to model mainstream epistemology examples, proposed by e.g. Russell, Dretske and Gettier, with the use of justification logic [7,8]. With the introduction of default justifications, however, we gain flexibility for a more full-blooded integration of the formal theory of justification with the

study of knowledge in philosophy, since paradigmatic examples include both incomplete specification of reasons and defeated reasons. Potential benefits of a non-monotonic system of justifications in this context were anticipated by Artemov in [7, p. 482] where he states that "to develop a theory of non-monotonic justifications which prompt belief revision" stands as an "intriguing challenge".

Acknowledgements

I am grateful to Allard Tamminga, Barteld Kooi and Rineke Verbrugge for their generous advice and valuable comments on the previous versions of this manuscript. My research is supported by *Ammodo KNAW* project *Rational Dynamics and Reasoning* awarded to Barteld Kooi. I am also grateful to the three reviewers of the Argument & Computation journal for their constructive suggestions that helped me to improve this paper.

Appendix

Proof of Theorem 6. The claim from left to right is obvious. For the other direction, take $\mathcal{CS}$ to be some specific axiomatically appropriate and injective constant specification. We first show that if a set Γ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable, then for all formulas $F \in Fm$, it holds that $\Gamma \cup \{F\}$ or $\Gamma \cup \{\neg F\}$ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable. Suppose that Γ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable and that $\Gamma \cup \{F\}$ and $\Gamma \cup \{\neg F\}$ are both not $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable. Then there would be finite subsets Γ' and Γ'' of Γ such that $\Gamma' \cup \{F\}$ and $\Gamma'' \cup \{\neg F\}$ are not $\mathbf{JT}_{\mathbf{CS}}$ satisfiable. Since for no interpretation $\mathcal{I}$ it holds that $\mathcal{I} \models \{F, \neg F\}$, $\Gamma' \cup \{F, \neg F\}$ is never $\mathbf{JT}_{\mathbf{CS}}$ satisfiable. But since for any possible interpretation $\mathcal{I}$ one of the formulas F or $\neg F$ holds, this means that $\mathcal{I} \models \Gamma' \subseteq \mathcal{A}'$ for a class of interpretations $\mathcal{A}'$ such that for each $\mathcal{I}' \in \mathcal{A}'$, it holds that $\mathcal{I}' \models \neg F$. In a similar way we get that $\mathcal{I} \models \Gamma'' \subseteq \mathcal{A}''$ for a class $\mathcal{A}''$ consisting of the interpretations $\mathcal{I}''$ such that $\mathcal{I}'' \models F$. Therefore, we have that $\mathcal{I} \models \Gamma' \cap \mathcal{I} \models \Gamma'' = \emptyset$ and, thus, $\Gamma' \cup \Gamma''$ is not $\mathbf{JT}_{\mathbf{CS}}$-satisfiable. But $\Gamma' \cup \Gamma''$ is a finite subset of Γ and this contradicts the assumption that Γ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable.

The next step is proving a $\mathbf{JT}_{\mathbf{CS}}$ variant of the Lindenbaum lemma. Using the above-proven statement that for any $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable set of formulas Γ and any formula F, $\Gamma \cup \{F\}$ or $\Gamma \cup \{\neg F\}$ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable together with the fact that $\Gamma \cup \{F, \neg F\}$ is never $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable, we can construct maximally $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable sets. Let $F_1, F_2, F_3, \ldots$ be an enumeration of $F \in Fm$. For a $\mathbf{JT}_{CS}$-finitely satisfiable set Γ and for all $i \in \mathbb{N}$ define an increasing sequence of sets of formulas as follows:

$\Gamma_0 = \Gamma$
$\Gamma_{i+1} = \Gamma_i \cup \{F_i\}$ if $\Gamma_i \cup \{F_1\}$ is $\mathbf{JT}_{CS}$-finitely satisfiable, otherwise $\Gamma_{i+1} = \Gamma_i \cup \{\neg F_i\}$
$\Gamma' = \bigcup_{i=0}^{\infty} \Gamma_i$

We can prove that Γ' is $\mathbf{JT}_{CS}$-finitely satisfiable by induction. The base case $\Gamma_0 = \Gamma$ holds by assumption. Then we claim that for all $i \in \mathbb{N}$, Γ_i is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable. For some $n \in \mathbb{N}$, take Γ_n to be $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable. Then either $\Gamma \cup \{F_n\}$ or $\Gamma \cup \{\neg F_n\}$ is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable and, therefore, Γ_{n+1} is also $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable.

