Arguing to Motivate Decisions

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (`http://www.icis.decis.nl/`), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

SIKS Dissertation Series No. 2011-33
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by Wöhrmann Print Service, Zutphen
LaTeX template by Susan van den Braak

ISBN 978-90-393-56494

Arguing to Motivate Decisions

Argumentatie voor het Motiveren van Beslissingen

(met een samenvatting in het Nederlands)

PROEFSCHRIFT

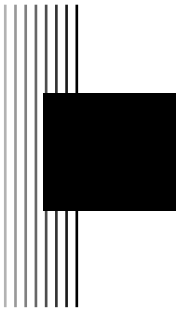
ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. G.J. van der Zwaan, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op maandag 10 oktober 2011 des middags te 12.45 uur

door

Thomas Leonardus van der Weide

geboren op 27 juli 1982 te Nijmegen

Promotoren:	Prof. dr. J.-J. Ch. Meyer
	Prof. dr. mr. H. Prakken
Co-promotoren:	Dr. F.P.M. Dignum
	Dr. G.A.W. Vreeswijk



Contents

1 Introduction

Consider a critical and complex decision situation such as being a fire commander in a situation where there is a fire in a factory containing toxic chemicals and with people trapped inside. The fire commander has to decide what course of action has to be taken. This decision impacts not only the lives of the victims trapped inside the factory, but also the lives of the fire fighters that may have to rescue them. Moreover, it is possible that the toxic chemicals inside the factory will escape into the environment. Because of the complexity of such decisions and the limited amount of time that is available, it is both useful and interesting to investigate how such decisions can be supported.

The purpose of a *decision support system* is to aid a decision-maker in determining what decision he[1] should take. Decision-making roughly requires determining (1) what alternatives are available; (2) in what outcome each alternative results; and (3) what expected outcome is preferred the most. Decisions can be supported in various ways, but in this thesis we will focus on how to support determining which expected outcome is preferred the most. It is natural for people to use argumentation to reason about and to communicate what decision is the best (Shafir et al., 1997). To support decision making in a natural way, we therefore address the following main research question in the thesis.

> **Main Research Question**:
> How can argumentation be used to support decision making in complex scenarios?

Running Example

In this thesis we will use a 'running example' to demonstrate what is needed in existing work and how our proposed framework improves upon existing work. The running example is in the context of so-called 'serious games'. Although computer games are mainly used to entertain, serious games are increasingly used for training and education (Michael and Chen, 2005). For example, Johnson et al. (2005) use serious games to teach language and Schurr et al. (2005) use them as a training tool for incident commanders. In particular, we are interested in serious games where students train to become fire commanders and learn how to make decisions in complex scenarios. When there is a fire, a fire commander has to

[1] For brevity, we use 'he' to refer to 'he or she' and 'his' to refer 'his or her'.

consider all relevant information and make a decision about what the fire fighters will do. In the computer game, the student plays the character of fire commander in a virtual world in which there potentially are multiple fires. The student has to decide what the fire fighters should do. There are many types of fires that a fire commander may have to deal with. Some fires are relatively simple (e.g., a burning car in a remote area), but others are complicated.

Example 1.1 (Running Example: Fire in a factory) There is a fire in a factory where toxic chemicals are stored. Inside the factory, there are several people whose lives are in danger and who need help to get out. Because the fire is damaging the factory, there is an increased risk that the toxic chemicals escape and damage the environment and health of the people in the surrounding area. The fire commander arrives at the scene and has to decide what course of action should be taken.

Although such complex situations do not occur frequently, the possible consequences are serious. It is thus important that fire commanders train how to act in such scenarios. The complexity of such decisions arises roughly from the following problems. On the one hand, there is uncertainty w.r.t. the outcome of actions and what the current situation exactly is (Bharosa et al., 2010). On the other hand, there are many different aspects to what makes one decision better than another and all these aspects are different by nature. Consequently, determining what decision is better is difficult, even if there is certainty concerning their outcomes. For example, in the running example the following perspectives may be important to a decision maker: the safety of the victims trapped inside, the safety of the people in the surrounding area, and the safety of the firefighting personnel, but also the impact on the environment, the costs and trouble for people e.g., caused by when a road is closed. When determining what decision to take, it is possible that one or multiple of these aspects are forgotten or that a mistake is made. To help a decision maker understand why a certain decision is better than another decision, a decision support system might argue to persuade the decision maker why the system believes that a certain decision should be taken. In this thesis, we focus on training how to determine what decision is better.

Supporting a user to make the best decision involves two aspects: the system needs to know what decision is the best and the system needs to be able to support the user effectively. Section 1.1 zooms in on how the system can determine what decision is the best for the user and Section 1.2 further investigates how a user can be supported. The introduction is ended by summarizing what we have discussed in the introduction and giving an overview of the rest of the thesis.

1.1 Identifying the Best Decision

To support a decision maker in making the best decision, a decision support system needs to have some idea about what decision is best. We will now further investigate what is required for this requirement.

1.1.1 Decision Theory

One possible technique that could be used to reason about and explain why a certain decision should be taken is utility theory. The interdisciplinary area of *decision theory* studies how

decisions are made and how they should be made (Keeney and Raiffa, 1976). Theories about how decisions are made by humans are called *descriptive*, see for example Kahneman and Tversky (1979), whereas theories about how decisions should be made are called *prescriptive* or *normative*. Prescriptive decision theories prescribe what is the rational choice for a person to make, which involves determining the future consequences of current decisions and the future preferences for those consequences (Savage, 1954).

Expected utility theory (EUT) is the most widely recognized prescriptive decision theory (Fishburn, 1970; von Neumann et al., 1947). A decision maker can choose from a set of *alternatives* and choosing an alternative results in an *outcome*. In other words, an outcome is a (future) consequence of choosing an alternative. In what outcomes alternatives may result is represented with a *probability distribution* over the outcomes given an alternative. Furthermore, EUT requires a *utility function* that maps outcomes of decisions to so-called utility values expressing the decision maker's satisfaction with each outcome. Given this probability distribution and the decision maker's utility function *the expected utility of an outcome* can be determined in a straightforward manner. EUT prescribes that a decision maker should choose the alternative with the maximal expected utility.

To apply EUT the following elements must be known: (1) the set of all alternatives the user can choose from; (2) the probability distribution of in what outcomes alternatives may result; and (3) the user's subjective utilities over outcomes. However, in decision situations like the one in the running example we cannot assume that these elements are all known. Moreover, because human decision makers do not reason in terms of quantitative utility values (Boutilier, 1994; Brafman and Domshlak, 2009), showing the calculations of the expected utilities of alternatives requires much effort of the decision maker to understand.

1.1.2 Value

A user's utility function is subjective and must be obtained before EUT can be applied. One way to elicit the user's utility function is to ask the user the utility value of each outcome. There are several problems with this approach. Obviously, if there are many possible outcomes, this process takes much time. But even when there are only two possible outcomes, expressing which one is preferred can be difficult for the user because there may be multiple aspects that matter, which may introduce tradeoffs that the user has to make. Furthermore, the concept of utility has an exact meaning in the EUT, but there is a great deal of misconception about what utility is and people do not think in utilities. Therefore, a user needs to translate its preferences to utilities. This translation is difficult and introduces many problems and errors.

Preference elicitation methods therefore focus on ordering outcomes by preference rather than asking utility values. These methods aim at easing the cognitive burden of ordering a set of outcomes, or finding an optimal item (Brafman and Domshlak, 2009). By strategically asking the user about his preferences between certain attributes of outcomes, the user does not need to specify its preferences between all outcomes. Preference elicitation methods consider several ways in which users can express their preferences. Users can give preference statements over outcomes, but also over attributes of outcomes. For example, the statement 'red cars are preferred to blue cars' expresses preferences over the colors of cars. Another way to express preferences is by using conditional preference statements such as 'given that the car is a Ferrari, red cars are preferred to blue cars'.

In order for a user to express his preferences he must know his preferences. However, in a complex decision situation, it is often not clear what to prefer. People often do not have well-established preferences a priori the decision making, but construct their preferences during decision making (Doyle and Thomason, 1999; March, 1978; Payne et al., 1992; Searle, 2001). Moreover, people have difficulties predicting future experiences of enjoyment and discomfort (Kahneman and Snell, 1990). For example, when deciding what car to buy, the decision-maker may predict that he will really enjoy a surround speaker set. However, after buying the car, the decision-maker finds out that he only listens to the radio, which is in mono. In critical scenarios, decision makers must evaluate the possible decisions from all the perspectives that they care about in order to determine what decision to prefer. If an important perspective is not considered, then a suboptimal decision may be made. Preference elicitation methods aim at eliciting the user's preferences, but are not concerned with supporting the user constructing his preferences, which is an important feature for decision support systems for complex decisions. In this thesis we are interested in how argumentation can be used in a decision support system to help a decision maker construct his preferences.

In the field of *decision analysis*, approaches have been developed concerning how to determine the utility function of a decision maker (Keeney and Raiffa, 1976). Keeney (1992) argues that when people are presented with a decision opportunity, they focus on a few alternatives that quickly come to mind. Given this set of alternatives, they start thinking of criteria that they could use to compare them. Keeney also argues that people should focus more on what they value. Namely, when a decision opportunity presents itself, a decision maker should focus thinking about why he cares about this opportunity and what objectives he should pursue rather than focusing on a few obvious alternatives. By doing so, the decision maker discovers what really matters and may find alternatives that were not obvious in the beginning.

To establish the utility function that accurately describes what a decision maker cares about, a decision maker and a decision analyst together analyze what the decision maker values. When the decision maker expresses that he wants to maximize a certain criterion, then the decision analyst will ask why this is so. If the decision maker answers that he has a certain objective or that he holds a certain value, then the decision analyst can ask whether there are other possible objectives and criteria that might be more suitable. In this way, the decision analyst starts to get a better understanding of what the decision maker really cares about. Moreover, the expertise and knowledge of the decision analyst can be used to propose alternative criteria and objectives that the decision maker might better use. In essence, the decision analyst and the decision maker are having a dialogue in which they exchange arguments, counterarguments, and questions for further justification. Consider the running example in which the decision maker may express that he wants to minimize the amount of smoke that is caused by the fire. Instead of simply accepting this objective, the decision analyst could ask why the decision maker wants this. In reply he might answer that smoke is bad for the environment and that he cares about the environment. Given this better understanding of what the decision maker really cares about (i.e., the environment and not about the smoke), the decision analyst could argue that in order to minimize the impact on the environment, the decision maker should also minimize the amount of toxic chemicals that escape into the environment, a perspective that the decision maker may have forgotten to take into account. In such a dialogue between the decision maker and the decision support system, arguments are exchanged justifying and attacking why certain perspectives should be taken

into consideration and why then a certain decision should be preferred. By doing so, the decision support system supports the decision maker in determining what he finds important, or, in other words, constructing the decision maker's preferences.

1.1.3 Research Questions on Argumentation

In order for a decision support system to use argumentation to justify and refute preferences, the system needs to be able to construct arguments and counterarguments concerning preferences between decisions. It is therefore necessary to have a formal description of argumentation, so that argumentation can be implemented in software. The field of *computational argumentation theory* is concerned with formal approaches of argumentation theory. Several approaches in argumentation theory have addressed the question of how to argue about what decision to take. In decision theoretic approaches the notion of preferences is used to express value and in argumentation approaches the notions of goals and the values a person holds are used to express value. In order to combine the strength of both kinds of approaches, we need to specify how these various notions relate to each other. This need is expressed in the following research question.

Research Question 1a:
What concepts are required to reason about what a decision maker values?

To answer this research question, Chapter 3 proposes a conceptual framework that informally describes several concepts related to value and proposes how to reason with them. More specifically, the notion of *perspective* is introduced and used to compare outcomes of decisions. Preferences are always seen from a perspective. Abstract values from the psychology literature are then taken as the basic values that people hold. Following the decision analysis literature, we then propose how to decompose an abstract value qualitatively into objectives and into specific evaluation criteria. Given the decomposition of a person's preferences, we then propose argument schemes to reason about what outcome to prefer. To develop software that can support a decision and because a computer cannot work with informal concepts, these concepts need to be formalized. Chapter 4 formalizes the conceptual framework that is proposed in Chapter 3.

If a user expresses a preference between two alternatives, then this does not mean that this accurately predicts his preferences. There are several reasons why this preference statement may be incorrect. Suppose the user justifies his preferences on a previous experience. Then the user's preferences may have changed or the user may have made an error in concluding that the current situation is similar to the situation in which the previous experience occurred. In contrast, perhaps the user forgot to consider an aspect that mattered to him while determining what to prefer. By supporting a user in constructing his preferences, such errors can be detected and resolved. It is therefore interesting and useful to support a user in constructing his preferences. To support a user in constructing its preferences, the system needs to be able to reason about and with preferences. This requires justifying and attacking preference statements.

Most approaches in the argumentation literature assume that what a decision maker values is given and is described in terms of goals he wants to achieve and values he wants to promote. For example, Atkinson et al. (2006) use goals and values to represent what matters to a decision maker. Although in Atkinson and Bench-Capon (2007a,b) it is possible that

agents disagree about what a value means, it is not possible to discuss what a value means to the decision maker, what is the best way to measure whether a value has been promoted or demoted, and whether a goal or state transition really promotes or demotes a value. In more complex decisions these aspects are not clear, e.g., in the running example what does fairness mean and how can you best measure it. Therefore, it is useful that a decision maker can be supported in determining this. A second example is Amgoud and Prade (2009), where what a decision maker values is represented with goals he wants to achieve and so-called rejections he wants to avoid. It is not possible to discuss why the decision maker has a certain goal or rejection. A third example is Modgil (2007, 2009), where argumentation is used to reason about preferences. However, the relation between preferences and values and goals is not clear in this work. These problems lead us to the following research question.

Research Question 1b:
How can argumentation be used to reason about, justify and refute what a decision maker values?

To answer this question, the conceptual framework proposed in Chapter 3 must first be formalized, which is done in Chapter 4. Chapter 5 then addresses research question 1b by proposing an argumentation logic that is based on Chapter 4's formalization of Chapter 3's conceptual framework.

Research question 1b concerns reasoning about what a decision maker values. Although this is necessary to make a decision, more is needed to make an actual decision. Reasoning about what to do is called *practical reasoning*. In the literature we can find several approaches that use formal argumentation theory for practical reasoning. Atkinson et al. (2006) proposed an argument scheme for practical reasoning that justifies performing an action if it achieves a goal of the decision maker that promotes a value that he holds. Sixteen critical questions are formulated that point to how this argument scheme can be attacked. For example, an argument that justifies an action can be attacked if that action does not achieve a certain goal or demotes a value of the decision maker. A second paper on using argumentation for decision making is Amgoud and Prade (2009). They proposed an abstract argumentation framework to make decisions in two steps: (1) arguments for beliefs and about alternatives are constructed, and (2) the alternatives are compared using several decision principles. The formalism assumes that the decision maker has a set of goals that he wants to achieve and a set of so-called rejections that he wants to avoid.

Furthermore, Simon (1957) points out that both human and artificial agents are *resource-bounded*, meaning that they are unable to execute an arbitrarily large number of computations in a constant time. Several approaches for practical reasoning have been proposed that take into account resource-boundedness. For example, Bratman et al. (1988) proposed an architecture for resource-bounded practical reasoning which distinguishes between means-end reasoning (finding suitable plans) and weighing competing alternatives.

These two requirements, i.e., the ability to justify making a decision and to do that with a limited amount of resources, lead us to the following research question.

Research Question 1c:
How can argumentation be used for resource-bounded practical reasoning using our framework for value?

To answer this research question, Chapter 6 extends the argumentation framework of Chapter 5 with two different notions of goals, that are used as a simplified representation of value. This simplification is inspired by Simon (1957) and allows comparing decisions using less resources. Furthermore, the notion of a goal is commonly used in daily conversations and therefore important to consider in our framework.

1.2 Research Questions on Dialogues

In order for a computer system to support the motivation behind decisions using argumentation, the system needs to represent and reason with arguments about decisions. The previous research questions have focused on this issue. In addition, the system needs to explain to the user why a certain decision should be taken, but it also needs to be able to respond to counterarguments and counterproposals that the user may give. To do this in an interactive manner, the system could have a dialogue with the user. To demonstrate the kind of dialogues we envision for decision support, consider the following example dialogue between fire commander Alice and decision support system Bob.

> ALICE: I should send the firefighters in to rescue the victims because it achieves my goal with respect to safety
> BOB: how do you measure safety?
> ALICE: by the amount of time that these victims are inside
> BOB: it is better to look at how long these victims are near fire
> ALICE: okay, but then I should still send the firefighters in to rescue them
> BOB: don't forget to also consider the safety of your personnel
> BOB: therefore, you should first extinguish the fire near the victims and then send the firefighters in to rescue them

Because dialogues are a natural way for people to exchange arguments, researchers in argumentation theory have investigated dialogue systems. For example, a logic-based formalism is proposed by McBurney and Parsons (2002) which represents complex dialogues as sequences of *dialogue moves* and allows dialogues to be embedded in one another. Prakken (2005a) proposes a general dialogue framework that agents could use to discuss a topic. In this framework, agents can put forward arguments, ask questions, and concede and retract premises. To ensure that dialogues have desirable properties, different kinds of protocols could be implemented. To regulate the interactions between agents in dialogues where they try to persuade each other, Prakken (2006) proposes a formal specification of the main elements of dialogue systems for persuasion. McBurney et al. (2007) propose a framework for deliberation dialogues. In a deliberation dialogue a group of agents discuss what joint action they should perform, which is related to the kind of dialogues that we envision for supporting decisions.

Because decision support dialogues are not exactly the same as deliberation or persuasion dialogues, we need to investigate what is necessary in a dialogue for our purposes. This brings us to the following research question.

> **Research Question 2a**:
> How can we formally represent a dialogue framework for arguing to motivate decisions?

At a particular point in a decision support dialogue, it is not uncommon that the system can choose from many possible dialogue moves. Because of the limited amount of time in the decision scenarios we envision (e.g., crisis management) it is therefore important that the system carefully considers what dialogue move to select. There will be a variety of possibly conflicting aspects that matter to the system when determining what dialogue move to select. For example, the system may want to minimize the duration of the dialogue, but may also want to be comprehensive. In the literature many criteria have been proposed to select a dialogue move. For example, Bench-Capon (2003); Perelman and Olbrechts-Tyteca (1969) argue that in order to be persuasive arguments have to be selected that use the values the audience holds, which is used in dialogue systems like Atkinson et al. (2006) and Black and Atkinson (2011). Hunter (2004b) formally defines the 'impact' of an argument. In Amgoud and de Saint Cyr (2008) several measures are proposed that can be used as criteria, such as aggressiveness and coherence, to determine the quality of a persuasion dialogue. Oren et al. (2007) introduce the criterion to minimize the amount of information that is shared, which may be important in domains where the system has privacy-sensitive information.

These are just some of the possible criteria that can be found and used in determining what dialogue move to select. Because multiple criteria may matter in a particular decision support system, this naturally leads us to the following research question.

> **Research Question 2b**:
> How can a decision support system reason about what move to make in a dialogue?

We will address this research question in Chapter 8.

1.3 Overview of the Thesis

In this introduction we have argued for the need for a decision support system that can argue to motivate decisions and posed a number of research questions that need to be addressed for this purpose. First we have looked closer at what is required for a formal system to reason about, justify and refute what a decision maker values and what decision he should take. Then we have further examined how a user can be supported in an interactive manner using dialogues in which the participants exchange arguments.

In Chapter 2, some background is given on computational argumentation theory, decision theory and the role of abstract values in decision making. After the background, each chapter corresponds to a research question given in this introduction. Chapter 3 proposes a conceptual framework for value to address research question 1a. Chapter 3 is based on previous work described in van der Weide et al. (2010, 2009b,c). Research question 1b asks how argumentation can be used to reason about value. To answer this question, Chapter 4 formalizes the conceptual framework of Chapter 3. Next, Chapter 5 uses the formalization of Chapter 4 to propose an argumentation framework to reason about value and thereby answers research question 1b. Research question 1c asks how to use argumentation for practical reasoning when agents are resource-bounded. Chapter 6 extends the argumentation framework of Chapter 5 for this purpose.

To answer research question 2a, Chapter 7 proposes a dialogue framework for supporting decisions. Chapter 7 is based on van der Weide and Dignum (2011). While supporting a

decision maker's decision, it is common that the support system has to choose from many possible alternative moves in the dialogue. Research question 2b asks how to reason about what move should be selected. This research question is addressed in Chapter 8, which is based on van der Weide et al. (2009a, 2011). The thesis is concluded in Chapter 9.

2 Background

The topic of this thesis is arguing to motivate decisions. Because it is natural to use arguments when discussing something, we are interested in software systems that can support a decision maker by giving arguments and counterarguments that explain, justify and refute the opinion of the software. To do this, a software system needs to be able to represent and reason with arguments. In addition, the system needs to understand how decisions are and should be made. Because much of research has focused on these two topics, the goal of this chapter is to give a background on them.

The first three sections concern argumentation. Section 2.1 describes one logic-based argumentation framework, the ASPIC+ framework, which has been used for a variety of purposes. Although strictly speaking it does not belong in the background chapter, we will extend the ASPIC+ framework in two ways that are required to argue to motivate decisions. Firstly, in decision making it is common that for each possible decision there are multiple arguments in favor of a decision and multiple reasons against a decision. To compare decisions, the arguments with the same conclusion need to be combined. This is called *accrual*. To argue to motivate decisions it is thus necessary to integrate accrual into the ASPIC+ framework. Section 2.2 makes the the accrual mechanism by Prakken (2005b) more precise and integrates the mechanism in the ASPIC+ framework. Secondly, when arguing about what to do it is also common to reason about the relative strength of arguments. Reasoning *about* the relative strength (object-level) arguments is done on a *meta level* w.r.t. the object-level arguments. The ASPIC+ framework also does not provide any specific tools for meta-level reasoning and therefore we propose a general meta-level argumentation framework in Section 2.3.

Next we will give a background on decision theory in Section 2.4. Because the *(abstract) values* a person holds have a significant effect on that person's motivation and are thus essential in what decision is the best for that person, we will give a background on abstract values in psychology in Section 2.5 and abstract values as used in argumentation theory in Subsection 2.5.4.

2.1 Argumentation

In order for a computer system to argue with a human user about a topic such as what decision to make, Section 2.1.1 defines what an argument is. An argument justifies a conclusion by

applying several inferences on a set of premises. For example, an argument could justify the conclusion 'user should rescue victims' with the premises 'user has the goal that the victims are safe' and 'rescuing victims achieves that goal'. Some of the inferences that are made are strict, meaning that the inference is true without exception. However, other inferences are what is called 'defeasible', which means that the inference presumably holds, but that there are exceptions in which the inference cannot be made. *Argument schemes* (Walton, 1996) are patterns of reasoning that are commonly used by people, but that cannot be modeled as strict inferences since they are presumptive. An argument scheme consists of a set of premises and a conclusion. If the premises are true, then, presumably, the conclusion is true. Because argument schemes are presumptive, critical questions can be associated with them that point to exceptional situations when the scheme cannot be applied (Verheij, 2003). When arguing what to do, many inferences that are made are not strict but defeasible and can thus better be modeled as argument schemes. Therefore, arguing what to do requires an argumentation system in which both strict and defeasible inferences can be made.

Arguments are constructed from a knowledge base. For example, the knowledge base may contain 'extinguishing the fire in the factory takes twenty minutes' or 'the user has the goal to rescue the victims'. However, the arguments that are constructed from a knowledge base may conflict with each other. For example, one argument says that the user should do action 1 and another argument says that the user should do action 2. In that case we say that one argument *attacks* another argument. Subsection 2.1.2 distinguishes between several kinds of attacks between arguments. Not all arguments are equally strong. For example, one argument to do an action may be stronger than another argument to do that action. The relative strength of arguments determines whether an attack is successful. Subsection 2.1.3 introduces the relative strength of arguments and the notion of defeat. Given a set of arguments and how they attack each other, we want to know what conclusions are acceptable. To do so, Subsection 2.1.4 describes Dung Argumentation Frameworks and what conclusions are acceptable can be determined with different criteria. Finally, we want to ensure that the acceptable conclusions satisfy several rationality postulates proposed in the literature. Subsection 2.1.5 describes several rationality postulates and shows under what conditions argumentation systems follow them.

To represent arguments, we will use the ASPIC+ abstract framework for structured argumentation, which provides an abstract account of the structure of arguments, the nature of attack and the effect of the relative strength of arguments on what attacks are successful (Prakken (2010)). For example, consider argument A stating "rescue the victims before extinguishing the fire because it saves their lives" and argument B stating "extinguish the fire before rescuing the victims because it is better for the environment". In ASPIC+, arguments A and B have a structure using logic. Furthermore, A and B attack each other because only one of these alternatives can be chosen. However, if A is a stronger argument than B, then the attack of A on B is successful whereas the attack of B on A is unsuccessful.

2.1.1 Argumentation Systems

For arguing to motivate decisions we need an argumentation framework that allows modeling both strict and defeasible inference rules (in the form of argument schemes) and where arguments differ in their relative strength. A suitable framework for this is ASPIC+ (Prakken, 2010), which extends and generalizes ASPIC. The ASPIC abstract framework for structured

argumentation (Amgoud et al., 2006) integrates work on rule-based argumentation with the abstract approach by Dung (1995). The ASPIC+ framework generalizes and extends the ASPIC framework in four ways: (1) a new way of attacking is introduced called undermining, i.e., attacking an argument on its premises; (2) attacks are generalized by an abstract relation of contrariness between formulae; (3) four types of premises are distinguished; and, (4) the successfulness of attacks depends on the relative strength of arguments. By extending and generalizing ASPIC in these ways, ASPIC+ has a number of advantages. Firstly, ASPIC+ captures more kinds of argumentation systems. Secondly, argument schemes can be formalized in ASPIC+. Lastly, ASPIC+ satisfies the rationality postulates proposed by Caminada and Amgoud (2007) for the more general case in which relative strength of arguments is accounted for. Because ASPIC+ captures more argument schemes and allows arguments to differ in their relative strength, we will use ASPIC+ in this thesis.

We will now introduce the ASPIC+ framework. The notion of an argumentation system extends the familiar notion of a proof system by distinguishing between strict and defeasible inference rules. The distinction between strict and defeasible rules was already made by Pollock (1987). The informal reading of a strict inference rule is that if its antecedent holds, then its conclusion holds without exception. For example, modus ponens is the strict inference rule with antecedents 'ϕ' and '$\phi \supset \psi$' and conclusion 'ψ' (where ϕ and ψ are well-formed formulae). The informal reading of a defeasible inference rule is that if its antecedent holds, then its conclusion tends to hold. Defeasible inference rules can be used to formalize presumptive reasoning patterns like argument schemes. An example defeasible inference rule is the presumptive scheme 'argument from expert opinion' (Walton (1996)): "if person P is an expert in the domain D, the statement S is in the domain of D, and P asserted that S is true, then, presumably, S is true". Deductively, this inference cannot be made even though it will typically be a correct inference. By modeling this argument scheme as a defeasible inference, it is possible to use this conclusion until evidence is found that states there is an exceptional situation in which the conclusion is not true.

Definition 2.1 (Argumentation System) *An* argumentation system[1] *is a tuple* $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ *with*

- $\mathcal{L}$ *is a logical language,*
- $\mathcal{R} = \mathcal{SR} \cup \mathcal{DR}$ *such that* $\mathcal{SR}$ *is a set of strict and* $\mathcal{DR}$ *is a set defeasible inference rules, and*
- cf *a contrariness function from* $\mathcal{L}$ *to* $2^{\mathcal{L}}$

We will use ϕ and ψ as typical elements of $\mathcal{L}$ and say that ϕ and $\neg\phi$ are each other's complements. In the meta-language, $\sim\phi$ denotes the complement of any formula ϕ, positive or negative. Furthermore, $\supset$ denotes the material implication. A contrariness function expresses conflict between formulae and is defined by Prakken as follows. We will assume that it must always be the case that $\neg\phi \in \mathsf{cf}(\phi)$ and $\phi \in \mathsf{cf}(\neg\phi)$.

Definition 2.2 (Contrariness Function) *A* contrariness function for a logical language $\mathcal{L}$ *is a function* $\mathsf{cf} : \mathcal{L} \rightarrow 2^{\mathcal{L}}$ *that at least has the following property for all formulae* ϕ *in* $\mathcal{L}$*:* $\neg\phi \in \mathsf{cf}(\phi)$ *and* $\phi \in \mathsf{cf}(\neg\phi)$.

[1] This definition adapts Prakken (2010)'s definition by removing the partial preorder over defeasible inference rules because this will be represented later in this thesis in a meta-level argumentation system that is used to reason about the relative strength of object-level arguments.

If $\phi \in \mathsf{cf}(\psi)$ and $\psi \in \mathsf{cf}(\phi)$, then ϕ and ψ are called *contradictory* and we will also write $\phi = -\psi$. Furthermore, non-symmetrical conflict is allowed like in Bondarenko et al. (1997). If $\phi \in \mathsf{cf}(\psi)$ and $\psi \notin \mathsf{cf}(\phi)$, then ϕ is called the *contrary* of ψ. Given a contrariness function we will now define when a set of formulae is called consistent.

Definition 2.3 (Consistent Set) *Let $\mathcal{L}$ be a logical language and* cf *a contrariness function for $\mathcal{L}$. A set X of formulae in $\mathcal{L}$ is* consistent *iff there are no $\phi, \psi \in X$ such that $\phi \in \mathsf{cf}(\psi)$.*

An argumentation system contains a set of strict and defeasible rules, which are defined formally as follows.

Definition 2.4 (Strict and Defeasible Rules) *Let $\phi_1, \ldots, \phi_m, \phi$ (with $m \geq 0$) be expressions in logical language $\mathcal{L}$.*

- *A* strict inference rule *is of the form $s : \phi_1, \ldots, \phi_m \rightarrow \phi$*
- *A* defeasible inference rule *is of the form $d : \phi_1, \ldots, \phi_m \Rightarrow \phi$*

We will call s and d the *rule name*, $\phi_1, \ldots, \phi_m$ the *antecedent* of the rule, and ϕ the *conclusion*. The conclusion of a strict inference rule holds without exception if the antecedent is true, whereas the conclusion of a defeasible inference rule tends to be true if the antecedent is true, that is, the conclusion is presumably true, but there are exceptions. Later chapters of this thesis will introduce new inference rules that are specified by schemes in which the antecedent and the conclusion are meta-variables over $\mathcal{L}$. Using schemes to specify inference rules is a common approach in logic.

This definition extends Prakken's definition by adding a rule name to each inference rule. This will be convenient when modeling certain kinds of attacks between arguments. We will assume that for every rule name r there is a constant $\underline{r}$ in the logical language that denotes r and that there is a unary predicate appl in the logical language and that $\mathsf{appl}(\underline{r})$ denotes that the inference rule with rule name r is applicable. Similarly, $\neg\mathsf{appl}(\underline{r})$ denotes that inference rule r is not applicable. For example, if d is the rule name of a defeasible rule, then $\neg\mathsf{appl}(\underline{d})$ denotes that there is an exceptional situation where d cannot be applied. The appl predicate is necessary to model undercutting attack between arguments in Section 2.1.2.

Arguments are defined following Vreeswijk (1997) and can be thought of as proof trees. The following functions are defined on arguments: (1) conc returns the argument's conclusion; (2) premises returns the argument's premises; (3) lastRule returns the last applied inference rule; (4) rules returns all applied inference rules; (5) lastDef returns the last applied defeasible inference rules; (6) dirSub returns the argument's direct subarguments; and, (7) sub returns all subarguments of the argument.

Definition 2.5 (Argument) *Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ be an argumentation system. An* argument *A* in *$\mathcal{AS}$ is either*

- *ϕ if ϕ is a formula in $\mathcal{L}$ with*
  - $\mathsf{conc}(A) = \phi$
  - $\mathsf{premises}(A) = \{\phi\}$
  - $\mathsf{lastRule}(A) = \mathsf{undefined}$

  - $\mathsf{rules}(A) = \emptyset$
  - $\mathsf{lastDef}(A) = \emptyset$
  - $\mathsf{dirSub}(A) = \emptyset$
  - $\mathsf{sub}(A) = \{A\}$

- $A_1, \ldots, A_n \rightarrow \phi$ *if* $A_1, \ldots, A_n$ *are arguments such that there is a strict rule* $s : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \rightarrow \phi$ *in* $\mathcal{SR}$.
  - $\mathsf{conc}(A) = \phi$
  - $\mathsf{premises}(A) = \bigcup_{i=1}^{n} \mathsf{premises}(A_i)$
  - $\mathsf{lastRule}(A) = s : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \rightarrow \phi$
  - $\mathsf{rules}(A) = \{s : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \rightarrow \phi\} \cup \bigcup_{i=1}^{n} \mathsf{rules}(A_i)$
  - $\mathsf{lastDef}(A) = \bigcup_{i=1}^{n} \mathsf{lastDef}(A_i)$
  - $\mathsf{dirSub}(A) = \{A_1, \ldots, A_n\}$
  - $\mathsf{sub}(A) = \{A\} \cup \bigcup_{i=1}^{n} \mathsf{sub}(A_i)$
- $A_1, \ldots, A_n \Rightarrow \phi$ *if* $A_1, \ldots, A_n$ *are arguments such that there is a defeasible rule* $d : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \Rightarrow \phi$ *in* $\mathcal{DR}$.
  - $\mathsf{conc}(A) = \phi$
  - $\mathsf{premises}(A) = \bigcup_{i=1}^{n} \mathsf{premises}(A_i)$
  - $\mathsf{lastRule}(A) = d : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \Rightarrow \phi$
  - $\mathsf{rules}(A) = \{d : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \Rightarrow \phi\} \cup \bigcup_{i=1}^{n} \mathsf{rules}(A_i)$
  - $\mathsf{lastDef}(A) = \{d : \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \Rightarrow \phi\}$
  - $\mathsf{dirSub}(A) = \{A_1, \ldots, A_n\}$
  - $\mathsf{sub}(A) = \{A\} \cup \bigcup_{i=1}^{n} \mathsf{sub}(A_i)$

We will use $\mathsf{Args}(\mathcal{AS})$ to denote the set of all arguments in $\mathcal{AS}$. Arguments can be visualized as inference trees.

Example 2.1 Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ be an argumentation system with $\mathcal{R} = \mathcal{DR} \cup \mathcal{SR}$ and

$$\begin{aligned} \mathcal{L} &= \{\phi_1, \phi_2, \phi_3\} \\ \mathcal{SR} &= \{s : \phi_1 \rightarrow \phi_2\} \\ \mathcal{DR} &= \{d : \phi_1, \phi_2 \Rightarrow \phi_3\} \end{aligned}$$

Let us consider the following three well-formed atomic arguments in $\mathcal{AS}$: argument A_1 concluding ϕ_1, argument A_2 concluding ϕ_2, and argument A_3 concluding ϕ_3. They have the properties as described in Table 2.1.

There are three 'compound' arguments that can be constructed from the atomic arguments A_1, A_2 and A_3. Their properties are described in Table 2.2. The compound arguments A_4, A_5 and A_6 can be visualized as follows.

$$A_4 = \frac{\phi_1}{\phi_2}\ s \qquad A_5 = \frac{\phi_1 \quad \phi_2}{\phi_3}\ d \qquad A_6 = \frac{\phi_1 \quad \dfrac{\phi_1}{\phi_2}\ s}{\phi_3}\ d$$

Because the defeasible inference rule d has been applied in arguments A_5 and A_6, they are defeasible arguments. In the other arguments no defeasible inference rule has been applied and therefore they are strict.

Table 2.1: *Well-Formed Atomic Arguments in Example 2.1*

	A_1	A_2	A_3
$\mathsf{conc}(A_i)$	ϕ_1	ϕ_2	ϕ_3
$\mathsf{premises}(A_i)$	$\{\phi_1\}$	$\{\phi_2\}$	$\{\phi_3\}$
$\mathsf{lastRule}(A_i)$	undefined	undefined	undefined
$\mathsf{rules}(A_i)$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathsf{lastDef}(A_i)$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathsf{dirSub}(A_i)$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathsf{sub}(A_i)$	$\{A_1\}$	$\{A_2\}$	$\{A_3\}$

Table 2.2: *Well-Formed Compound Arguments in Example 2.1*

	A_4	A_5	A_6
$\mathsf{conc}(A_i)$	ϕ_2	ϕ_3	ϕ_3
$\mathsf{premises}(A_i)$	$\{\phi_1\}$	$\{\phi_1, \phi_2\}$	$\{\phi_1\}$
$\mathsf{lastRule}(A_i)$	$s : \phi_1 \to \phi_2$	$d : \phi_1, \phi_2 \Rightarrow \phi_3$	$d : \phi_1, \phi_2 \Rightarrow \phi_3$
$\mathsf{rules}(A_i)$	$\{s : \phi_1 \to \phi_2\}$	$\{d : \phi_1, \phi_2 \Rightarrow \phi_3\}$	$\{s : \phi_1 \to \phi_2,\ d : \phi_1, \phi_2 \Rightarrow \phi_3\}$
$\mathsf{lastDef}(A_i)$	$\emptyset$	$\{d : \phi_1, \phi_2 \Rightarrow \phi_3\}$	$\{d : \phi_1, \phi_2 \Rightarrow \phi_3\}$
$\mathsf{dirSub}(A_i)$	$\{A_1\}$	$\{A_1, A_2\}$	$\{A_1\}$
$\mathsf{sub}(A_i)$	$\{A_1, A_4\}$	$\{A_1, A_2, A_5\}$	$\{A_1, A_6\}$

In our framework an agent constructs arguments from his knowledge base. A knowledge base contains formulae of a logical language. The elements in a knowledge base are also called *premises*. Following Prakken (2010), we distinguish between premises that are necessarily true, 'ordinary' premises and assumptions. Note that Prakken also considers so-called issues, i.e., controversial premises that must always be justified, but these are not important for our purposes. Intuitively, an argument with only necessary premises and which only applies strict rules cannot be attacked. In contrast, an attack on an assumption always succeeds.