From the construction of the increasing sequence, we have that for any finite set $\Gamma_k \subseteq \Gamma'$ there is a $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable finite set $\Gamma_{k+1} \subseteq \Gamma'$ such that $\Gamma_k \subseteq \Gamma_{k+1}$ and, therefore, Γ_k is $\mathbf{JT}_{\mathbf{CS}}$-satisfiable. Since any finite subset of Γ' is $\mathbf{JT}_{\mathbf{CS}}$ satisfiable, Γ' is $\mathbf{JT}_{\mathbf{CS}}$-finitely satisfiable. The set Γ' is maximal

according to the enumeration of the set of formulas Fm and contains exactly one of F_i or $\neg F_i$ for all $i \in \mathbb{N}$.

Now we define a valuation v such that $v(P) = \mathit{True}$ iff $P \in \Gamma'$ and the reason assignment $*(t) = \{F \mid t : F \in \Gamma'\}$. We only need to check the conditions on the reason assignment function. First, we show that $*(\cdot)$ satisfies the application condition. Since the formula $t : (F \to G) \to (u : F \to (t \cdot u) : G)$ is $\mathbf{JT_{CS}}$ valid, it is contained in Γ'. If $F \to G \in *(t)$ and $F \in *(u)$, then $\{t : (F \to G), u : F\} \in \Gamma'$. Since Γ is closed under *Modus ponens*, we have that $(t \cdot u) : G \in \Gamma'$ and, therefore, $G \in *(t \cdot u)$. Similarly, since the formulas $t : F \to (t + u) : F$ and $u : F \to (t + u) : F$ are both in Γ' we can easily check that the sum condition holds for $*(\cdot)$.

Finally, we have defined an interpretation $\mathcal{I} = (*, v)$ that meets $\mathcal{CS}$ and we need to prove that truth in this interpretation is equivalent to inclusion in Γ':

$$\mathcal{I} \models F \text{ iff } F \in \Gamma'$$

The proof is by induction on the structure of F. For the base case, suppose F is an atomic formula P: $\mathcal{I} \models P$ iff $v(P) = \mathit{True}$ iff $P \in \Gamma'$.

For the inductive step, suppose that if the result holds for F and G, then it also holds for $\neg F$, $F \wedge G$, $F \vee G$, $F \to G$ and $t : F$. For the negation case: $\mathcal{I} \models \neg F$ iff $\mathcal{I} \not\models F$. By the inductive hypothesis, $\mathcal{I} \not\models F$ iff $F \notin \Gamma'$. By the maximality of Γ', we have that $F \notin \Gamma'$ iff $\neg F \in \Gamma'$.

For the conjunction case: $\mathcal{I} \models F \wedge G$ iff $\mathcal{I} \models F$ and $\mathcal{I} \models G$. By the inductive hypothesis, $\mathcal{I} \models F$ and $\mathcal{I} \models G$ iff $F \in \Gamma'$ and $G \in \Gamma'$ iff $F \wedge G \in \Gamma'$. Since other connectives are definable in terms of $\neg$ and $\wedge$, we skip the remaining cases.

Finally for the justified formula case: $\mathcal{I} \models t : F$ iff $F \in *(t)$. By the definition of $*(\cdot)$, it holds that $F \in *(t)$ iff $t : F \in \Gamma'$.

Therefore, for any $\mathbf{JT_{CS}}$-finitely satisfiable set Γ there is an interpretation $\mathcal{I}$ based on a maximal $\mathbf{JT_{CS}}$-finitely satisfiable extension Γ' of Γ such that $\mathcal{I} \models \Gamma$. □

Proof of Proposition 32. The proof is by induction on the acceptance conditions for a formula $t : F = \mathit{cons}(\delta)$ given by the definitions of $\mathbf{JT_{CS}}$-complete, $\mathbf{JT_{CS}}$-grounded, $\mathbf{JT_{CS}}$-preferred and $\mathbf{JT_{CS}}$-stable extension definitions for default theories.