Definition 2.6 (Knowledge Base) *Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ be an argumentation system. A* knowledge base for $\mathcal{AS}$ *is a tuple $\langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ where $\mathcal{K}_{\mathsf{np}}$, $\mathcal{K}_{\mathsf{op}}$ and $\mathcal{K}_{\mathsf{as}}$ are disjoint sets of formulae in $\mathcal{L}$.*

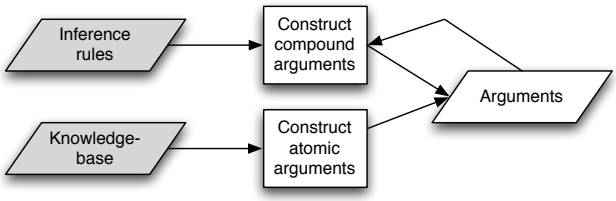
The set $\mathcal{K}_{\mathsf{np}}$ denotes the set of *necessary premises*, i.e., those premises that are necessarily true. $\mathcal{K}_{\mathsf{np}}$ should include all the axioms. Intuitively, necessary premises cannot be attacked. The set $\mathcal{K}_{\mathsf{op}}$ denotes the set of *ordinary premises*. In contrast, ordinary premises can be attacked. The set $\mathcal{K}_{\mathsf{as}}$ denotes the set of *assumptions*. Every attack on an assumption is successful. We say that *argument A can be constructed from a knowledge base* iff every premise of A is in that knowledge base.

Consider the flow chart[2] in Figure 2.1 that shows schematically how arguments can be constructed from a knowledge base and an argumentation system. The process 'construct

[2]A rectangle is a process and a parallelogram is a dataset. An arrow from a dataset to a process denotes that the process uses the data from that dataset. An arrow from a process to a dataset denotes that the process puts its result in that dataset. A grey dataset denotes that this dataset is required in order for the flow chart to work.

atomic arguments' constructs atomic arguments from the formulae in the knowledge base and puts those arguments in the dataset 'Arguments'. The process 'construct compound arguments' uses the inference rules in $\mathcal{R}$ together with existing arguments in the dataset 'Arguments' to construct new compound arguments, which are put in the dataset 'Arguments'.

Figure 2.1: *Flow Chart of the Construction of Arguments.*



We can now define several properties of arguments that will be useful later to determine the relative strength of arguments. These properties are taken from Prakken.

Definition 2.7 (Argument Properties) *Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ be an argumentation system and $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ be a knowledge base for $\mathcal{AS}$. An argument $A \in \mathsf{Args}(\mathcal{AS})$ is called*

- strict *if* $\mathsf{rules}(A) \cap \mathcal{DR} = \emptyset$,
- defeasible *iff* $\mathsf{rules}(A) \cap \mathcal{DR} \neq \emptyset$,
- firm *iff* $\mathsf{premises}(A) \subseteq \mathcal{K}_{\mathsf{np}}$, *and*
- plausible *iff* $\mathsf{premises}(A) \not\subseteq \mathcal{K}_{\mathsf{np}}$.

Intuitively, an argument that is both strict and firm is stronger than an argument that is defeasible or plausible.

2.1.2 Attack between Arguments

The definition of a knowledge base does not contain any constraints about whether the knowledge base should be consistent. It is thus possible that a knowledge base contains both ϕ and $\neg\phi$. Furthermore, although a knowledge base may not contain conflicting formulae, it may occur that conflicting conclusions are inferred using the defeasible inferences rules. In both cases, arguments with conflicting conclusions can be constructed from a knowledge base. In this section, we distinguish between several kinds of conflict between argument. If an argument conflicts with another argument, then we will say that they attack each other.

Prakken distinguishes between the following three kinds of attack: rebutting, undermining, and undercutting. Intuitively, if the conclusion of argument A is in conflict with an intermediary conclusion or the final conclusion of argument B, then we say that A rebuts B. An argument A rebuts argument B if A's conclusion is in conflict with the conclusion of the application of a defeasible inference rule applied somewhere in B.

Definition 2.8 (Rebutting Attack) Argument A rebuts argument B (on B') *iff there is a* $B' \in \mathsf{sub}(B)$ *such that* $\mathsf{lastRule}(B') \in \mathcal{R}_d$ *and* $\mathsf{conc}(A) \in \mathsf{cf}(\mathsf{conc}(B'))$.

We will say that argument A *contrary-rebuts* argument B on B' iff $\mathsf{conc}(A)$ is a contrary of $\mathsf{conc}(B')$. Note that if argument A rebuts argument B on B', then A rebuts every argument C that makes use of B', i.e., $B' \in \mathsf{sub}(C)$. Further note that the rebutting attack is not necessarily symmetric. For example, if A rebuts B on B' and A's last applied inference step is strict, then B' cannot rebut A.

If an argument attacks a premise of another argument, then we shall call this an undermining attack. However, necessary premises cannot be undermined. Formally:

Definition 2.9 (Undermining Attack) Argument A undermines argument B *iff there is a* $\phi \in \mathsf{premises}(B)$ *such that* $\mathsf{conc}(A) \in \mathsf{cf}(\phi)$ *and* $\phi \notin \mathcal{K}_{\mathsf{np}}$.

We will say that argument A *contrary-undermines* argument B iff $\mathsf{conc}(A)$ is a contrary of ϕ or ϕ is an assumption, i.e., $\phi \in \mathcal{K}_{\mathsf{as}}$.

There may be exceptions in which a defeasible inference rule cannot be applied. The application of a defeasible inference rule is denoted with the instantiated identifier of the rule. For example, if $d : \phi \Rightarrow \psi$ is a defeasible inference rule, then $\underline{d}$ is a constant in the logical language and the formula $\neg\mathsf{appl}(\underline{d})$ denotes that the defeasible inference rule d cannot be applied. If an argument A concludes that a defeasible inference rule cannot be applied in a certain case, then A undercuts arguments that use that application of that defeasible inference rule.

Definition 2.10 (Undercutting Attack) Argument A undercuts argument B (on B') *iff there is an argument* $B' \in \mathsf{sub}(B)$ *with* $r = \mathsf{lastRule}(B')$ *such that* $r \in \mathcal{R}_d$ *and* $\mathsf{conc}(A) = \neg\mathsf{appl}(\underline{r})$.

Note that if argument A undercuts argument B, it remains possible (although perhaps not likely) that a subargument of B attacks A.

Definition 2.11 (Attack) Argument A attacks argument B *iff* A *rebuts, undermines or undercuts* B.

2.1.3 Argument Strength and Defeat

Arguments may not necessarily have the same strength. For example, this can be caused by that one argument uses stronger premises than another argument. The relative strength (or conclusive force) of two arguments determines whether the attack of one argument on the other is successful. For example, an argument A_1 based on imprecise observations is weaker than an argument A_2 based on scientific facts. If A_1 and A_2 have conflicting conclusions, then they attack each other. However, because A_2 is stronger than A_1, only A_2's attack on A_1 should be successful. Note that even though A_2's attack on A_1 is successful, it is possible that A_2 is successfully attacked by another argument.

Following Prakken (2010); Vreeswijk (1997), strength of arguments is modeled as an ordering over arguments. Intuitively, the strength of an argument should at least depend on the strength of its premises and the strength of the inference rules that are applied. However, there is no consensus of how to determine the strength of arguments. For example, if argument A has strong premises but applies weak defeasible rules and argument B has weak premises but applies strong defeasible rules, then it is not clear when A should be stronger than B. There is

some consensus regarding several minimal conditions that argument orderings should follow. Namely, argument orderings must be what is called 'admissible', which means that strict and firm arguments are always preferred to arguments that are not strict and firm, and that a strict inference cannot make an argument strictly better or worse than its weakest proper subargument.

Definition 2.12 (Argument Ordering) *Let $\mathcal{K}$ be a knowledge base in argumentation system $\mathcal{AS}$. An* argument ordering for $\mathcal{AS}$ w.r.t. knowledge base $\mathcal{K}$ *is a partial preorder $\leq$ on* $\mathsf{Args}(\mathcal{AS})$ *such that for all $A, B \in \mathsf{Args}(\mathcal{AS})$:*

- *if A is firm and strict and B is defeasible or plausible, then $B < A$*
- *if* $\mathsf{lastRule}(A)$ *is strict, then for all $A' \in \mathsf{dirSub}(A)$ it is true that $A \leq A'$ and there is an $A' \in \mathsf{dirSub}(A)$ such that $A' \leq A$.*

Amgoud et al. (2006) propose two principles to define argument orderings: the *last link principle* and *weakest link principle*. Both principles are based on an ordering of sets of applied defeasible rules and on an ordering of sets of premises, which we will denote with $\preceq_{\mathcal{R}}$ and $\preceq_{\mathcal{L}}$ respectively. Orderings on sets could be defined in all kinds of ways using the elements of the set. We define the ordering on a set using the ordering on its elements. Namely, if X and Y are sets, then $X \prec Y$ iff $\exists x \in X$ s.t. $\forall y \in Y$ it is true that $x \prec y$.

Argument orderings following the 'last link principle' use the last defeasible rules that an argument has applied to determine the strength between arguments. Recall from Definition 2.5 that the function $\mathsf{lastDef}$ gives the set of last defeasible rules that have been applied in the given argument. The last link principle states that an argument is stronger if the set of last defeasible rules it applies is stronger according to $\preceq_{\mathcal{R}}$. Arguments that are strict and firm are always stronger than arguments that are defeasible or plausible. However, if two arguments do not apply any defeasible rules, then the relative strength of the premises determines the relative strength of the arguments.

Definition 2.13 (Last Link Principle) *Let $\leq$ be an argument ordering over* $\mathsf{Args}(\mathcal{AS})$. Argument ordering $\leq$ follows the last link principle *iff for all $A, B \in \mathsf{Args}(\mathcal{AS})$: it is true that $A < B$ iff either*

- *A is defeasible or plausible and B is firm and strict, or*
- $\mathsf{lastDef}(A) \prec_{\mathcal{R}} \mathsf{lastDef}(B)$*, or*
- $\mathsf{lastDef}(A)$ *and* $\mathsf{lastDef}(B)$ *are empty and* $\mathsf{premises}(A) \prec_{\mathcal{L}} \mathsf{premises}(B)$.

Instead of only considering the last defeasible inferences that have been made, the weakest link principle considers all the defeasible rules that an argument uses. We will use $\mathsf{defRules}(A)$ to abbreviate $\mathsf{rules}(A) \cap \mathcal{DR}$, which denotes the set of defeasible rules that have been applied in argument A. The weakest link principle states that argument B is stronger than A iff B's premises are stronger than A's premises and if B applies defeasible rules, then the defeasible rules that B applies are stronger than the defeasible rules that A applies.

Definition 2.14 (Weakest Link Principle) *Let $\leq$ be an argument ordering over* $\mathsf{Args}(\mathcal{AS})$. Argument ordering $\leq$ follows the weakest link principle *iff $A < B$ iff*

- *A is defeasible or plausible and B is firm and strict, or*

- $\mathsf{premises}(A) \prec_{\mathcal{L}} \mathsf{premises}(B)$ *and furthermore if* $\mathsf{defRules}(B) \neq \emptyset$*, then* $\mathsf{defRules}(A) \prec_{\mathcal{R}} \mathsf{defRules}(B)$

We will now define *argumentation theories* which combine an argumentation system, knowledge base and argument ordering. Argumentation theories will be used to construct a set of arguments and attacks between those arguments.

Definition 2.15 (Argumentation Theory) *An* argumentation theory *is a tuple* $\langle \mathcal{AS}, \mathcal{K}, \leq \rangle$*, where* $\mathcal{AS}$ *is an argumentation system,* $\mathcal{K}$ *a knowledge base in* $\mathcal{AS}$*, and* $\leq$ *an argument ordering for* $\mathcal{AS}$ *w.r.t.* $\mathcal{K}$*.*

If A is an argument in $\mathcal{AS}$ and all premises of A are in the knowledge base $\mathcal{K}$, then A can be constructed from an argumentation theory $\langle \mathcal{AS}, \mathcal{K}, \leq \rangle$. We will use $\mathsf{Args}(\mathcal{AT})$ to denote the set of arguments that can be constructed from an argumentation theory.

Since arguments can differ in strength, not all attacks are successful. Namely, if an argument rebuts another argument that is stronger, then the attack fails. Otherwise, the attack is successful and becomes a defeat. Undercutting, contrary-rebutting, and contrary-undermining are all asymmetrical attack.

Definition 2.16 (Defeat) *Let* $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ *be an argumentation theory and* A, B *arguments in* $\mathsf{Args}(\mathcal{AT})$*.* Argument A defeats B in $\mathcal{AT}$ *iff*

- A *undercuts, contrary-rebuts, or contrary-undermines* B*, or*
- A *rebuts* B *on* B' *and* $A < B'$ *is not true, or*
- A *undermines* B *on* B' *and* $A < B'$ *is not true.*

The success of undercutting, contrary-rebutting, and contrary-undermining does not depend on preferences between arguments. Therefore, when we say *preference-independent attacks*, we refer to undercutting, contrary-rebutting, and contrary-undermining. In contrast, *preference-dependent attacks* refer to rebutting and undermining.

2.1.4 Argumentation Frameworks

Given an argumentation theory, we can construct a set of arguments and the defeat relation between those arguments. To determine which of those arguments are justified, defensible and overruled, Dung argumentation frameworks will be used (Dung, 1995).

Definition 2.17 (Dung Argumentation Framework) *A* Dung argumentation framework *(AF) is a tuple* $\langle \mathsf{Args}, \mathsf{Attack} \rangle$ *with* Args *a set of arguments and* $\mathsf{Attack} \subseteq \mathsf{Args} \times \mathsf{Args}$ *a binary relation.*

Given an argumentation framework, several properties can be defined of sets of arguments in the argumentation framework. Intuitively, a set of arguments D is conflict-free if D does not contain an argument that attacks another argument in D. Furthermore, a set of arguments D defends an argument A if D attacks each argument that attacks A.

Definition 2.18 (Conflict-Free Sets and Defence) *Let* $\langle \mathsf{Args}, \mathsf{Attack} \rangle$ *be an AF with* $D \subseteq \mathsf{Args}$ *a set of arguments.*

- D *is* conflict-free *iff there exist no* $A, B \in D$ *such that* $(A, B) \in \mathsf{Attack}$.
- D defends *an argument* $A \in \mathsf{Args}$ *iff for each argument* $B \in \mathsf{Args}$ *such that* $(B, A) \in \mathsf{Attack}$ *there exists an argument* $C \in D$ *such that* $(C, B) \in \mathsf{Attack}$.

Definition 2.19 (Acceptability Semantics) *Let* $\langle \mathsf{Args}, \mathsf{Attack} \rangle$ *be an AF with* $D \subseteq \mathsf{Args}$ *a conflict-free set of arguments. Let* $F : 2^{\mathsf{Args}} \rightarrow 2^{\mathsf{Args}}$ *be a function such that* $F(D) = \{A \in \mathsf{Args} \mid D \text{ defends } A\}$.

- D *is* admissible *iff* $D \subseteq F(D)$.
- D *is a* complete extension *iff* $D = F(D)$.
- D *is a* grounded extension *iff it is the smallest complete extension with respect to set-inclusion.*
- D *is a* preferred extension *iff it is a maximal complete extension with respect to set-inclusion.*
- D *is a* stable extension *iff it is a preferred extension that defeats all arguments in* $\mathsf{Args} \setminus D$.

Note that this implies that each grounded, preferred or stable extension of an AF is also a complete extension of that AF. Each AF has exactly one unique grounded extension, whereas all other semantics allow for multiple extensions. The grounded extension of an AF is contained in all other extensions of that AF. Furthermore, each AF has at least one preferred and complete extension. There are AFs without stable extensions.

Definition 2.20 An argument $A \in \mathsf{Args}$ is on the basis of $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$*:*

- justified *iff* A *is in the grounded extension,*
- defensible *iff* A *is in a preferred extension and not in the grounded extension,*
- overruled *iff* A *is not in any extension*

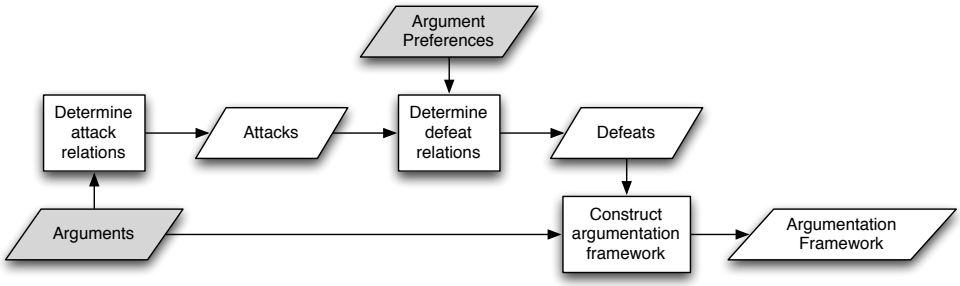
We will now show how a Dung argumentation framework can be constructed from an argumentation theory.

Definition 2.21 (Concrete Argumentation Framework) An argumentation framework on the basis of argumentation theory $\mathcal{AT}$ *is a Dung argumentation framework* $\langle \mathsf{Args}, \mathsf{Attack} \rangle$ *with* Args *the set of arguments that can be constructed from* $\mathcal{AT}$ *and* Attack *the defeat relation on* Args *as in Definition 2.16.*

Figure 2.2 contains a flow chart of how argumentation frameworks are constructed. From the set of arguments that can be constructed from an argumentation theory, the set of attack relations can be determined using Definition 2.11. Given an argument ordering, the attack relation is used to determine the defeat relation between arguments in S following Definition 2.16. Finally, the argumentation framework is constructed from the set of arguments and the defeat relation on them.

Definition 2.20 can now be used to define what status a formula gets in the argumentation framework of an argumentation theory.

Figure 2.2: *Flow Chart of the Construction of an Argumentation Framework.*



Definition 2.22 *Let $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ be an argumentation theory with $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ an argumentation system and let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$ be the argumentation framework on the basis of $\mathcal{AT}$.* Formula $\phi \in \mathcal{L}$ is on the basis of $\mathcal{AT}$*:*

- justified *iff there is a justified argument A such that* $\mathsf{conc}(A) = \phi$*,*
- defensible *iff there is a defensible argument A such that* $\mathsf{conc}(A) = \phi$ *and there is no justified argument B such that* $\mathsf{conc}(B) = \phi$*,*
- overruled *iff there is an overruled argument A such that* $\mathsf{conc}(A) = \phi$ *and there is no justified argument B such that* $\mathsf{conc}(B) = \phi$*,, and*
- rejected *iff there is no argument $A \in \mathsf{Args}$ such that* $\mathsf{conc}(A) = \phi$*.*

2.1.5 Rationality of Conclusions

Caminada and Amgoud (2007) identify several problems that occur in argumentation systems with strict and defeasible inference rules where these systems have unintuitive results. To judge the quality of a rule-based argumentation system, a number of 'rationality postulates' are proposed. Four of these postulates constrain the extensions of an argumentation framework corresponding to an argumentation theory as follows:

1. closure under subarguments, i.e., if A is an argument in an extension, then every subargument of A is also in the extension;
2. closure under strict rules: all the conclusions of arguments in an extension are closed under strict-rule application;
3. direct consistency, i.e., the set of conclusions of arguments in an extension is consistent; and,
4. indirect consistency, i.e., the closure of the set of all conclusions of arguments in an extension under strict-rule application is consistent.

To ensure that the outcome of our argumentation system is intuitive, it is thus important that these postulates are satisfied. For example, it is undesirable that the outcome of arguing about what to do is that the user should choose two alternatives that cannot both be performed. Prakken (2010) proves that the first two postulates are satisfied unconditionally in ASPIC+.

To prove that ASPIC+ satisfies the remaining two rationality postulates under certain conditions, the notions of transposition of a strict rule and closure of sets of strict rules under transpositions are required. These notions are proposed by Caminada and Amgoud (2007). Recall from Subsection 2.1.1 that if $\phi = -\psi$, then ϕ and ψ are contradictory, i.e., it is true that $\phi \in \mathsf{cf}(\psi)$ and $\psi \in \mathsf{cf}(\phi)$.

Definition 2.23 (Transposition) *Strict rule s' is a* transposition *of strict rule s iff s and s' are as follows:*

$$s : \phi_1, \ldots, \phi_m \rightarrow \psi$$
$$s' : \phi_1, \ldots, \phi_{i-1}, -\psi, \phi_{i+1}, \ldots, \phi_m \rightarrow -\phi_i$$

Given the notion of transposition, we will now define a transposition operator that is used to define when a set of strict rules is closed under transposition.

Definition 2.24 (Transposition Operator) *Let $\mathcal{SR}$ be a set of strict rules. $Cl_{tp}(\mathcal{SR})$ is the smallest set such that (1) $\mathcal{SR} \subseteq Cl_{tp}(\mathcal{SR})$; and, (2) if $s \in Cl_{tp}(\mathcal{SR})$ and t is a transposition of s, then $t \in Cl_{tp}(\mathcal{SR})$. We say that $\mathcal{SR}$ is* closed under transposition *iff $Cl_{tp}(\mathcal{SR}) = \mathcal{SR}$.*

An argumentation system $\langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ is called *closed under transposition* iff $\mathcal{SR}$ is closed under transposition. Similarly, an argumentation theory $\langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ is called *closed under transposition* iff $\mathcal{AS}$ is closed under transposition.

Definition 2.25 (Closure of a set of formulae) *Let $\mathcal{L}$ be a logical language, X a set of formulae in $\mathcal{L}$ and $\mathcal{SR}$ a set of strict rules. The* closure *of X under $\mathcal{SR}$, denoted $Cl_{\mathcal{SR}}(X)$, is the smallest set such that:*

- *$X \subseteq Cl_{\mathcal{SR}}(X)$; and,*
- *if $s : \phi_1, \ldots, \phi_m \rightarrow \psi$ in $\mathcal{SR}$ and $\phi_1, \ldots, \phi_n \in Cl_{\mathcal{SR}}(X)$, then $\psi \in Cl_{\mathcal{SR}}(X)$.*

If $X = Cl_{\mathcal{SR}}(X)$, then X is called closed*.*

If $\langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ is an argumentation system and $X \subseteq \mathcal{L}$ a set of formulae, then we will write $X \vdash \phi$ iff there exists a strict argument concluding ϕ of which the premises are in X.

Definition 2.26 (Closure under contraposition) *An argumentation system $\langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ is* closed under contraposition *iff for all $X \subseteq \mathcal{L}$, $\phi \in X$ and $\psi \in \mathcal{L}$: if $S \vdash \psi$, then $S \setminus \{\phi\} \cup \{-\psi\} \vdash -\phi$.*

The postulates of direct and indirect consistency only hold in a special class of argumentation theories, those that are called 'well-formed'. Assume that we have two formulae ϕ and ψ such that $\phi \in \mathsf{cf}(\psi)$, but $\psi \notin \mathsf{cf}(\phi)$. If ψ is a necessary premise, then an argument A concluding ϕ does not undermine an argument B with premise ψ. Consequently, A and B do not attack each other so it is possible that both A and B are in the same extension. Moreover, if there is a strict rule s with ψ as consequent, then it is possible to construct a strict and firm argument A that concludes ψ by applying s and that a defeasible or plausible argument B concludes ϕ. In that case, B rebuts A, but because A is strict and firm and B is defeasible or plausible, argument B does not defeat A. Consequently, it is possible that both A and B are in the same extension.

Definition 2.27 (Well-Formed Argumentation Theory) Argumentation theory $\mathcal{AT}$ is well-formed[3] *iff for all formulae* ψ *it is true that if there is a* ϕ *such that* $\phi \in \mathsf{cf}(\psi)$ *but not* $\psi \in \mathsf{cf}(\phi)$*, then* $\psi \notin \mathcal{K}_{\mathsf{np}}$ *and* ψ *is not the consequent of a strict rule.*

Recall from Definition 2.3 that a set of formulae is called *consistent* iff it does not contain formulae ϕ and ψ s.t. $\phi \in \mathsf{cf}(\psi)$. Prakken (2010) proves that all rationality postulates are satisfied in the ASPIC+ framework under the following conditions. Let $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ be an argumentation theory with $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ and $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$. The four rationality postulates hold for every extension E of the AF corresponding to $\mathcal{AT}$ under a given semantics subsumed by complete semantics if

- $\mathcal{AT}$ is well-formed as in Definition 2.27,
- $\mathcal{SR}$ is closed under transposition or closed under contraposition,
- $\leq$ satisfies the last-link or weakest-link principle, and
- the closure of $\mathcal{K}_{\mathsf{np}}$ under strict-rule application is consistent.

Because we will introduce a new class of argumentation systems and theories, we will show that they satisfy these conditions. This ensures that the conclusions corresponding to an argumentation theory satisfy the rationality postulates.

2.2 Accrual

Like in proof theories, argumentation theories allow that multiple arguments can be constructed that have the same conclusion. Intuitively, each argument provides a different reason for why its conclusion is true. These arguments combined provide an even more compelling defeasible reason to believe that their conclusion is true. Accruing arguments means that different arguments with the same conclusion are taken together, i.e., are accrued. This is particularly important in the context of decision making where there may be multiple reasons in favor of and against making a decision. In this section, an accrual mechanism is proposed for the ASPIC+ argumentation framework as proposed in Section 2.1. This accrual mechanism makes the approach of Prakken (2005b) more precise[4].

Prakken formulates the following three accrual principles that he argues should be satisfied in an accrual mechanism.

1. *An accrual can be weaker than its accruing elements*. For example, it being hot is a reason not to jog, that it is raining is a reason not to jog, but together they may form a weaker reason not to jog because the combination may be more pleasant.
2. *An accrual makes its accruing elements inapplicable*. This means that if there are n arguments that can be accrued, then all n should be accrued and not less than n.
3. *Flawed arguments may not accrue*. For example, if it is not true that it is hot, then the argument not to jog because it is hot cannot be accrued.

[3] This definition of well-formed argumentation theories can be found in Modgil and Prakken (2011) and is a correction of the version in Prakken (2010).

[4] As already noted, this section makes a contribution to existing work and does therefore not strictly fit in the background chapter. However, because we want to have the argumentation mechanism in one chapter, we have decided to include this section here.

A possible approach to include an accrual mechanism is to extend the definition of an argument with an accrual step. Since this alters ASPIC+ on a fundamental definition, all properties of ASPIC+ need to be proved again. In contrast, Prakken's formalization of accrual can be added to ASPIC+ without changing ASPIC+. To avoid changing ASPIC+, we will follow Prakken's approach. The key element of his approach is that formulae in the logical language can be labeled or unlabeled. A labeled formula can be thought of as an 'intermediate conclusion', i.e., a conclusion that has not yet been accrued. In contrast, an unlabeled formula can be thought of as a conclusion that has been accrued. Next, defeasible inference rules are adapted such that they only use unlabeled formulae in their antecedent and have a labeled formula in their conclusion. Finally, accrual inference rules are used to accrue labeled formulae to infer an unlabeled formula.

We will now formalize this approach in the ASPIC+ framework. In contrast to Prakken, we will first define the possible labels that formulae can have. A label can be either the rule name of a defeasible inference rule or the special label kb denoting that the formula is in the knowledge base. Note that Prakken does not consider the label kb. Because the knowledge base can contain premises that are not certain, it is useful to be able to label formulae in the knowledge base with kb so that they can be accrued with arguments that justify a formula in a different way.

Definition 2.28 (Accrual Labels) *Let $\mathcal{R} = \mathcal{SR} \cup \mathcal{DR}$ be a set of inference rules. The* accrual labels for $\mathcal{R}$ *is a set L such that*

- *if $d : \phi_1, \ldots, \phi_m \Rightarrow \phi$ is in $\mathcal{DR}$, then d is a label in L*
- kb *is a label in L*

Nothing else is an accrual label.

The logical language is extended with labeled formula. Each formula can be labeled with an accrual label.

Definition 2.29 (Labeled Language) *Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ be an argumentation system. A* labeled language $\mathcal{L}^{\mathtt{lbl}}$ for $\mathcal{AS}$ *is a first-order language with*

- *if ϕ is a formula of $\mathcal{L}$, then ϕ is a formula of $\mathcal{L}^{\mathtt{lbl}}$, and*
- *if ϕ is a formula of $\mathcal{L}$ and l an accrual label for $\mathcal{R}$, then ϕ^l is a formula of $\mathcal{L}^{\mathtt{lbl}}$*

For labeled languages we consider a special class of inference rules that we will call *labeled inference rules*. A labeled inference rule differs from a normal inference rule in that its conclusion must be labeled and that its antecedent can only contain unlabeled formulae. Following Prakken, only defeasible rules are labeled, but in contrast to Prakken, the conclusions of defeasible rules are labeled with the rule name instead of its antecedent. Because every defeasible rule has a different rule name, the label exactly identifies the rule that has been applied.

Definition 2.30 (Labeled Defeasible Inference Rules) *Let $\mathcal{L}^{\mathtt{lbl}}$ be a labeled language w.r.t. some argumentation system.* Labeled defeasible inference rules *w.r.t. $\mathcal{L}^{\mathtt{lbl}}$ are of the form:*

$$d : \phi_1, \ldots, \phi_m \Rightarrow \phi^d$$

with $\phi_1, \ldots, \phi_m$ unlabeled formulae (with $m \geq 0$).

Inference rules that do not contain any labeled formulae in their antecedent or conclusion are called *unlabeled*. Note that it is straightforward to transform an unlabeled inference rule into a labeled inference rule by labeling its conclusion with the rule name, i.e., the labeled version of $d : \phi_1, \ldots, \phi_m \Rightarrow \phi$ is $d : \phi_1, \ldots, \phi_m \Rightarrow \phi^d$.

To accrue labeled formulae, we will follow Prakken by introducing the following set of *accrual rules*. The set of accrual rules consists of accrual inferences and accrual undercutters. Labeled formulae are seen as intermediate conclusions and unlabeled formulae are seen as normal conclusions. An *accrual inference* infers a 'normal conclusion' from a number of 'intermediate conclusions'. More specifically, an accrual inference rule for formula ϕ takes a number of differently labeled instances of ϕ (e.g., ϕ^{d_1} and ϕ^{d_2}) and defeasibly infers that ϕ. Accrual undercutters ensure that if an accrual is applicable, then it makes its elements inapplicable. For each accrual inference, there are a number of accrual undercutters. An accrual undercuts all accruals of its elements.

Definition 2.31 (Accrual Rules) *Let $\mathcal{DR}^{\mathtt{lbl}}$ be a set of labeled defeasible inference rules. The* accrual rules for $\mathcal{DR}^{\mathtt{lbl}}$ *is the set $\mathcal{AR}$ such that*

- $d_\phi : \phi^{l_1}, \ldots, \phi^{l_i} \Rightarrow \phi$ *in* $\mathcal{AR}$
- $d_{uc} : \phi^{l_1}, \ldots, \phi^{l_i} \Rightarrow \neg\mathsf{appl}(d_\phi)$ *in $\mathcal{AR}$ for any accrual rule* $d_\phi : \phi^{m_1}, \ldots, \phi^{m_j} \Rightarrow \phi$ *in $\mathcal{AR}$ s.t.* $\{m_1, \ldots, m_j\} \subset \overline{\{l_1, \ldots, l_i\}}$.

Note that accrual rules are normal defeasible inference rules. Further note that the accrual undercutter rules ensure that the second accrual principle is satisfied. We are now ready to define an *argumentation system for accrual*.

Definition 2.32 (Argumentation System for Accrual) *An* argumentation system for accrual *is an argumentation system $\langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}^{\mathtt{lbl}} \cup \mathcal{AR}, \mathsf{cf} \rangle$ with*

- *$\mathcal{L}$ is a first-order language with labeled formulae,*
- *$\mathcal{SR}$ is a set of unlabeled strict inference rules,*
- *$\mathcal{DR}^{\mathtt{lbl}}$ a set of labeled defeasible inference rules, and*
- *$\mathcal{AR}$ the set of accrual rules for $\mathcal{DR}^{\mathtt{lbl}}$.*

Note that argumentation systems for accrual are a class of argumentation systems, which means that arguments can be constructed as usual.

Example 2.2 (Accrual) Let $\mathcal{R} = \mathcal{SR} \cup \mathcal{DR}$ be a set of inference rules with $\mathcal{DR} = \{d_1 : \phi_1 \Rightarrow \psi, d_2 : \phi_2 \Rightarrow \psi\}$ a set of defeasible inference rules. The set of accrual labels for $\mathcal{R}$ is $L = \{\mathsf{kb}, d_1, d_2\}$. Transforming $\mathcal{DR}$ into labeled rules results in $\mathcal{DR}^{\mathtt{lbl}} = \{d_1 : \phi_1 \Rightarrow \psi^{d_1}, d_2 : \phi_2 \Rightarrow \psi^{d_2}\}$. The set $\mathcal{AR}$ of accrual rules for $\mathcal{DR}^{\mathtt{lbl}}$ contains the following accrual inferences:

$$
\begin{aligned}
&\{d_{\phi_1\mathsf{kb}} : \phi_1^{\mathsf{kb}} \Rightarrow \phi_1, \quad d_{\phi_2\mathsf{kb}} : \phi_2^{\mathsf{kb}} \Rightarrow \phi_2, \quad d_{\psi\mathsf{kb}} : \psi^{\mathsf{kb}} \Rightarrow \psi,\\
&d_{\psi 1} : \psi^{d_1} \Rightarrow \psi, \quad d_{\psi 2} : \psi^{d_2} \Rightarrow \psi,\\
&d_{\psi 12} : \psi^{d_1}, \psi^{d_2} \Rightarrow \psi, \quad d_{\psi 1\mathsf{kb}} : \psi^{d_1}, \psi^{\mathsf{kb}} \Rightarrow \psi, \quad d_{\psi 2\mathsf{kb}} : \psi^{d_2}, \psi^{\mathsf{kb}} \Rightarrow \psi,\\
&d_{\psi 12\mathsf{kb}} : \psi^{d_1}, \psi^{d_2}, \psi^{\mathsf{kb}} \Rightarrow \psi, \}
\end{aligned}
$$

The first five accrual inferences concern accruing a single labeled formula. The first three rules are accrual inferences for if the available formulae are in the knowledge base. The next two rules are the accrual inferences for in defeasible rules d_1 and d_2 have been applied. The next three accrual inferences accrue combinations of two labeled formulae. The last accrual inference accrues all the labeled versions of ψ. Note that $\mathcal{AR}$ also contains accrual undercutter rules.

Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}^{\mathtt{lbl}} \cup \mathcal{AR}, \mathsf{cf} \rangle$ be the corresponding argumentation system for accrual. Some arguments in $\mathcal{AS}$ are as follows.

$$\frac{\dfrac{\phi_1}{\psi^{d_1}}\ d_1}{\phi}\ d_{\phi 1} \qquad \frac{\dfrac{\phi_2}{\psi^{d_2}}\ d_2}{\psi}\ d_{\psi 2} \qquad \frac{\dfrac{\phi_1}{\psi^{d_1}}\ d_1 \quad \dfrac{\phi_2}{\psi^{d_2}}\ d_2}{\psi}\ d_{\psi 12}$$

Argumentation systems for accrual satisfy the three accrual principles that were proposed by Prakken (2005b). Firstly, because we give no restriction on the relative strength of accrual arguments it is possible to model that an accrual is weaker than its accruing elements. Secondly, as already noted, by including accrual undercutters in the argumentation system, an accrual makes its accruing elements inapplicable. Finally, if an argument A is flawed, i.e., attacked successfully, then an accrual argument that accrues A is also attacked successfully because A is a subargument. Consequently, flawed arguments do not accrue.

We will now show how an argumentation system with no accrual can be transformed into an argumentation system for accrual.

Definition 2.33 (Accrual Transformation) *Let $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ be an argumentation system with no accrual. The* accrual transformation *of $\mathcal{AS}$ is an argumentation system for accrual $\langle \mathcal{L}^{\mathtt{lbl}}, \mathcal{SR} \cup \mathcal{DR}^{\mathtt{lbl}} \cup \mathcal{AR}, \mathsf{cf}' \rangle$ with*

- *$\mathcal{L}^{\mathtt{lbl}}$ the labeled language for $\mathcal{AS}$,*
- *$\mathcal{DR}^{\mathtt{lbl}}$ the labeled versions of $\mathcal{DR}$,*
- *$\mathcal{AR}$ the accrual rules for $\mathcal{DR}^{\mathtt{lbl}}$,*
- *cf' a contrariness function such that if $\phi \in \mathsf{cf}(\psi)$, then $\phi \in \mathsf{cf}'(\psi)$.*

Arguments will typically be constructed from a knowledge base. The elements in a knowledge base are called premises. Some premises are necessarily true and others may be true. The set $\mathcal{K}_{\mathsf{np}}$ denotes the set of *necessary premises*, which, intuitively, cannot be attacked. The set $\mathcal{K}_{\mathsf{np}}$ should contain the axioms for an argumentation system. The set $\mathcal{K}_{\mathsf{op}}$ denotes the set of *ordinary premises*, which can be attacked.

If ϕ is an ordinary premise, then ϕ may not be true. Because it is possible that ϕ can be inferred from other premises, it is useful to be able to accrue different reasons for why ϕ is true and ϕ being an ordinary premise is one of such reasons. Accruing can only be done with labeled formulae. To be able to accrue ordinary premises, ordinary premises are labeled with the special label kb. Note that because necessary premises are always true, they do not need to be accrued and thus they do not need to be labeled.