The following is an argument for the $\mathbf{JT_{CS}}$-preferred extension case. Take as the induction base default theory $T = (W, D)$ such that T has a non-empty process $\Pi = (\delta)$ and $t : F = \mathit{cons}(\delta)$. The theory has a $\mathbf{JT_{CS}}$-preferred extension $Th^{JT_{CS}}(\Gamma)$, where $\Gamma = W \cup \{\mathit{cons}(\delta)\}$. The forgetful projection of T is defined as $Af = (Arg, Att)$, where $Arg = \{A\}$ and Att is empty. The only preferred extension of Af is A.

For the inductive step, assume that if $t : F = \mathit{cons}(\delta_k)$ is in a $\mathbf{JT_{CS}}$-preferred extension of a default theory $T_n = (W_n, D_n)$ such that $t : F$ occurs in a closed process $\Pi = (\delta_1, \ldots, \delta_m)$ of T, then it is also in a preferred extension of its forgetful projection $Af_n = (Arg_n, Att_n)$. By the induction hypothesis and the definition of $\mathbf{JT_{CS}}$-preferred extensions, it holds that $t : F \in \Gamma$ such that Γ is a $\mathbf{JT_{CS}}$-admissible extension and for no other $\mathbf{JT_{CS}}$-admissible extension Γ' it holds that $\Gamma \subset \Gamma'$. By the definition of a $\mathbf{JT_{CS}}$-admissible set, it holds that Γ is conflict-free and each formula in Γ is acceptable w.r.t. Γ in Π. For a non-complex default theory and the formula $t : F$, this means that for any undercutting reason $u : G \in Th^{JT_{CS}}(W \cup \{\mathit{cons}(\delta_j)\})$ for t being a reason for F in Π, Γ undercuts u being a reason for G. The forgetful projection maps all the formulas $\mathit{cons}(\Pi')$ for any process Π' into arguments $A_1, \ldots, A_n$ of the framework Af_n and for the undercutter $u : G$ ascribes an attack relation (A_j, A_k), and analogously for any other possible undercutter. Moreover, any conflict-free set is also $\mathbf{JT_{CS}}$ consistent and for each formula $v : H = \mathit{cons}(\delta_p)$ such that $v : H \in \Gamma'$ for a $\mathbf{JT_{CS}}$-admissible extension Γ' of T and $v : H \notin \mathit{cons}(\Pi)$,

it holds that $v : H$ and $t : F$ do not occur together in any process Π' because T is non-complex. According to the forgetful projection, (A_k, A_p) and (A_p, A_k) are both in Att_n. It is easy to check that, by the definition of Dung's preferred extension, the forgetful projection maps Γ into a preferred extension S of Af_n such that A_k is in S. □

Proof of Proposition 35. The proof is by induction on the acceptance conditions for an argument A given by the definitions of complete, grounded, preferred and stable extension definitions for Dung's abstract argumentation frameworks restricted to the subclass of warranted frameworks.

The proof of the inductive step relies on the fact that the realization procedure $\longrightarrow$ preserves the direction of attacks specified by Dung's attack relation. The direction of argument attacks in our operational semantics is defined exactly as semantics in abstract argumentation, where realized extensions can be instantiated with the corresponding consistent models from $\mathbf{JT}_{\mathbf{CS}}$, modulo specifying the logical structure of attacks and closing the realized extensions under the $\mathbf{JT}_{\mathbf{CS}}$ consequence relation.

The realization procedure is straightforward for Af's that amount to directed acyclic graphs as well for (most) Af's whose cycles include two-node-cycle components, which translate into rebuttal between formulas. For example, for the existence of a directed path, the realization assigns an undercutter formula $u : \neg[t : (F \rightarrow G)]$ as an argument that realizes the starting node such that any warrant of a subsequent node is a subformula of G. With the presence of other types of cycles, the realization forces the existence of sub-arguments for at least one argument $t : F$ corresponding to a node from a realized cycle. This follows from Theorem 33. □

References

[1] L. Amgoud, Postulates for logic-based argumentation systems, *International Journal of Approximate Reasoning* **55**(9) (2014), 2028–2048. doi:10.1016/j.ijar.2013.10.004.

[2] L. Amgoud and P. Besnard, A formal characterization of the outcomes of rule-based argumentation systems, in: *International Conference on Scalable Uncertainty Management, SUM 2013*, W. Liu, V.S. Subrahmanian and J. Wijsen, eds, LNCS, Vol. 8078, Springer, 2013, pp. 78–91.

[3] G. Antoniou, *Nonmonotonic Reasoning*, MIT Press, Cambridge, MA, 1997.

[4] G. Antoniou, On the dynamics of default reasoning, *International Journal of Intelligent Systems* **17**(12) (2002), 1143–1155. doi:10.1002/int.10065.

[5] S.N. Artemov, Operational modal logic, Technical Report, MSI 95–29, Mathematical Sciences Institute, Cornell University, 1995.

[6] S.N. Artemov, Explicit provability and constructive semantics, *Bulletin of Symbolic Logic* (2001), 1–36.

[7] S.N. Artemov, The logic of justification, *The Review of Symbolic Logic* **1**(4) (2008), 477–513. doi:10.1017/S1755020308090060.

[8] S.N. Artemov, Justification awareness models, in: *International Symposium on Logical Foundations of Computer Science*, S.N. Artemov and A. Nerode, eds, LNCS, Vol. 10703, Springer, 2018, pp. 22–36. doi:10.1007/978-3-319-72056-2_2.

[9] S.N. Artemov and M. Fitting, Justification logic, in: *The Stanford Encyclopedia of Philosophy*, E.N. Zalta, ed., Metaphysics Research Lab, Stanford University, 2016.

[10] S.N. Artemov and M. Fitting, *Justification Logic: Reasoning with Reasons*, Cambridge Tracts in Mathematics, Vol. 216, Cambridge University Press, 2019.

[11] S.N. Artemov and E. Nogina, Introducing justification into epistemic logic, *Journal of Logic and Computation* **15**(6) (2005), 1059–1073. doi:10.1093/logcom/exi053.

[12] A. Baltag, B. Renne and S. Smets, The logic of justified belief change, soft evidence and defeasible knowledge, in: *International Workshop on Logic, Language, Information, and Computation*, L. Ong and R. de Queiroz, eds, Springer, 2012, pp. 168–190. doi:10.1007/978-3-642-32621-9_13.

[13] A. Baltag, B. Renne and S. Smets, The logic of justified belief, explicit knowledge, and conclusive evidence, *Annals of Pure and Applied Logic* **165**(1) (2014), 49–81. doi:10.1016/j.apal.2013.07.005.

[14] H. Barendregt, W. Dekkers and R. Statman, *Lambda Calculus with Types*, Cambridge University Press, 2013.