Definition 2.34 (Knowledge Base for Accrual) *Let $\mathcal{L}$ be a language with labeled formulae. A* knowledge base for $\mathcal{L}$ *is a knowledge base $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ of formulae in $\mathcal{L}$ such that $\mathcal{K}_{\mathsf{np}}$ is a set of unlabeled formulae, $\mathcal{K}_{\mathsf{op}}$ and $\mathcal{K}_{\mathsf{as}}$ are sets of labeled formulae, and $\mathcal{K}_{\mathsf{np}}$, $\mathcal{K}_{\mathsf{op}}$ and $\mathcal{K}_{\mathsf{as}}$ are disjoint.*

Note that Prakken does not label formulae in the knowledge base. If a formula ϕ is an ordinary premise, then that is a reason to believe that ϕ is true, but it is not certain knowledge. If there are also a number of arguments that conclude ϕ from other premises, then you should accrue these different reasons to believe that ϕ is true. By labeling premises in the knowledge base with a special label, it is possible to accrue these atomic arguments with other arguments.

Example 2.3 (Accruals and Knowledge Bases) Let the argumentation system for accrual $\mathcal{AS}$ be as in Example 2.2 and let $\mathcal{K} = \langle \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{as}} \rangle$ be a knowledge base for accrual with $\mathcal{K}_{\mathsf{np}} = \emptyset$, $\mathcal{K}_{\mathsf{op}} = \{\phi_1^{\mathsf{kb}}, \phi^{\mathsf{kb}}\}$ and $\mathcal{K}_{\mathsf{as}} = \emptyset$. From $\mathcal{K}$, the following arguments can be constructed.

$$A_1 = \frac{\phi_1^{\mathsf{kb}}}{\phi_1}\ d_{\phi_1 \mathsf{kb}} \qquad A_2 = \frac{A_1}{\psi^{d_1}}\ d_1 \qquad A_3 = \frac{A_2}{\phi}\ d_{\phi 1}$$

Argument A_1 applies the accrual inference on the labeled formula ϕ_1^{kb} that is in the knowledge base and A_1 concludes the unlabeled formula ϕ_1. Argument A_2 applies defeasible rule d_1 on the conclusion of A_1 to conclude the labeled formula ψ^{d_1}. Finally, A_3 concludes that ψ by accruing the conclusion of A_2. Because the knowledge base also contains ψ^{kb}, the following arguments can also be constructed.

$$A_4 = \frac{\psi^{\mathsf{kb}}}{\psi}\ d_{\psi \mathsf{kb}} \qquad A_5 = \frac{\psi^{\mathsf{kb}} \quad A_2}{\psi}\ d_{\psi \mathsf{kb} 1}$$

Argument A_4 accrues only ϕ^{kb}, whereas A_5 accrues both ϕ^{kb} and ϕ^{d_1} (the conclusion of A_2). Finally, the following arguments can be constructed that apply the accrual undercutter rule.

$$A_6 = \frac{\psi^{\mathsf{kb}} \quad A_2}{\neg \mathsf{appl}(d_{\psi 1})}\ d_{uc \mathsf{kb} 1} \qquad A_7 = \frac{\psi^{\mathsf{kb}} \quad A_2}{\neg \mathsf{appl}(d_{\psi \mathsf{kb}})}\ d_{uc 1 \mathsf{kb}}$$

Argument A_6 concludes that the accrual inference on just d_1 cannot be applied because both kb and d_1 can be accrued. For the same reason, A_7 concludes that not just kb should be accrued. This means that A_6 undercuts A_3 and that A_7 undercuts A_4.

Just like an argumentation system with no accrual can be transformed into an argumentation system for accrual, an (non-accrual) argumentation theory can be transformed into an accrual argumentation theory as follows.

Definition 2.35 (Accrual Theory Transformation) *Let $\mathcal{AT} = \langle \mathcal{AS}, \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle, \leq \rangle$ be an argumentation theory with $\mathcal{AS}$ an argumentation system with no accrual. The* accrual theory transformation of $\mathcal{AT}$ *is an argumentation theory $\langle \mathcal{AS}', \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}^l_{\mathsf{op}}, \mathcal{K}^l_{\mathsf{as}} \rangle, \leq' \rangle$ with*

- *$\mathcal{AS}'$ the accrual transformation of $\mathcal{AS}$,*
- *if $\phi \in \mathcal{K}_{\mathsf{op}}$, then $\phi^{\mathsf{kb}} \in \mathcal{K}^l_{\mathsf{op}}$,*
- *if $\phi \in \mathcal{K}_{\mathsf{as}}$, then $\phi^{\mathsf{kb}} \in \mathcal{K}^l_{\mathsf{as}}$, and*
- *$\leq'$ is an argument ordering over $\mathsf{Args}(\mathcal{AS}')$.*

Because accrual argumentation theories are a class of argumentation theories, it is straightforward to construct the argumentation framework corresponding to an accrual argumentation theory.

2.3 Meta-Level Argumentation

In this section, meta-level argumentation is introduced to reason about the relative strengths of object-level arguments. Reasoning about the relative strength of arguments is reasoning *about* arguments and is therefore on a meta-level with respect to those arguments. In this section we propose a meta-level argumentation framework to reason about the relative strengths of 'object-level' arguments. By using a meta-level argumentation framework, we avoid that the logical language needs to be self-referential.

In (Genesereth and Nilsson, 1987, Chapter 10), a first-order meta-logic is described as a first-order logic of which the terms include the sentences of another language (the object language). Two kinds of meta-languages are distinguished: those that refer to themselves, i.e., self-referential languages, and those that do not. A self-referential meta-language tends to be rather complex and intricate because axioms tend to become inconsistent and paradoxical statements such as the liar paradox can be expressed (Perlis (1985); Turner (1990)). First-order hierarchical meta-languages are believed to provide a more stable logical foundation (Turner (1990)). In Wooldridge et al. (2005), first-order hierarchical meta-languages are used for argumentation. This approach is extended in Hunter (2008) to reason about whether a particular argument is appropriate for a particular agent.

2.3.1 Meta-Argumentation System

We will now introduce a meta-language that does not refer to itself, but which does refer to elements of an argumentation system, i.e., an argumentation system's arguments, formulae, and inference rules. The language is thus on a meta-level with respect to one particular argumentation system.

Definition 2.36 (Meta-Language) *Let $\mathcal{AS} = (\mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf})$ be an argumentation system and let $\mathcal{R} = \mathcal{SR} \cup \mathcal{DR}$. A* Meta-Language w.r.t. *$\mathcal{AS}$ is a first-order language $\mathcal{L}'$ such that $\mathcal{L} \neq \mathcal{L}'$ and with at least:*

- *constants:*
  - *for each formula ϕ in $\mathcal{L}$, there is a constant $\underline{\phi}$,*
  - *for each inference rule r in $\mathcal{R}$, there is a constant $\underline{r}$,*
  - *for each argument A in $\mathsf{Args}(\mathcal{AS})$ there is a constant $\underline{A}$,*
  - *for each subset X of $\mathcal{L}$, $\mathcal{R}$ or $\mathsf{Args}(\mathcal{AS})$ there is a constant $\underline{X}$*
- *functions:*
  - *for each function on arguments defined in Definition 2.5 there is a function symbol, and*
  - *there is an unary function symbol for* cf*,*
- *predicates:*
  - *unary predicates:* np*,* op *and* as
  - *binary predicates:* in*, $\preceq_{\mathsf{Args}}$, $\preceq_{\mathcal{R}}$ and $\preceq_{\mathcal{L}}$*

The unary predicates "np", "op" and "as" denote that the given formula is a necessary premise, ordinary premise and assumption respectively in the object-level knowledge base. The binary predicate $\preceq_{\mathsf{Args}}(\underline{x}, \underline{y})$ is also written as $\underline{x} \preceq_{\mathsf{Args}} \underline{y}$ and denotes that $\underline{y}$ is as strong as or stronger than $\underline{x}$. For example, if $\underline{A}$ and $\underline{B}$ denote object-level arguments, then $\underline{A} \preceq_{\mathsf{Args}} \underline{B}$ denotes that $\underline{B}$ is at least as strong as $\underline{A}$. The predicates $\preceq_{\mathcal{R}}$ and $\preceq_{\mathcal{L}}$ are used similarly and denote relative strength of sets of object-level defeasible rules and sets of object-level formulae.

For ease of notation, if $\underline{x}$ is a constant, then we will write simply x. For example, instead of $\mathsf{in}(\underline{\phi}, \underline{\mathcal{L}})$, we will write $\mathsf{in}(\phi, \mathcal{L})$. Furthermore, the constants $\underline{\mathcal{L}}$, $\underline{\mathsf{Args}(\mathcal{AS})}$, $\underline{\mathcal{SR}}$ and $\underline{\mathcal{DR}}$ are used to denote the sets of object-level formulae, object-level arguments, strict inference rules and defeasible inference rules respectively. To further ease the notation, the abbreviations in Table 2.3 are used.

Table 2.3: *Abbreviations in Meta-Languages*

Abbreviation	Of
$p \prec_X q$	$p \preceq_X q \wedge \neg(q \preceq_X p)$ (X either Args, $\mathcal{R}$ or $\mathcal{L}$)
$\exists_{x \in X}[\phi]$	$\exists_x[\mathsf{in}(x, X) \wedge \phi]$
$\forall_{x \in X}[\phi]$	$\forall_x[\mathsf{in}(x, X) \supset \phi]$
$\exists_{x_1,\ldots,x_n \in X}[\phi]$	$\exists_{x_1,\ldots,x_n}[\mathsf{in}(x_1, X) \wedge \ldots \wedge \mathsf{in}(x_n, X) \wedge \phi]$
$\forall_{x_1,\ldots,x_n \in X}[\phi]$	$\forall_{x_1,\ldots,x_n}[\mathsf{in}(x_1, X) \wedge \ldots \wedge \mathsf{in}(x_n, X) \supset \phi]$
$X \subseteq X'$	$\forall_{x \in X}[\mathsf{in}(x, X')]$
$X \subset X'$	$X \subseteq X' \wedge \neg(X' \subseteq X)$
$\mathsf{strict}(A)$	$\neg\exists_{r \in \mathsf{rules}(A)}[\mathsf{in}(r, \mathcal{DR})]$
$\mathsf{firm}(A)$	$\forall_{\phi \in \mathsf{premises}(A)}[\mathsf{np}(\phi)]$

To construct an argumentation system with a meta-language we need to define a contrariness function. To ensure that argumentation theories are well-formed, we will restrict the contrariness function to symmetrical conflict, i.e., there is no formula that has a contrary.

Definition 2.37 (Contrariness Function for Meta-Languages) *Let $\mathcal{L}$ be a meta-language for argumentation system $\mathcal{AS}$. A* contrariness function for meta-language $\mathcal{L}$ *is a contrariness function* $\mathsf{cf} : \mathcal{L} \to 2^{\mathcal{L}}$ *with at least*

- $x \prec y \in \mathsf{cf}(y \prec x)$*, and*
- *there are no formulae ϕ and ψ s.t. $\phi \in \mathsf{cf}(\psi)$ and $\psi \notin \mathsf{cf}(\phi)$.*

To construct arguments that are on a meta-level with respect to arguments of some argumentation system $\mathcal{AS}$, we will define *meta-argumentation systems* that contains a meta-language and a contrariness function for meta-languages.

Definition 2.38 (Meta-Argumentation System) *Let $\mathcal{AS}$ be an argumentation system. A* Meta-Argumentation System w.r.t. $\mathcal{AS}$ *is an argumentation system $\langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ such that*

- *$\mathcal{L}$ is a meta-language for $\mathcal{AS}$,*

- *$\mathcal{SR}$ is the set of all valid first-order inferences*[5]*,*
- *$\mathcal{DR}$ is a set of defeasible inference rules, and*
- cf *is a contrariness function for $\mathcal{L}$.*

Meta-argumentation systems are a class of argumentation systems where the logical language and contrariness are instantiated as stated. Therefore, arguments are constructed as in Definition 2.5 and the definitions of attack, an argumentation theory and of a Dung argumentation framework can also be used with meta-argumentation systems.

2.3.2 Meta-Argumentation Theories

Because meta-argumentation systems are a class of argumentation systems, it is straightforward to construct an argumentation theory (see Definition 2.15) for a meta-argumentation system. An argumentation theory is a tuple consisting of an argumentation system $\mathcal{AS}$, a knowledge base in $\mathcal{AS}$ and an argument ordering over arguments in $\mathcal{AS}$.

In the knowledge base of each argumentation theory we will put several axioms to ensure some desirable properties. Meta-argumentation systems have a predicate $\preceq$ that is used to denote the relative strength between object-level arguments. To ensure that $\preceq$ is an argument ordering, it is necessary to enforce several properties on $\preceq$ such as reflexivity and transitivity. Table 2.4 describes the axioms that we will use in meta-argumentation theories. Axioms $\mathtt{trns}'_{\mathsf{Args}}$, $\mathtt{trns}'_{\mathcal{R}}$ and $\mathtt{trns}'_{\mathcal{L}}$ ensure that $\preceq_{\mathsf{Args}}$, $\preceq_{\mathcal{R}}$ and $\preceq_{\mathcal{L}}$ respectively are transitive. Furthermore, axioms $\mathtt{rflx}'_{\mathsf{Args}}$, $\mathtt{rflx}'_{\mathcal{R}}$ and $\mathtt{rflx}'_{\mathcal{L}}$ ensure that $\preceq_{\mathsf{Args}}$, $\preceq_{\mathcal{R}}$ and $\preceq_{\mathcal{L}}$ respectively are reflexive.

Next, axioms $\mathtt{adm}'_1$ and $\mathtt{adm}'_2$ ensure that $\preceq_{\mathsf{Args}}$ is an admissible ordering over arguments. More specifically, $\mathtt{adm}'_1$ formalizes that arguments that are strict and firm are stronger than arguments that are defeasible or plausible, and $\mathtt{adm}'_2$ formalizes that if the last inference in argument A is strict, then A is as strong as the weakest direct subargument of A. Axioms $\mathtt{lstLnk}'_1$ and $\mathtt{lstLnk}'_2$ formalize the last link principle[6] as described in Definition 2.13. The last link principle states that argument A is stronger than argument B if the set of last applied defeasible rules of A is stronger than the set of last applied defeasible rules of B. If A and B do not apply any defeasible inference rules, then A is stronger than B if the set of A's premises is stronger than the set of B's premises.

The last link principle as in Definition 2.13 requires that sets of defeasible rules can be compared and that sets of premises can be compared. Axioms $\mathtt{set}'_{\mathcal{R}}$ and $\mathtt{set}'_{\mathcal{L}}$ formalize the approach of Prakken (2010) to compare sets, but there are other approaches[7] that can be chosen. Set Y is stronger than set X if there is a x in X that is weaker than every y in Y. This axiom can be used for sets of inference rules but also for sets of premises. Finally, axioms $\mathtt{prem}'_1$ and $\mathtt{prem}'_2$ concern the relative strength of sets of single premises. Namely, $\mathtt{prem}'_1$ states that if x is an ordinary premise and y is a necessary premise, then y is a stronger than

[5] Note that the set of all valid first-order strict inference rules is closed under contraposition and transposition.

[6] If the weakest link principle should be used instead of the last link principle, then axioms $\mathtt{lstLnk}'_1$ and $\mathtt{lstLnk}'_2$ should be replaced by the axiom $\mathsf{premises}(A) \prec_{\mathcal{L}} \mathsf{premises}(B) \wedge (\mathsf{defRules}(B) \neq \emptyset \supset \mathsf{defRules}(A) \prec_{\mathcal{R}} \mathsf{defRules}(B)) \supset A \prec_{\mathsf{Args}} B$ where $\mathsf{defRules}(A)$ is the function that returns the set of all defeasible rules applied in the given argument.

[7] For example, instead of only looking at the weakest member of a set, only the strongest member could be considered: set Y is stronger than X if there is a y in Y that is stronger than every x in X.

x. Similarly, axiom $\texttt{prem}'_2$ states that if x is an assumption and y is an ordinary premise, then y is stronger than x.

Table 2.4: *Axioms for Meta-Argumentation Theories*

Name	Axiom[a]
$\texttt{trns}'_{\mathsf{Args}}$	$x \preceq_{\mathsf{Args}} y \wedge y \preceq_{\mathsf{Args}} z \supset x \preceq_{\mathsf{Args}} z$
$\texttt{trns}'_{\mathcal{R}}$	$x \preceq_{\mathcal{R}} y \wedge y \preceq_{\mathcal{R}} z \supset x \preceq_{\mathcal{R}} z$
$\texttt{trns}'_{\mathcal{L}}$	$x \preceq_{\mathcal{L}} y \wedge y \preceq_{\mathcal{L}} z \supset x \preceq_{\mathcal{L}} z$
$\texttt{rflx}'_{\mathsf{Args}}$	$\forall_{x \in \mathsf{Args}(\mathcal{AS})}[x \preceq_{\mathsf{Args}} x]$
$\texttt{rflx}'_{\mathcal{R}}$	$\forall_{x \subseteq \mathcal{DR}}[x \preceq_{\mathcal{R}} x]$
$\texttt{rflx}'_{\mathcal{L}}$	$\forall_{x \subseteq \mathcal{L}}[x \preceq_{\mathcal{L}} x]$
$\texttt{adm}'_1$	$\mathsf{strict}(x) \wedge \mathsf{firm}(x) \wedge \neg(\mathsf{strict}(y) \wedge \mathsf{firm}(y)) \supset y \prec_{\mathsf{Args}} x$
$\texttt{adm}'_2$	$\mathsf{in}(\mathsf{lastRule}(x), \mathcal{SR}) \supset \forall_{y \in \mathsf{dirSub}(x)}[x \preceq_{\mathsf{Args}} y] \wedge \exists_{y \in \mathsf{dirSub}(x)}[y \preceq_{\mathsf{Args}} x]$
$\texttt{lstLnk}'_1$	$\mathsf{lastDef}(x) \neq \mathsf{lastDef}(y) \wedge \mathsf{lastDef}(x) \prec_{\mathcal{R}} \mathsf{lastDef}(y) \supset x \prec_{\mathsf{Args}} y$
$\texttt{lstLnk}'_2$	$\mathsf{lastDef}(x) = \emptyset \wedge \mathsf{lastDef}(y) = \emptyset \wedge \mathsf{premises}(x) \prec_{\mathcal{L}} \mathsf{premises}(y) \supset x \prec_{\mathsf{Args}} y$
$\texttt{set}'_{\mathcal{L}}$	$\exists_{x \in X} \forall_{y \in Y}[\{x\} \prec_{\mathcal{L}} \{y\}] \supset X \prec_{\mathcal{L}} Y$
$\texttt{set}'_{\mathcal{R}}$	$\exists_{x \in X} \forall_{y \in Y}[\{x\} \prec_{\mathcal{R}} \{y\}] \supset X \prec_{\mathcal{R}} Y$
$\texttt{prem}'_1$	$\mathsf{op}(x) \wedge \mathsf{np}(y) \supset \{x\} \prec_{\mathcal{L}} \{y\}$
$\texttt{prem}'_2$	$\mathsf{as}(x) \wedge \mathsf{op}(y) \supset \{x\} \prec_{\mathcal{L}} \{y\}$

[a] All formulae with free variables are implicitly universally quantified.