[15] P. Baroni, F. Cerutti, M. Giacomin and G. Guida, AFRA: Argumentation framework with recursive attacks, *International Journal of Approximate Reasoning* **52**(1) (2011), 19–37. doi:10.1016/j.ijar.2010.05.004.
[16] P. Baroni and M. Giacomin, Solving semantic problems with odd-length cycles in argumentation, in: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, T. Dyhre Nielsen and N. Lianwen Zhang, eds, LNCS, Vol. 2711, Springer-Verlag, 2003, pp. 440–451.
[17] T.J. Bench-Capon and P.E. Dunne, Argumentation in artificial intelligence, *Artificial Intelligence* **171**(10–15) (2007), 619–641. doi:10.1016/j.artint.2007.05.001.
[18] P. Besnard and A. Hunter, A logic-based theory of deductive arguments, *Artificial Intelligence* **128**(1–2) (2001), 203–235. doi:10.1016/S0004-3702(01)00071-6.
[19] P. Besnard and A. Hunter, Practical first-order argumentation, in: *Proceedings of the National Conference on Artificial Intelligence, AAAI'05*, Vol. 20, AAAI Press, 2005, p. 590.
[20] P. Besnard and A. Hunter, Constructing argument graphs with deductive arguments: A tutorial, *Argument & Computation* **5**(1) (2014), 5–30. doi:10.1080/19462166.2013.869765.
[21] V. Brezhnev, On the logic of proofs, in: *Proceedings of the Sixth ESSLLI Student Session, Helsinki*, K. Striegnitz, ed., 2001, pp. 35–46.
[22] M.W.A. Caminada, Contamination in formal argumentation systems, in: *Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence, BNAIC 2005*, K. Verbeeck, K. Tuyls, A. Nowé, B. Manderick and B. Kuijpers, eds, Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, 2005.
[23] M.W.A. Caminada, Rationality postulates: Applying argumentation theory for non-monotonic reasoning, *Journal of Applied Logics* **4**(8) (2017), 2707–2734.
[24] M.W.A. Caminada, A gentle introduction to argumentation semantics (Summer 2008), Lecture material.
[25] M.W.A. Caminada and L. Amgoud, On the evaluation of argumentation formalisms, *Artificial Intelligence* **171**(5–6) (2007), 286–310. doi:10.1016/j.artint.2007.02.003.
[26] M.W.A. Caminada, W.A. Carnielli and P.E. Dunne, Semi-stable semantics, *Journal of Logic and Computation* **22**(5) (2012), 1207–1254. doi:10.1093/logcom/exr033.
[27] M.W.A. Caminada and D.M. Gabbay, A logical account of formal argumentation, *Studia Logica* **93**(2–3) (2009), 109. doi:10.1007/s11225-009-9218-x.
[28] R.M. Chisholm, *Theory of Knowledge*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[29] J.P. Delgrande and T. Schaub, Expressing preferences in default logic, *Artificial Intelligence* **123**(1–2) (2000), 41–87. doi:10.1016/S0004-3702(00)00049-7.
[30] 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) (1995), 321–357. doi:10.1016/0004-3702(94)00041-X.
[31] M. Elvang-Gøransson, P. Krause and J. Fox, Dialectic reasoning with inconsistent information, in: *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence*, D. Heckerman and A. Mamdani, eds, Morgan Kaufmann Publishers Inc., 1993, pp. 114–121. doi:10.1016/B978-1-4832-1451-1.50018-4.
[32] T.-F. Fan and C.-J. Liau, A logic for reasoning about justified uncertain beliefs, in: *Proceedings of the IJCAI 2015*, Q. Yang and M. Wooldridge, eds, AAAI Press, 2015, pp. 2948–2954.
[33] M. Fitting, A logic of explicit knowledge, in: *Logica Yearbook 2004*, L. Běhounek and M. Bílková, eds, Filosofia, Prague, 2005, pp. 11–22.
[34] M. Fitting, The logic of proofs, semantically, *Annals of Pure and Applied Logic* **132**(1) (2005), 1–25. doi:10.1016/j.apal.2004.04.009.
[35] M. Fitting, Justification logics, logics of knowledge, and conservativity, *Annals of Mathematics and Artificial Intelligence* **53**(1–4) (2008), 153–167. doi:10.1007/s10472-009-9112-2.
[36] M. Fitting, Reasoning with justifications, in: *Towards Mathematical Philosophy*, Springer, 2009, pp. 107–123. doi:10.1007/978-1-4020-9084-4_6.
[37] M. Fitting, Possible world semantics for first-order logic of proofs, *Annals of Pure and Applied Logic* **165**(1) (2014), 225–240. doi:10.1016/j.apal.2013.07.011.
[38] M. Fitting, Modal logics, justification logics, and realization, *Annals of Pure and Applied Logic* **167**(8) (2016), 615–648. doi:10.1016/j.apal.2016.03.005.
[39] M. Fitting, Paraconsistent logic, evidence, and justification, *Studia Logica* **105**(6) (2017), 1149–1166. doi:10.1007/s11225-017-9714-3.
[40] J. Fox, D. Glasspool and J. Bury, Quantitative and qualitative approaches to reasoning under uncertainty in medical decision making, in: *Conference on Artificial Intelligence in Medicine in Europe, AIME 2001*, S. Quaglini, P. Barahona and S. Andreassen, eds, Springer, 2001, pp. 272–282.
[41] A.J. García and G.R. Simari, Defeasible logic programming: An argumentative approach, *Theory and Practice of Logic Programming* **4**(1–2) (2004), 95. doi:10.1017/S1471068403001674.
[42] A.J. García and G.R. Simari, Defeasible logic programming: DeLP-servers, contextual queries, and explanations for answers, *Argument & Computation* **5**(1) (2014), 63–88. doi:10.1080/19462166.2013.869767.

[43] K. Gödel, Vortrag bei Zilsel/Lecture at Zilsel's (1938a), in: *Kurt Gödel: Collected Works: Volume III: Unpublished Essays and Lectures*, Vol. 3, Oxford University Press, 1995, pp. 87–114.
[44] D. Grooters and H. Prakken, Two aspects of relevance in structured argumentation: Minimality and paraconsistency, *Journal of Artificial Intelligence Research* **56** (2016), 197–245. doi:10.1613/jair.5058.
[45] D. Grossi, Argumentation in the view of modal logic, in: *7th International Workshop on Argumentation in Multi-Agent Systems, ArgMAS 2010*, P. McBurney, I. Rahwan and S. Parsons, eds, LNCS, Vol. 6614, Springer, 2010, pp. 190–208.
[46] A. Hecham, P. Bisquert and M. Croitoru, On a flexible representation for defeasible reasoning variants, in: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018*, M. Dastani, G. Sukthankar, E. André and S. Koenig, eds, International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 1123–1131.
[47] J.F. Horty, *Reasons as Defaults*, Oxford University Press, 2012.
[48] I. Kokkinis, P. Maksimović, Z. Ognjanović and T. Studer, First steps towards probabilistic justification logic, *Logic Journal of the IGPL* **23**(4) (2015), 662–687. doi:10.1093/jigpal/jzv025.
[49] I. Kokkinis, Z. Ognjanović and T. Studer, Probabilistic justification logic, in: *International Symposium on Logical Foundations of Computer Science*, S.N. Artemov and A. Nerode, eds, LNCS, Vol. 9537, Springer, 2016, pp. 174–186. doi:10.1007/978-3-319-27683-0_13.
[50] R. Koons, Defeasible reasoning, in: *The Stanford Encyclopedia of Philosophy*, E.N. Zalta, ed., Metaphysics Research Lab, Stanford University, 2017.
[51] P. Krause, S. Ambler, M. Elvang-Gøransson and J. Fox, A logic of argumentation for reasoning under uncertainty, *Computational Intelligence* **11**(1) (1995), 113–131. doi:10.1111/j.1467-8640.1995.tb00025.x.
[52] R. Kuznets, On the complexity of explicit modal logics, in: *Computer Science Logic: 14th International Workshop, CSL 2000*, P.G. Clote and H. Schwichtenberg, eds, LNCS, Vol. 1862, Springer-Verlag, 2000, pp. 371–383. doi:10.1007/3-540-44622-2_25.
[53] R. Kuznets and T. Studer, *Logics of Proofs and Justifications*, College Publications, 2019.
[54] R.S. Milnikel, Derivability in certain subsystems of the Logic of Proofs is Π_2^p-complete, *Annals of Pure and Applied Logic* **145**(3) (2007), 223–239. doi:10.1016/j.apal.2006.03.001.
[55] R.S. Milnikel, The logic of uncertain justifications, *Annals of Pure and Applied Logic* **165**(1) (2014), 305–315. doi:10.1016/j.apal.2013.07.015.
[56] A. Mkrtychev, Models for the Logic of Proofs, in: *Logical Foundations of Computer Science, 4th International Symposium, LFCS '97*, S. Adian and A. Nerode, eds, LNCS, Vol. 1234, Springer-Verlag, 1997, pp. 266–275. doi:10.1007/3-540-63045-7_27.
[57] S. Modgil and H. Prakken, The ASPIC+ framework for structured argumentation: A tutorial, *Argument & Computation* **5**(1) (2014), 31–62. doi:10.1080/19462166.2013.869766.
[58] S.H. Nielsen and S. Parsons, A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments, in: *International Workshop on Argumentation in Multi-Agent Systems*, Springer, 2006, pp. 54–73.
[59] Z. Ognjanović, N. Savić and T. Studer, Justification logic with approximate conditional probabilities, in: *Logic, Rationality and Interaction, 6th International Workshop, LORI 2017*, A. Baltag, J. Seligman and T. Yamada, eds, LNCS, Vol. 10455, Springer, 2017, pp. 681–686.
[60] S. Pandžić, A logic of default justifications, in: *17th International Workshop on Nonmonotonic Reasoning, NMR 2018*, E. Fermé and S. Villata, eds, 2018, pp. 126–135.
[61] S. Pandžić, Reifying default reasons in justification logic, in: *Proceedings of the KI 2019 Workshop on Formal and Cognitive Reasoning, DKB-KIK 2019*, C. Beierle, M. Ragni, F. Stolzenburg and M. Thimm, eds, Vol. 2445, CEUR Workshop Proceedings, 2019, pp. 59–70.
[62] S. Pandžić, On the dynamics of structured argumentation: Modeling changes in default justification logic, in: *Foundations of Information and Knowledge Systems, 11th International Symposium, FoIKS 2020*, A. Herzig and J. Kontinen, eds, LNCS, Vol. 12012, Springer, 2020, pp. 222–241.
[63] J.L. Pollock, Defeasible reasoning, *Cognitive Science* **11**(4) (1987), 481–518. doi:10.1207/s15516709cog1104_4.
[64] J.L. Pollock, *Cognitive Carpentry: A Blueprint for How to Build a Person*, MIT Press, Cambridge, MA, 1995.
[65] J.L. Pollock, Defeasible reasoning with variable degrees of justification, *Artificial Intelligence* **133**(1–2) (2001), 233–282. doi:10.1016/S0004-3702(01)00145-X.
[66] J.L. Pollock, A recursive semantics for defeasible reasoning, in: *Argumentation in Artificial Intelligence*, I. Rahwan and G.R. Simari, eds, Springer, 2009, pp. 173–197. doi:10.1007/978-0-387-98197-0_9.
[67] H. Prakken, An argumentation framework in default logic, *Annals of Mathematics and Artificial Intelligence* **9**(1–2) (1993), 93–132. doi:10.1007/BF01531263.
[68] H. Prakken, An abstract framework for argumentation with structured arguments, *Argument and Computation* **1**(2) (2010), 93–124. doi:10.1080/19462160903564592.
[69] H. Prakken and J.F. Horty, An appreciation of John Pollock's work on the computational study of argument, *Argument & Computation* **3**(1) (2012), 1–19. doi:10.1080/19462166.2012.663409.