We will now define knowledge bases for meta-argumentation systems. We will call such knowledge bases *meta-knowledge bases*. The necessary premises of meta-knowledge bases should contain the axioms proposed in Table 2.4 and the ordinary premises should contain the following knowledge. A meta-language contains constants denoting sets and constants denoting members of these sets. A meta-language also has a binary predicate in. If $\mathsf{in}(x, X)$ is true, then we say that the denotation of x is a member of the denotation of X. Knowledge bases for meta-argumentation systems must contain formulae stating when the in predicate is true for all constants denoting sets and the members of these sets. For example, if X is a set of formulae on the object-level and $x \in X$, then on its meta-level $\mathsf{in}(\underline{x}, \underline{X})$ is added to the set of ordinary premises.

Furthermore, the meta-language has the unary predicates np, op and as. The expression $\mathsf{np}(\phi)$ denotes that object-level formula ϕ is a necessary premise in the object-level knowledge base. Similarly, $\mathsf{op}(\phi)$ and $\mathsf{as}(\phi)$ denote that ϕ is an ordinary premise and assumption respectively in the object-level knowledge base. A knowledge base for meta-argumentation must also be instantiated with this knowledge.

Definition 2.39 (Meta-Knowledge Base) *Let $\mathcal{AS}'$ be a meta-argumentation system w.r.t. argumentation system $\mathcal{AS}$ and $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ a knowledge base in $\mathcal{AS}$. A* meta-knowledge base in $\mathcal{AS}'$ w.r.t. $\mathcal{K}$ *is a tuple $\langle \mathcal{K}'_{\mathsf{np}}, \mathcal{K}'_{\mathsf{op}}, \mathcal{K}'_{\mathsf{as}} \rangle$ with*

- $\mathcal{K}'_{\mathsf{np}}$ *contains the axioms in Table 2.4,*
- $x \in X$ *iff* $\mathsf{in}(\underline{x}, \underline{X}) \in \mathcal{K}'_{\mathsf{op}}$,
- $\phi \in \mathcal{K}_{\mathsf{op}}$ *iff* $\mathsf{op}(\underline{\phi}) \in \mathcal{K}'_{\mathsf{op}}$,

- $\phi \in \mathcal{K}_{\mathsf{np}}$ *iff* $\mathsf{np}(\underline{\phi}) \in \mathcal{K}'_{\mathsf{op}}$, *and*
- $\phi \in \mathcal{K}_{\mathsf{as}}$ *iff* $\mathsf{as}(\underline{\phi}) \in \mathcal{K}'_{\mathsf{op}}$.

We have now defined all the ingredients necessary for an argumentation theory for meta-argumentation systems. Such an argumentation theory is called a *meta-argumentation theory*.

Definition 2.40 (Meta-Argumentation Theory) *Let $\mathcal{AS}$ be an argumentation system and $\mathcal{K}$ a knowledge base in $\mathcal{AS}$. A* meta-argumentation theory w.r.t. $\mathcal{AS}$ and $\mathcal{K}$ *is an argumentation theory $\langle \mathcal{AS}', \mathcal{K}', \leq \rangle$ with*

- *$\mathcal{AS}'$ a meta-argumentation system w.r.t. $\mathcal{AS}$,*
- *$\mathcal{K}'$ a meta-knowledge base in $\mathcal{AS}'$ w.r.t. $\mathcal{K}$, and*
- *$\leq$ an argument ordering over $\mathsf{Args}(\mathcal{AS}')$ that satisfies the last-link or weakest link principle.*

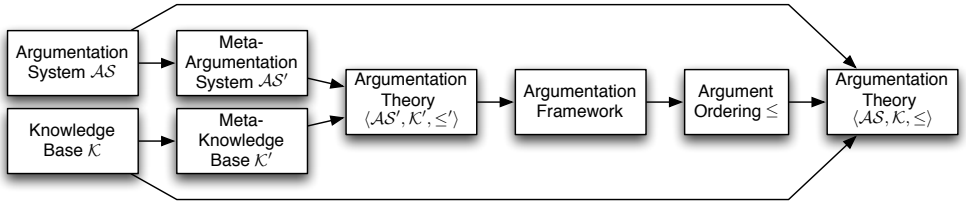
Note that meta-argumentation theories are a class of argumentation theories. Recall that if $\mathcal{AT} = \langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ is an argumentation theory, then $\mathsf{Args}(\mathcal{AT})$ denotes the set of all arguments in $\mathsf{Args}(\mathcal{AS})$ that can be constructed from $\mathcal{AT}$, i.e., $A \in \mathsf{Args}(\mathcal{AT})$ iff $A \in \mathsf{Args}(\mathcal{AS})$ and for all $\phi \in \mathsf{premises}(A)$ it is true that ϕ in $\mathcal{K}$.

2.3.3 Meta-Argumentation Framework

Because a meta-argumentation theory is an argumentation theory, it is straightforward to construct its corresponding argumentation framework. In the meta-language, the predicate $\preceq$ denotes the relative strength of object-level arguments. We will use the status of formulae w.r.t. $\preceq$ in the meta-level argumentation framework to determine the argument ordering of the object-level argumentation theory.

Figure 2.3 sketches how we propose to do this where an arrow from A to B denotes that A is used in or by B. A meta-argumentation system $\mathcal{AS}'$ is constructed on the basis of an object-level argumentation system $\mathcal{AS}$ and a meta-knowledge base $\mathcal{K}'$ is constructed on the basis of object-level knowledge base $\mathcal{K}$. From the argumentation theory consisting of $\mathcal{AS}'$ and $\mathcal{K}'$ an argumentation framework is built. In that AF there are arguments that conclude how arguments in $\mathcal{AS}$ compare in strength. From the justified and / or the defensible conclusions, an argument order $\leq$ over arguments in $\mathcal{AS}$ will be constructed, which is used in the argumentation theory for $\mathcal{AS}$ and $\mathcal{K}$.

Figure 2.3: *Meta-Argumentation Theories and Object-level Argument Orderings*



The conclusions of the AF of a meta-level argumentation theory will be used to construct the argument ordering of object-level arguments. To do this we need to decide what acceptability semantics we will use. Every argumentation framework has one unique grounded extension and at least one preferred extension. If grounded semantics is used, then we will have exactly one argument ordering over object-level arguments. In contrast, if preferred semantics is used, then we may obtain multiple argument orderings over object-level arguments and thus multiple object-level argumentation theories. It is interesting to explore how multiple argumentation theories could be combined. For example, an argument is justified if it is justified in every argumentation framework that corresponds to an argumentation theory. For simplicity, this thesis will focus on using the grounded semantics.

Definition 2.41 (Grounded Argument Ordering) *Let $\mathcal{AF}$ be the argumentation framework of a meta-argumentation theory $\mathcal{AT}$ w.r.t. $\mathcal{AS}$ and $\mathcal{K}$.* The grounded argument ordering on the basis of $\mathcal{AT}$ *is a binary relation* $\leq$ *over* $\mathsf{Args}(\mathcal{AS})$ *such that* $A \leq B$ *iff* $\underline{A} \preceq \underline{B}$ *is a justified conclusion of* $\mathcal{AF}$.

Note that $\underline{A} \preceq \underline{B}$ is an unlabeled conclusion. Recall from Definition 2.22 that a conclusion is justified in an AF if and only if there is an argument in the grounded extension of that AF. Because each AF has exactly one grounded extension, there is exactly one grounded argument ordering on the basis of a meta-argumentation theory. Given a meta-argumentation theory, we can build an object-level argumentation theory with the grounded argument ordering of the meta-argumentation theory.

Definition 2.42 (Argumentation Theory Based On Meta-Argumentation) *Let $\mathcal{AT}$ be a meta-argumentation theory w.r.t. $\mathcal{AS}$ and $\mathcal{K}$.* The object-level argumentation theory based on $\mathcal{AT}$ *is an argumentation theory* $\langle \mathcal{AS}, \mathcal{K} \leq \rangle$ *where* $\leq$ *is the grounded argument ordering on the basis of* $\mathcal{AT}$.

To ensure that the conclusions of an object-level argumentation theory based on a meta-argumentation theory satisfy the rationality postulates as described in Section 2.1.5, it suffices to show that grounded argument orderings satisfy either the last-link or weakest-link principle. First we need to show that the conclusions of a meta-argumentation theory satisfy the rationality postulates. Let $\mathcal{AT} = \langle \mathcal{AS}, \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle, \leq \rangle$ be a meta-argumentation with $\mathcal{AS} = \langle \mathcal{L}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$. Then $\mathcal{AT}$ satisfies the rationality postulates iff (1) $\mathcal{AT}$ is well-formed; (2) $\mathcal{SR}$ is closed under transposition or contraposition; (3) $\leq$ satisfies the last-link or weakest-link principle[8]; and (4) the closure of $\mathcal{K}_{\mathsf{np}}$ under $\mathcal{SR}$ is consistent.

By Definition 2.38, the strict rules in a meta-argumentation system are closed under transposition or contraposition. By Definition 2.40 the argument ordering in a meta-argumentation theory satisfies the last-link or weakest-link principle. Furthermore, by the definition of a contrariness function for meta-languages (Defintion 2.37), there there are no formulae that have a contrary. Consequently, every meta-argumentation theory is well-formed. We thus only need to show that the closure of the necessary premises of a meta-knowledge base under strict rule application is consistent and that the grounded argument ordering satisfies the last-link

[8] In fact, Prakken proves this for a certain class of argument orderings that are called *reasonable*. However, he shows that argument orderings that satisfy the last-link or weakest-link principle are reasonable. Because we will use the last-link principle, we will only look at these argument orderings.

or weakest-link principle. Because the necessary premises of a meta-knowledge base only contains the axioms in Table 2.4, we need to show that this set of axioms is consistent.

Proposition 2.1 *Let $\mathcal{K}'_{\mathsf{np}}$ be the set of axioms in Table 2.4. The closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent.*

Proof To prove that the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent we need to find a model in which all formulae in $\mathcal{K}'_{\mathsf{np}}$ are true and to show that applying strict rules does not lead to inconsistencies. We will now construct such a model $M = \langle \langle D, R, F \rangle, I \rangle$ with I the interpretation function[9] , and

$$\begin{aligned} D =& \{o_\emptyset, o_1, o_2\} \\ R =& \{\mathsf{in}, \preceq_{\mathsf{Args}}, \preceq_{\mathcal{R}}, \preceq_{\mathcal{L}}, \mathsf{np}, \mathsf{op}, \mathsf{as}\} \\ F =& \{\mathsf{cf}, \mathsf{rules}, \mathsf{lastDef}, \mathsf{dirSub}, \mathsf{premises}, \mathsf{sub}, \mathsf{conc}\} \end{aligned}$$

the set of objects, relations and functions respectively. The constants in the meta-language are interpreted as follows:

- $I(\underline{\emptyset}) = I(\underline{\mathcal{DR}}) = I(\underline{\mathcal{SR}}) = o_\emptyset$
- $I(\underline{\phi}) = o_1$, and $I(\underline{\{\phi\}}) = I(\underline{\mathsf{Args}(\mathcal{AS})}) = I(\underline{\mathcal{L}}) = o_2$

The predicates in the meta-language are interpreted as:

- $I(\mathsf{np}) = I(\mathsf{op}) = I(\mathsf{as}) = \emptyset$
- $I(\preceq_{\mathsf{Args}}) = \{(o_1, o_1)\}$, $I(\preceq_{\mathcal{R}}) = \{(o_\emptyset, o_\emptyset)\}$ and $I(\preceq_{\mathcal{L}}) = \{(o_\emptyset, o_\emptyset), (o_2, o_2)\}$
- $I(\mathsf{in}) = \{(o_\emptyset, o_2), (o_1, o_2)\}$

The function symbols in the meta-language are interpreted as follows:

- $I(\mathsf{cf}) = \{(o_1, o_\emptyset)\}$, i.e., $I(\mathsf{cf})$ maps o_1 onto $o_\emptyset$ and is undefined for the rest.
- $I(\mathsf{rules}) = I(\mathsf{lastDef}) = I(\mathsf{dirSub}) = \{(o_1, o_\emptyset)\}$,
- $I(\mathsf{premises}) = I(\mathsf{sub}) = \{(o_1, o_2)\}$,
- $I(\mathsf{conc}) = \{(o_1, o_1)\}$

Recall from Table 2.3 that the predicates symbols strict, firm and $\prec$ are abbreviations. Their interpretation is thus as follows: $I(\mathsf{strict}) = \{(o_1)\}$, $I(\mathsf{firm}) = \emptyset$ and $I(\prec) = \emptyset$.

We will now show that all formulae in the set $\mathcal{K}'_{\mathsf{np}}$ (which consists of all axioms in Table 2.4) are true in model M. Because no x, y, and z can be found to make the antecedent of $\mathtt{trns}'_{\mathsf{Args}}$ true, axiom $\mathtt{trns}'_{\mathsf{Args}}$ is true in M. For the same reason, $\mathtt{trns}'_{\mathcal{R}}$ and $\mathtt{trns}'_{\mathcal{L}}$ are true. Axiom $\mathtt{rflx}'_{\mathsf{Args}}$ is true in M because for only ϕ it is true that $\mathsf{in}(\phi, \mathsf{Args}(\mathcal{AS}))$ and it is true that $\phi \preceq_{\mathsf{Args}} \phi$. Axiom $\mathtt{rflx}'_{\mathcal{R}}$ is true in M because for only $\underline{\emptyset}$ it is true that $\underline{\emptyset} \subseteq \mathcal{DR}$ and it is true that $\underline{\emptyset} \preceq_{\mathcal{R}} \underline{\emptyset}$. Similarly, axiom $\mathtt{rflx}'_{\mathcal{L}}$ is true in M because (1) for $\underline{\emptyset} \subseteq \mathcal{L}$ it is true that $\underline{\emptyset} \preceq_{\mathcal{L}} \underline{\emptyset}$; and, (2) for $\underline{\{\phi\}} \subseteq \mathcal{L}$ it is true that $\underline{\{\phi\}} \preceq_{\mathcal{L}} \underline{\{\phi\}}$. Axioms $\mathtt{adm}'_1$, $\mathtt{adm}'_2$, $\mathtt{lstLnk}'_1$ and $\mathtt{lstLnk}'_2$ are true in M because there are no x and y that make their antecedents true.

[9] An interpretation function I assigns (1) an object $I(c) \in D$ to each constant c in the meta-language; (2) a set of n-tuples of objects to each n-ary predicate symbol in the meta-language; and, (3) a function mapping n-tuples of elements of D to elements of D to each n-ary function symbol.

Similarly, the axioms $\mathtt{set}'_{\mathcal{R}}$, $\mathtt{set}'_{\mathcal{L}}$, $\mathtt{prem}'_1$ and $\mathtt{prem}'_2$ are true because there are no x and y that make their antecedents true.

Because all formulae in $\mathcal{K}'_{\mathsf{np}}$ are universally quantified, the inference rule universal generalization can be applied freely without adding (semantically) new formulae. Also material implication can be applied freely (and vacuously for members of $\mathcal{K}'_{\mathsf{np}}$, for which the antecedent of all material implications is always false). In summary, because we can find a model in which all formulae in $\mathcal{K}'_{\mathsf{np}}$ are true and the application of strict rules does not lead to inconsistencies, the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent. ■

Because of Proposition 2.1, the conclusions of a meta-argumentation theory are consistent. This means that its grounded extension will not contain two arguments A and B such that $\mathsf{conc}(A) \in \mathsf{cf}(\mathsf{conc}(B))$. We still need to show that the grounded argument ordering satisfies the last-link or weakest-link principle. For this we will first show that strict and firm arguments are always in the grounded extension.

Lemma 2.1 *Every strict and firm argument is in the grounded extension.*

Proof A strict argument cannot be undercut because only defeasible inference rules can be undercut. Arguments can only be rebutted on sub-arguments that have applied a defeasible rule last. A strict argument can thus not be rebutted. Arguments cannot be undermined on a premise that is a necessary premise. Since firm arguments by definition only have premises that are necessary premises, firm arguments cannot undermined. Consequently, a strict and firm argument cannot be undercut, rebutted or undermined and thus cannot be attacked and thus must be in the grounded extension. ■

Making use of this lemma, we will now show that every grounded argument ordering is reflexive, which is required to show that grounded argument orderings are indeed argument orderings.

Proposition 2.2 *Every grounded argument ordering is reflexive.*

Proof By Definition 2.40, the knowledge base of every meta-argumentation theory $\mathcal{AT}'$ contains axiom $\mathtt{rflx}'_{\mathsf{Args}}$ stating $\underline{x} \preceq_{\mathsf{Args}} \underline{x}$. This means that the AF of the $\mathcal{AT}'$ contains an atomic argument concluding $\underline{A} \preceq_{\mathsf{Args}} \underline{A}$ for every argument-constant $\underline{A}$. Because such arguments are strict and firm, Lemma 2.1 shows it will be in the grounded extension and thus the grounded argument ordering is reflexive.

The following properties can be proved in a similar way using the other axioms in a meta-knowledge base:

- because of $\mathtt{trns}'_{\mathsf{Args}}$, the grounded argument ordering $\leq$ is transitive,
- because of $\mathtt{adm}'_1$, the grounded argument ordering has $A < B$ if B is a strict and firm object-level argument and A is not a strict and firm object-level argument, and
- because of $\mathtt{adm}'_2$, if the last applied rule of object-level argument A is strict, then for all direct subarguments A' it is true for the grounded argument ordering that $A \leq A'$ and for one direct subargument A' it is true that $A' \leq A$.

Because these properties hold, every grounded argument ordering is in fact an argument ordering as defined in Definition 2.12.

Now that we have shown that a grounded argument ordering is in fact an argument ordering, we still need to show that a grounded argument ordering satisfies either the last-link or weakest-link principle. The axioms of Table 2.4 model the last-link principle.

Proposition 2.3 *Every grounded argument ordering satisfies the last-link principle.*

Proof Axioms $\texttt{lstLnk}'_1$ and $\texttt{lstLnk}'_2$ directly model the last-link principle. Furthermore, Lemma 2.1 shows that strict and firm arguments are in the grounded extension and Proposition 2.1 shows that the set of axioms is consistent under strict rule application. Consequently, it is straightforward to see that the grounded argument ordering satisfies the last-link principle. ■

Summarizing, this section has proposed a meta-level argumentation framework to reason about the relative strength of object-level arguments. Evaluating the meta-level arguments results in an argument ordering over object-level arguments. In order for the object-level argumentation to satisfy the rationality postulates described in Section 2.1.5, it suffices that the object-level argument ordering satisfies the last-link or weakest-link principle. We have then shown that the meta-level argumentation framework satisfies the rationality postulates and that the object-level argument ordering it describes satisfies the last-link principle.

2.3.4 Summary on Argumentation

The main research aim in this thesis is to investigate how argumentation can be used to support decision making in complex scenarios. For this purpose, Section 2.1 provided a background on argumentation and discussed one particular argumentation system, ASPIC+, in detail. When arguing about what decision to make it is common that there are multiple arguments in favor of and multiple against a decision. To compare decisions it is necessary to combine the strengths of these arguments. This is called the accrual of arguments. Section 2.2 makes the the accrual mechanism of Prakken (2005b) more precise and integrates the mechanism in the ASPIC+ framework. This is a contribution of this thesis and does therefore not belong in the background chapter. However, we wanted that the argumentation mechanism was in one chapter and therefore we included it in this chapter.