[70] G. Priest, Intensional paradoxes, *Notre Dame Journal of Formal Logic* **32**(2) (1991), 193–211. doi:10.1305/ndjfl/1093635745.
[71] A.N. Prior, On a family of paradoxes, *Notre Dame Journal of Formal Logic* **2**(1) (1961), 16–32. doi:10.1305/ndjfl/1093956750.
[72] R. Reiter, A logic for default reasoning, *Artificial Intelligence* **13**(1–2) (1980), 81–132. doi:10.1016/0004-3702(80)90014-4.
[73] B. Renne, Multi-agent justification logic: Communication and evidence elimination, *Synthese* **185**(1) (2012), 43–82. doi:10.1007/s11229-011-9968-7.
[74] C.-P. Su, T.-F. Fan and C.-J. Liau, Possibilistic justification logic: Reasoning about justified uncertain beliefs, *ACM Transactions on Computational Logic (TOCL)* **18**(2) (2017), 15. doi:10.1145/3091118.
[75] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, *Pacific Journal of Mathematics* **5**(2) (1955), 285–309. doi:10.2140/pjm.1955.5.285.
[76] F. Toni, A tutorial on assumption-based argumentation, *Argument & Computation* **5**(1) (2014), 89–117. doi:10.1080/19462166.2013.869878.
[77] S.E. Toulmin, *The Uses of Argument*, Cambridge University Press, 2003.
[78] G. Uzquiano, Quantifiers and quantification, in: *The Stanford Encyclopedia of Philosophy*, E.N. Zalta, ed., Metaphysics Research Lab, Stanford University, 2020.
[79] F.H. van Eemeren, B. Garssen, E.C.W. Krabbe, A.F.S. Henkemans, H.B. Verheij and J.H.M. Wagemans, Argumentation and artificial intelligence, in: *Handbook of Argumentation Theory*, Springer, 2014, pp. 615–675.
[80] B. Verheij, DefLog: On the logical interpretation of prima facie justified assumptions, *Journal of Logic and Computation* **13**(3) (2003), 319–346. doi:10.1093/logcom/13.3.319.
[81] B. Verheij, The Toulmin argument model in artificial intelligence, in: *Argumentation in Artificial Intelligence*, I. Rahwan and G.R. Simari, eds, Springer, 2009, pp. 219–238. doi:10.1007/978-0-387-98197-0_11.
[82] Y. Wu and M. Podlaszewski, Implementing crash-resistance and non-interference in logic-based argumentation, *Journal of Logic and Computation* **25**(2) (2015), 303–333. doi:10.1093/logcom/exu017.
[83] M. Zorn, A remark on method in transfinite algebra, *Bulletin of the American Mathematical Society* **41**(10) (1935), 667–670. doi:10.1090/S0002-9904-1935-06166-X.