Another element that is required to argue about what to do is to argue about the strength of arguments. For example, the argument to rescue the victims inside the burning building first may be a stronger argument than the argument to first extinguish the fire because it results in less material damage. In order to argue about the strength of arguments, this section proposed a meta-level argumentation system. This is also a contribution of this thesis and is included in this chapter for the same reason.

2.4 Decision Theory

To support making a decision, it is necessary to understand what makes one decision better than another. One of the most popular approaches for decision making can be found in decision theory and prescribes that a decision maker should make the decision with the highest

expected utility. *Decision theory* is mostly concerned with prescribing the best decision that a decision maker (henceforth DM) can make. To obtain a basic understanding of what the expected utility of a decision is, Section 2.4.1 introduces the notions of preferences, value, uncertainty and expected utility. To support a decision maker, it is necessary that the decision maker communicates what he values. Typically, he can only express himself in terms of attributes of outcomes, i.e., properties of outcomes. For example, no chemicals escaping into the environment is preferred to a large amount of chemicals escaping. To facilitate this, researchers have proposed multi-attribute utility theory on which we give a background in Section 2.4.2. Finally, because we want to support a decision maker in determining what he values more, Section 2.4.3 gives a background on the field of decision analysis, which focuses on techniques to find a (multi-attribute) utility function that correctly describes what a decision maker values. The argumentation-based mechanism that we propose in later chapters is based on these techniques.

2.4.1 Value

The decisions from which a DM can choose are abstractly described by a set of *alternatives*. The result of making a decision is called an *outcome*. We will use Alt and Ω to denote the set of all alternatives and outcomes respectively. Decision makers do not value all outcomes the same. These differences in value can be described by a preference relation.

Definition 2.43 (Preference Relation) *A* preference relation *over a set Ω of outcomes is a transitive and reflexive binary relation $\leq$ over Ω.*

If $o \leq o'$ is true then we will also say that o' is *weakly preferred* to o. Furthermore, $o < o'$ abbreviates $o \leq o'$ and not $o' \leq o$ and denotes that o' is *strictly preferred* to o. Also $o \equiv o'$ abbreviates $o \leq o'$ and $o' \leq o$ and denotes that o and o' are equally preferred. If for every $o, o' \in \Omega$ it is true that either $o \leq o'$ or $o' \leq o$, then $\leq$ is called *total*.

Another way to model what a DM values is to use a (subjective) *value function* (Brafman and Domshlak, 2009), which specifies how much value outcomes have.

Definition 2.44 (Value Function) *A* value function *is a function $v : \Omega \to \mathbb{R}$.*

A subjective value function for a DM describes his preferences as follows: for all outcomes $o_1, o_2 \in \Omega$: $o_1 \leq o_2$ iff $v(o_1) \leq v(o_2)$. The notion of the *utility* of an outcome for a DM is used to represent how desirable that outcome is to the DM. Value functions are also called *utility functions*.

Often there is uncertainty with respect to in what outcome choosing a certain alternative results. In such decision scenarios, the effect of a decision can be described with a conditional probability distribution. We will use $Pr(o \mid a)$ to denote the probability of outcome o given that alternative a is chosen. The expected utility of an alternative a, denoted $EU(a)$, is defined as follows.

$$EU(a) = \sum_{o \in \Omega} Pr(o \mid a) \cdot v(o)$$

Decision theory prescribes that rational decision makers should choose the alternative with the maximal expected utility (von Neumann et al., 1947; Savage, 1954).

2.4.2 Multi-Attribute Utility Theory

In the previous subsection, outcomes were entities without any structure. However, often outcomes can be given a structure that describes the outcome. Giving outcomes structure is convenient to elicit, specify and also reason about utility functions. In *Multi-Attribute Utility Theory* (MAUT), the outcomes of alternatives are described in terms of a set of *attributes* (Keeney and Raiffa, 1976). If there are n attributes, then an outcome is represented by an n-tuple, where the i-th number denotes the score or performance of the outcome on the i-th attribute. A multi-attribute utility function specifies the utilities of outcomes that are described using a set of attributes. The definitions in this subsection are taken from Brafman and Domshlak (2009).

We will use $\mathcal{A}$ to denote the set of all attributes and the variables x and y for specific attributes. Each attribute has a domain of *attribute values* that outcomes can have. If V denotes the set of all possible attribute values, then the function $\mathsf{dom}(x) : \mathcal{A} \rightarrow 2^V$ returns the set of attribute values that can be assigned on a given attribute, i.e., dom returns the domain of attribute values of an attribute. Note that the domain of an attribute can be infinite. If $X \subseteq \mathcal{A}$, then the function $\mathsf{dom}(X)$ returns all combinations of attribute values of the attributes in X. Furthermore, let X and Y be two disjoint subsets of $\mathcal{A}$. If $x \in \mathsf{dom}(X)$ and $y \in \mathsf{dom}(Y)$, then $xy \in \mathsf{dom}(X \cup Y)$. For example, let $\mathcal{A} = \{x_1, x_2, x_3\}$ with $\mathsf{dom}(x_1) = \{v_1, v_2\}$ and $\mathsf{dom}(x_2) = \{w_1, w_2\}$. If $X = \{x_1, x_2\}$, then $\mathsf{dom}(X) = \{(v_1, w_1), (v_1, w_2), (v_2, w_1), (v_2, w_2)\}$.

Definition 2.45 (Multi-Attribute Utility Function) *Let $\mathcal{A} = \{x_1, \ldots, x_n\}$ be the set of all attributes. A function of the form $u : \mathsf{dom}(x_1) \times \ldots \times \mathsf{dom}(x_n) \rightarrow \Re$ is called a* multi-attribute utility function *over $\mathcal{A}$.*

Often in the preferences of a decision maker there are independencies between attributes. Independencies between attributes allow for more compact representations of utility functions and easier elicitation. Let X, Y be a partition of $\mathcal{A}$. The set of attributes X is *preferentially independent* of the set of attributes Y if the preferences over outcomes involving only changes in the levels of the attributes in X do not depend on the levels at which the attributes in Y are fixed.

Definition 2.46 (Preferential Independence) *Let $X \subset \mathcal{A}$ and $Y = \mathcal{A} \setminus X$. The set of attributes X is* preferentially independent *of set of attributes Y if and only if*

$$\forall x_1, x_2 \in \mathsf{dom}(X) : \big(\exists y \in \mathsf{dom}(Y) : x_1 y \leq x_2 y\big) \supset \forall y \in \mathsf{dom}(Y) : x_1 y \leq x_2 y$$

Preferential independence is not a symmetrical relationship, i.e., X being preferentially independent of Y does not imply Y being preferentially independent of X.

Example 2.4 (Preferential Independence and Cars) This example is taken from Brafman and Domshlak (2009). Suppose the outcome of buying a car is described by two attributes: price (cheap or expensive) and brand (BMV or Toyota). I always prefer a cheaper car to a more expensive one for any fixed brand. This means that the attribute price is preferentially independent of the attribute brand. This statement encodes the following preferences.

$$(\mathsf{cheap\ Toyota}) < (\mathsf{expensive\ Toyota})$$
$$(\mathsf{cheap\ BMW}) < (\mathsf{expensive\ BMW})$$

However, given that the car is cheap, I prefer Toyota to BMW, whereas if the car is expensive, then I prefer BMW to Toyota. Because preferences between brands depend on the price, the attribute brand is not preferentially independent of the attribute price.

$$(\text{cheap BMW}) < (\text{cheap Toyota})$$
$$(\text{expensive Toyota}) < (\text{expensive BMW})$$

The two statements combined result in the following preference ordering:

$$(\text{expensive Toyota}) < (\text{expensive BMW}) < (\text{cheap BMW}) < (\text{cheap Toyota})$$

We will now look at independence between attributes in value functions rather than in preferences. The most widely used independence assumption in value functions is called *additive independence* as is defined as follows.

Definition 2.47 (Additive Independence) *Let $\mathcal{A} = \{x_1, \ldots, x_n\}$ and let $x_1, \ldots, x_n$ be mutually preferential independent. Value function V is* additive independent *iff*

$$V(x_1, \ldots, x_n) = V_1(x_1) + \ldots + V_n(x_n)$$

An additive independent value function (AIVF) is much easier to elicit because we only need to ask preferences among individual attributes. An AIVF can be represented in a compact way and computation only requires $O(n)$ parameters. However, additive independence is a strong assumption because preferences and their strength must be unconditional.

2.4.3 Decision Analysis

In complex decisions scenarios, there are many aspects of what a decision maker values. This can make determining your preferences even between two outcomes difficult. For example, if you are buying a house, then it is hard to determine what house to prefer because so many different aspects matter. The field of *decision analysis* investigates how decision makers can be helped finding the utility function that corresponds to their preferences. Multiple approaches have been proposed for this, but here we will describe three of the most influential approaches.

In Keeney and Raiffa (1976) and Keeney (1992), what matters to a DM is decomposed into a hierarchy of objectives that the DM has. An objective is characterized by a decision context, an object, and a direction of preference. For example, in the decision context of the running example, some objectives are to minimize casualties and maximize safety of your personnel. The objective hierarchy decomposes what motivates the DM into so-called fundamental objectives. Fundamental objectives are further decomposed into means-objectives, which are further decomposed until they are operational.

In a similar fashion, von Winterfeldt and Edwards (1986) decompose what an agent values into a so-called *value tree*. A value tree hierarchically relates general areas of concern, intermediate objectives, and specific evaluation criteria defined on measurable attributes. The purpose of a value tree is to explicate and operationalize higher level values.

Another possible technique that can be used to determine what the DM cares about is the *Analytical Hierarchical Process* (AHP) (Saaty, 2008). The AHP decomposes what a DM

values in a hierarchy of criteria and sub-criteria. Next, the DM makes judgments about the importance of the elements. These judgments are then quantified and used to determine what decision is best. In this manner, intangible values of the DM (parts that are difficult to define) are made measurable by assigning criteria and sub-criteria to them.

What is common to all these approaches is that they identify elements that matter to the DM and structure these elements in a hierarchical manner. This hierarchy has abstract elements in the top and concrete elements in the bottom.

2.5 Abstract Values

The values that a person holds are the reason that a person is motivated to spend any effort in decision making (Keeney, 1992). In this section we will explore the psychology literature about the (abstract) values that people pursue. Researchers have found ten basic value types that people all over the world find important and have found structure between these value types. By understanding better what values people pursue, a decision support system can better predict and understand what a person values and thus support him better. Argumentation theory has also recognized the importance of values and the next section will describe how values are used there.

Abstract values are a person's principles or standards of behavior or a person's judgment of what is important in life. Examples of values that people pursue are honesty, politeness, social order, pleasure, success, freedom, independence, and equality. In the literature, abstract values are referred to as simply 'values', but because the word 'value' is used in so many different ways, e.g., numerical values, utility values, as a verb, as a noun, and so on, we will use *abstract values* to denote the values that a person pursues. If it is clear from the context, we will use 'values' rather than 'abstract values'.

Different areas of research have argued that abstract values are important in decision-making. In psychology, abstract values are seen as motivational constructs that people use as criteria to evaluate actions, people, and events (Allport and Willard, 1961; Rokeach, 1973; Schwartz, 1992). Although abstract values are defined in a comprehensive manner and are well tested empirically, work remains to be done on how to formalize abstract values. In argumentation theory, the different abstract values that people pursue are considered to be the cause of rational disagreement between people (Perelman and Olbrechts-Tyteca, 1969). In argumentation in A.I., abstract values have been used for practical reasoning, e.g., see Atkinson et al. (2006); Bench-Capon (2003); Grasso et al. (2000). However, the way how abstract values are used is not grounded in psychology literature. In decision theory, decision analysis focuses on how to elicit what a person values (Keeney, 1992), but the link with abstract values in psychology is not discussed.

The abstract values that a person pursues have a significant impact on what decisions he makes. Therefore, it is important to formalize abstract values, so that they can be used in decision making even though there are many inconsistencies between the works that do define what values are (Rohan, 2000). In this section, we will first explore how the psychology literature defines abstract values. Next, we discuss what abstract values people all over the world pursue and what correlations have been found concerning how important people find specific abstract values. Finally, we will discuss how values are used in the argumentation literature.

2.5.1 What Are Abstract Values?

In the psychology literature there is a consensus about the following five features of abstract values (Rohan, 2000; Schwartz and Bilsky, 1987).

1. *Values are beliefs*. In other words, a value is a conception of the desirable. However, values are not objective beliefs, values are tied to emotions. Because a value is a conception of the desirable, it is possible to discuss what a value means and what importance is given to it. However, because values are tied to emotions, it may be hard to convince someone.

2. *Values are a motivational construct*. Values describe desirable goals that people want to achieve.

3. *Values are what is called 'trans-situational'*. In other words, values transcend specific actions and situations and are therefore 'abstract goals'. The abstract nature of values distinguishes values from concepts like norms and attitudes, which usually refer to specific actions, objects, or situations.

4. *Values guide selection and evaluation of behavior and events*. Values serve as standards or criteria.

5. *Values are relatively ordered according to importance*. The values people pursue are structured in a value system in which each value is given a relative importance to other values. This hierarchical feature of values also distinguishes them from norms and attitudes.

Abstract values cannot be achieved, but they can be pursued (Lewin, 1951). Although an abstract value cannot be true or false, a statement about an abstract value can be made. For example, the statement that some agent pursues a certain abstract value is a belief that can be true or false. Other kinds of statements about abstract values are that one abstract value is more important than another to some agent or that a certain criterion is used to measure an abstract value.

Because abstract values are used to evaluate or judge outcomes and to make decisions, they can be seen as criteria. This means that one outcome may be better than another from the perspective of some abstract value, but worse from the perspective of another abstract value. In other words, abstract values can be *promoted* and *demoted* (Bench-Capon, 2003). *Promoting an abstract value* means that a change resulted in something that is preferred to what was before from the perspective of that abstract value. Promoting an abstract value can also be done relative to some action. For example, compared to action B, action A promotes value V. Similarly, *demoting an abstract value* means that the result of something is preferred less to what was before from the perspective of that abstract value.

Furthermore, abstract values are what is called 'trans-situational', meaning that they can be used to evaluate all kinds of situations. This means that agents want to promote all their abstract values in every decision context. Because abstract values are abstract, agents must interpret how abstract values can be pursued in a specific decision context. For example, people interpret the abstract value of fairness differently when the decision context is a sport like kickboxing than when the decision context is the supermarket. Although different persons may both find an abstract value important, they may not agree about its interpretation

in a specific decision context. For example, some people may find that giving a murderer the death penalty promotes fairness, whereas other people find that it demotes fairness. The reason for this disagreement is that different people have a different conception of what some abstract value means in a specific context.

2.5.2 Human Abstract Values

Much research has investigated what abstract values people in all parts of the world pursue. At first, researchers produced lists of values with the aim to be comprehensive as to all the values people in the world can pursue. The most influential of such lists are the 18 values proposed by Rokeach (1973). Rokeach's list of abstract values lacks an underlying structure, which makes it impossible to understand the relations between abstract values. If a person finds one abstract value very important, then he typically finds another abstract value less important. For example, if a person finds security very important then he typically finds privacy less important. An underlying structure of abstract values provides a way to understand otherwise unrelated correlations between the importance of abstract values.

In Schwartz (1992), values were seen as responses to universal requirements to human functioning and *value types* were derived from or motivated as ways to deal with these universal requirements. Schwartz (1992) identifies three very general universal requirements for human functioning and uses them to derive ten different motivational goals for dealing with different requirements for human functioning. These ten different motivational goals were used to define 10 value types, which are described in Table 2.5. The intention behind the ten basic values is to include all the core values recognized in cultures around the world. These ten value types are classified in four classes that are organized along two dimensions.

2.5.3 Priorities between Values

The underlying assumption of value theories is that all humans agree on the importance of a finite number of value types, but that individuals disagree in terms of relative importance of these value types. The relative importance a person assigns to each value type is called that person's *value priorities*. All basic value types are important in human functioning, but value priorities are seen as an individual's response to the universal requirements of human existence. People's value priorities are how they respond to what they ought to do to survive and strive for in their social environments. If a person's social environment changes, then he may need to respond in a different way to be successful. A change in someone's life can therefore bring about a change in value priorities. For example, Altemeyer and Smith (1988) found that when people become parents, they tend to find tradition, conformity, and security values more important.

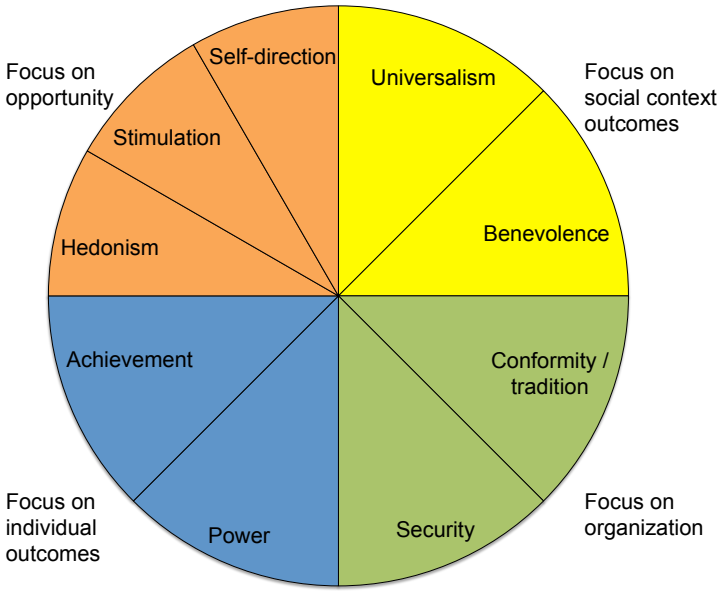
Schwartz (1992) also proposes a structure between basic values types that is based on the observation that some basic values are congruent and others are often in conflict. For example, the pursuit of achievement values is often in conflict with the pursuit of benevolence values because actions to increase success typically have a negative effect on the success of others. In contrast, actions in pursuit of tradition values often are congruent with actions in pursuit of conformity values such as obedience. Figure 2.4 is taken from Schwartz and visualizes the congruence and conflict relations between basic values types: the closer two

Table 2.5: *Schwartz (1992)'s Basic Value Types and their Motivational Goals and Representative Values*

VALUE TYPE	MOTIVATIONAL GOAL	REPRESENTATIVE VALUES
POWER	Social status and prestige, control or dominance over people and resources.	Social power, authority, wealth.
ACHIEVEMENT	Personal success through demonstrating competence according to social standards.	Success, capability, ambition, or influence.
HEDONISM	Pleasure and sensuous gratification of oneself.	Pleasure, enjoyment in life.
STIMULATION	Excitement, novelty and challenge in life.	Daring, a varied life, an exciting life.
SELF-DIRECTION	Independent thought and action-choosing, creating, exploring.	Creativity, freedom, independence, self-reliance, self-sufficiency, curiosity, choose own goals.
UNIVERSALISM	Understanding, appreciation, tolerance, and protection of the welfare of all people and of nature.	Broadminded, wisdom, social justice, equality, peace, beauty, unity with nature, the environment.
BENEVOLENCE	Preservation and enhancement of the welfare of people with whom one is in frequent personal contact.	Helpful, honesty, forgiving, loyalty, responsibility.
TRADITION	Respect, commitment, and acceptance of customs and ideas that traditional culture or religion provide.	Humility, submission to life's circumstances, devotion, respect for tradition, moderateness.
CONFORMITY	Restraint of actions, inclinations, and impulses likely to upset or harm others and violate social expectations or norms.	Politeness, obedience, self-discipline, honor parents and elders.
SECURITY	Safety, harmony, and stability of society, relationships and self.	Family security, national security, social order, cleanliness, reciprocation of favors.

values are in either direction around the circle (e.g., if they are adjacent), the more congruent they are. The more distant two values are (e.g., if they are opposing), the more they conflict.

Figure 2.4: *Structure between basic value types: the closer two values in either direction around the circle, the more congruent they are.*



The structure visualized in Figure 2.4 can be summarized in two dimensions: (1) focus on opportunity versus focus on organization and (2) focus on social context outcomes versus focus on individual outcomes.

Summarizing, psychological research has shown that people all over the world all recognize the importance of a set of ten basic value types. However, each individual may give each basic value type a different priority. Researchers have also found correlations between what priorities people give to the basic value types. These findings are useful in a decision support system because it allows the system to make better predictions concerning an individual's priorities between perspectives, which also allows the system to make better suggestions concerning what values the decision maker cares about in a certain decision context.

2.5.4 Abstract Values and Argumentation

This section described literature on what abstract values are and what kinds of values people hold all over the world. The values that people pursue have a significant effect on their preferences and are thus important in decision making. This has been recognized in the argumentation literature. We will now explore the argumentation literature about practical reasoning and abstract values. Using the running example, we will determine what can be done and what cannot yet be done.

Walton (1996) uses presumptive justification and critical questions for practical reasoning in the form of two argument schemes. One scheme is called the necessary condition scheme

and the other is called the sufficient condition scheme. The sufficient condition scheme is the following: *G is a goal for agent A, doing action X is sufficient for agent A to achieve goal G, therefore, agent A ought to do action X.*

Atkinson et al. (2006) take Walton's sufficient condition scheme and extend the notion of a goal into its subjective and objective components. The values that a goal promotes and demotes constitute the subjective component of a goal. The following argument scheme, named AS1, is proposed that an agent can use to argue about what action he should perform taking into account his goals and the values he wants to promote.

> In the current circumstances R,
> agent α should perform action ac1,
> which will result in new circumstances S,
> which will realize goal G,
> which will promote value V.

Agents have a priority ordering over the values they pursue, which is used to determine the strength of the practical arguments that are generated. Suppose that argument A concludes that agent α should perform action a because it achieves goal G which promotes value V and that argument B concludes that α should perform action b because it achieves goal H which promotes value W. Arguments A and B attack each other because the agent can only choose to perform one action. If agent α finds value V more important than value W, then argument A is stronger than argument B and thus A defeats B.

AS1 is formalized using a set of states Q and set of values V. The values of an agent are modeled with a 'valuation function' specified as $val : V \times Q \times Q \rightarrow \{+, -, 0\}$. If $val(v, q_1, q_2) = +$, then the state transition from q_1 to q_2 *promotes* the value v. If $val(v, q_1, q_2) = 0$, then the transition *demotes* value v. Finally, if $val(v, q_1, q_2) = 0$, then the transition is *neutral towards* value v.

Bench-Capon and Atkinson (2009) extend the formalism of Atkinson et al. (2006) to take into account temporal aspects and the intrinsic worth of actions. In the extension, agents can also justify an action if that action promotes a value without achieving a goal. Three additional argument schemes are proposed for practical reasoning.

1. AS1a: *In the current circumstances R, Agent α should perform action $ac1$, which will promote value V.*
2. AS2: *In the current circumstances R, Agent α should perform action $ac1$, since otherwise goal G will not be realized, and realizing G would promote value V.*
3. AS2a: *In the current circumstances R, Agent α should perform action $ac1$, since otherwise it will not be possible to perform $ac2$, which would promote value V.*

In the original paper, values were used to justify goals ("Looking to values, these in turn are different from goals as they provide the actual reasons for which an agent wishes to achieve a goal"), whereas in the new paper, goals are not directly related to value, but instead a transition of one state to another can promote, demote or be neutral towards a value.

The valuation function completely specifies what a particular value means. However, the number of state transitions grows exponentially with the number of states. This requires the designer of the agent to specify a complicated valuation function. Moreover, what criteria should the designer use to determine whether some state transition promotes or demotes a

value? If the designer uses several criteria for this, then the agent could also be equipped with these criteria and a mechanism to determine whether a value is promoted.

Perelman wrote "If men oppose each other concerning a decision to be taken, it is not because they commit some error of logic or calculation. They discuss apropos the applicable rule, the ends to be considered, the meaning to be given to values, the interpretation and characterisation of facts." (Perelman and Olbrechts-Tyteca, 1969, p. 150). If agents disagree about what constitutes a particular value, i.e., the meaning that is given to values, then they should argumentation to resolve their disagreement. Although in Atkinson and Bench-Capon (2007a) it is possible that agents disagree about the meaning of a value, it is not possible to discuss this matter. By arguing about what constitutes a value, agents may make a better decision. For example, an agent may have forgotten to consider an important criterion for some value. Values are used to justify goals, but it is not clear when a goal promotes or demotes a value. As Perelman wrote, what ends (i.e., what goals) to consider should be discussed when making a decision. A mechanism to justify a goal with a value would facilitate agents to argue about what goals they should adopt.

Running example

In the running example described in Chapter 1, a decision support system is helping a fire-commander student with deciding what alternative to choose in a situation where a factory is on fire with people inside and with toxic chemicals that could escape into the environment. Both the value of the safety of the people inside and the value of the environment are important to the user, but it is not clear a priori what alternative results in the safest outcome.

Let action A denote 'first send firefighters in to rescue the victims, then extinguish the fire' and action B denote 'first extinguish fire, then send firefighters in to rescue the victims'. When arguing about what decision the student should take in the running example, the student gives the following argument Arg1 instantiating argument scheme AS1.

> In the current circumstances,
> the student should perform action A,
> which will result in victims being saved in 10 minutes,
> which will realize the goal of getting the victims out in 10 minutes,
> which will promote the value of safety of victims.

Although action A gets the victim out quickly, it is not good for the environment since chemicals will escape into the environment. Action B (extinguish first, then rescue) does prevent any chemicals from escaping. Consequently, the following argument Arg2 could be given instantiating argument scheme AS1.

> In the current circumstances,
> the student should perform action B,
> which will result in new circumstances S,
> which will realize goal that no chemicals escape into the environment,
> which will promote the value of the environment.

Arguments Arg1 and Arg2 have conflicting conclusions concerning what the student should do and therefore attack each other. However, because the student finds the value

of safety more important than the value of environment, the first argument defeats the second argument.

Let C denote the action 'first extinguish fire near victims, then rescue them and finally extinguish the remaining fire'. Action[10] C results in the fire near the victim being extinguished in five minutes. Then the firefighters still need to go in to rescue the victims, which takes ten minutes. In total, the victims are inside for fifteen minutes and thus C does not achieve the student's goal of getting the victims out in ten minutes. Furthermore, because C does not extinguish the entire fire immediately, some chemicals will escape, which prevents C from achieving the student's goal of no chemicals escaping. In summary, even though C is in fact safer for the victims and only has a small effect on the environment, no argument can be constructed concluding that C should be done.

To promote the value of safety, the student uses the criterion to minimize the time that victims are inside. However, this criterion is rather coarse. A criterion like minimizing the time that victims are near fire is a better criterion and allows for that actions like C are considered adequately. It would thus be interesting to discuss *why* the student has the goal to get the victims out in ten minutes. To do this, the approach of Atkinson et al. (2006) needs to be extended.

[10] If an action comprises of multiple sequential actions, then it can be seen as a plan. Medellin-Gasque et al. (2011) and Tonolio et al. (2011) have extended Atkinson's practical argumentation scheme to incorporate plans.

3 Conceptual Framework For Value

What a person values is the motivation to spend any effort in making a decision (Keeney (1992)). To determine what decision is the best for the user it is thus required to determine what decision an agent values most. Value is a starting point in making a decision. Therefore, this chapter has proposed a conceptual framework for what value is and how to argue with and about value. This conceptual framework will be referred to as the Perspective-based Value Model (PVM). By doing this, we address research question 1a. For a large part this conceptual framework is inspired and based on Keeney (1992), Hansson (2001), and Schwartz (1992). Chapter 4 formalizes this conceptual framework. In addition, this chapter proposes a number of argument schemes[1] that capture stereotypical patterns of reasoning with respect to value. Chapter 5 proposes an argumentation logic in which these argument schemes are formalized as defeasible inference rules. This argumentation logic is based on the ASPIC+ framework, that we have introduced in Chapter 2.1. Finally, Chapter 6 proposes how the conceptual framework of this chapter can be used in practical reasoning.

In decision-making, the notions of *alternative*, *outcome*, and *attributes of outcomes* are central (see Section 2.4 for a background). If we talk about the *value* of some element or elements, then these elements are either alternatives, outcomes, or attribute values. Section 3.1 describes the concepts that this thesis uses that can have value for an agent. A *value language* is a tool that can be used to express what one values. A value language consists of a number of *value statements*, which express what a user values / desires / cares about. Value statements are made from a perspective. For example, a house is good from the perspective of location, worse than another from the perspective of costs, or the more central a house is the better it is from the perspective of location. Hansson (2001) assumes that all value statements are made according to the same criterion, or as we say, from the same perspective. This is called *criterial constancy*. We will not assume criterial constancy and will therefore explicitly represent the perspective from which a value statement is made. Therefore, perspective is a central notion in this thesis.

We will distinguish between monadic and dyadic value statements. A value statement that concerns one element is called *monadic*, e.g., car A is beautiful. A value statement that expresses how two elements compare in value is called *dyadic*, e.g., car A is faster than car

[1] Recall from Section 2.1 that argument schemes are stereotypical patterns of reasoning. An argument scheme consists of a set of premises and a conclusion and states that if the premises are true, then, presumably, the conclusion is true. Critical questions associated to a scheme point to exceptional situations where the scheme cannot be applied.

B. Section 3.4 describes monadic value statements in detail and Section 3.2 dyadic value statements.

Typically there are many different sides to what people value, which complicates determining what element is valued more. However, people are typically able to decompose what is valued from their perspective into several other perspectives that are less abstract. For example, if I state that 'the faster, the better', then this is interpreted as 'the better a car is from the perspective of speed, the better it is from my perspective'. Such statements use how elements compare from one perspective (e.g., speed) to describe how elements compare from another perspective (e.g., some agent's perspective). Namely, given the speed of any two cars, 'the faster, the better' expresses what car I value more, i.e., what car I find better. Section 3.3 describes the notion of a value tree, which results from decomposing what an agent values into specific evaluation criteria that are used to compare outcomes. Furthermore, it describes the relation between value trees and the notion of abstract values as used in psychology literature (see Section 2.5).

3.1 Alternatives and Assignments

In a decision situation, a person, or more generally *an agent*, can choose to make one decision from a number of alternatives. Determining what decision is best for an agent is interpreted as determining what alternative is maximally preferred from the perspective of the agent. Rational decision making prescribes what alternative is rational for an agent to prefer the most. Performing an alternative results in exactly one outcome. However, it may not be known in advance in what outcome an alternative will result, but it may be known that the alternative results in one of a number of outcomes. To determine the value of alternatives, the value of the outcomes of those alternatives is used.

The outcomes of performing an alternative are described using *attributes* (also called variables or features). There may be many attributes with which outcomes can be described, but an agent typically only cares about a subset of those attributes. Caring about an attribute means that the attribute values of that attribute differ in the amount of value for the agent. If an agent does not care about an attribute, then each attribute value of that attribute has the same amount of value for that agent. Because what an agent values is subjective, different agents may care about different attributes in different ways.

An *attribute assignment* assigns an attribute value v to an attribute x such that v belongs to the domain of x. For example, if 'genre' is an attribute and 'comedy' is an attribute value of 'genre', then 'genre is comedy' is an attribute assignment. Following Boutilier et al. (2004), an *assignment* on the set of attributes X is a collection of attribute assignment for each attribute in X (this is formalized in Section 4.1). Assignments of the set of all attributes are called *complete*, others are called *partial*.

Example 3.1 (Assignments) Let 'genre' and 'lead actor' be two attributes, 'comedy' be an attribute value of the attribute 'genre', and 'Kevin Spacey' an attribute value of the attribute 'lead actor'. Then 'genre is comedy' is an assignment on the set of attributes containing only 'genre' and 'lead actor is Kevin Spacey' is an assignment on the set of attributes containing only 'lead actor'. Furthermore, 'genre is comedy and lead actor is Kevin Spacey' is an assignment on the set of attributes containing both 'genre' and 'lead actor'.

If s is an assignment on the set attributes X, t an assignment on the set attributes $X \cup Y$ and t makes the same attribute-assignments as s for each attribute in X, then we say that assignment s *satisfies* assignment t. In Example 3.1, the assignment 'genre is comedy and lead actor is Kevin Spacey' satisfies both the assignment 'genre is comedy' and the assignment 'lead actor is Kevin Spacey'.

Agents make value statements over assignments. For example, 'genre is comedy' has more value than 'genre is drama' from the perspective of fun, or 'genre is drama and lead actor is Kevin Spacey' has more value than 'genre is drama and lead actor is Jim Carey' from the perspective of an agent.

3.2 Dyadic / Comparative Value

For our purpose of practical reasoning, two kinds of elements are relevant over which an agent can express value: alternatives and assignments. Recall from Section 3.1 that assignments are more general than outcomes and attribute values.

Comparative statements like 'I prefer this to that' or 'the location of house 1 is better than the location of house 2' are common in informal discourse about what is valued. A *dyadic value statement* expresses how two elements compare in value from a certain perspective. A dyadic value statement is also called a *preference statement*.

If only one element can be chosen from a set E of elements, then dyadic value statements about E are called *mutually exclusive*. If multiple elements can be chosen from E, then dyadic value statements about E are called *combinative*. In this thesis, alternatives, outcomes and attribute values are structured such that they are mutually exclusive. Therefore, this thesis focuses on exclusionary dyadic value statements.

We distinguish four different (exclusionary) dyadic value statements.

1. *"element 1 is strictly preferred to element 2 from perspective p"* denoting that element 1 has more value than element 2 from perspective p.

2. *"element 1 is equally preferred to element 2 from perspective p"* denoting that element 1 and 2 have the same amount of value from perspective p. We may also say that perspective p is *indifferent* about element 1 and element 2.

3. *"element 1 is weakly preferred to element 2 from perspective p"* denoting that element 1 has as much as or more value than element 2 from perspective p. In other words, if this statement is true, then statement 1 or statement 2 is true.

4. *"element 1 and element 2 are incomparable from perspective p"* denoting that it is not possible to compare the value of element 1 and 2 from perspective p.

Note that the value of elements can be compared from perspectives like size or location, but that they can also be compared from the perspective of an agent. An agent's preferences between elements can be seen as how elements compare in value from the perspective of the agent. If agent α prefers element 1 to element 2, then we will also say that element 1 is preferred to element 2 from the perspective of agent α. Furthermore, some perspectives have a common name, but have different meanings for different agents. Such perspectives are called subjective. For example, different agents could order elements differently from

the perspective of 'fun'. Although the perspective of 'fun' has one name, it is a subjective perspective.

Example 3.2 (Comparing Houses From Different Perspectives) Suppose that there are three houses that agent α can buy. House 1 is in the city centre whereas houses 2 and 3 are in a suburb. Therefore, house 1 is strictly preferred to both house 2 and 3 from the perspective of location. Because house 2 and 3 both are in a suburb, they are equally preferred from the perspective of location. House 1 is a historic building whereas house 2 is designed by a famous contemporary designer. From the perspective of esthetics they are therefore incomparable. However, house 3 is in a flat from the 70s and therefore both house 2 and 3 are strictly preferred to house 3 from the perspective of esthetics.

3.2.1 Desired Properties

Weak, equal, strict and incomparable preference from a perspective are relations between elements. This section describes several properties of those relations that should hold.

Weak and equal preference from a perspective should be reflexive, meaning that each element is weakly and equally preferred to itself from each perspective. Strict and incomparable preference from a perspective should be irreflexive, meaning that no element is strictly preferred or incomparable to itself from any perspective. Furthermore, equal and incomparable preference from a perspective should be symmetrical, meaning that if element 1 is equally preferred or incomparable to element 2, then element 2 is also equally preferred or incomparable to element 1.

It is convenient and common to use weak preference as the primitive notion of preferences. By introducing strict, equal and incomparable preference from a perspective as derived relations, several desirable properties follow automatically. Let $e_1 <_p e_2$ denote that element 2 is strictly preferred to element 1 from perspective p, $e_1 \leq_p e_2$ denote that element 2 is weakly preferred to element 1 from perspective p, and $e_1 \equiv_p e_2$ denote that element 1 and 2 are equally preferred from perspective p. Then:

- $e_1 <_p e_2$ if and only $e_1 \leq_p e_2$ and $e_2 \not\leq_p e_1$,
- $e_1 \equiv_p e_2$ if and only if $e_1 \leq_p e_2$ and $e_2 \leq_p e_1$, and
- element 1 is incomparable to element 2 if and only if $e_1 \not\leq_p e_2$ and $e_2 \not\leq_p e_1$.

Note that this means that if two elements are incomparable, one cannot be strictly, weakly nor equally preferred to the other. Also, if element 1 is strictly preferred to element 2, then element 2 cannot be strictly preferred to element 1 and element 1 cannot be weakly or equally preferred to element 2. Consequently, assume argument A concludes that element 1 is strictly preferred to element 2 from perspective p and argument B concludes that element 2 is strictly preferred to element 1 from perspective p, then arguments A and B attack each other.

Completeness

In many works, orderings over elements are assumed to be complete. If an ordering is complete, then the value of every two elements can be compared. It is however sometimes difficult or even impossible to compare the value of every two elements. For example, it may be possible to compare the value of two pieces of music from the perspective of beauty, but it is

impossible to compare the value of a piece of music with the value of a drawing of your own child from the perspective of beauty. Furthermore, because comparing the value of elements from a perspective takes time and effort, it is possible that not all elements have been compared. For example, when making a decision, it is not interesting to compare the value of two elements when a third elements is better than both.

Transitivity

The most discussed logical property of preferences is the transitivity of weak preference (Luce, 1956). Because strict, equal, and incomparable preference are defined in terms of weak preference, if weak preference is transitive, then strict and equal preference are also transitive.

Many examples have been used to show that transitivity of weak preference does not hold in general. A classic counterexample that shows that preferences are not be transitive is the so-called Sorites Paradox. Consider 1000 cups of coffee that vary in the amount of sugar they contain and a user that likes sweet coffee. The cups are numbered $c_0, c_1, c_2, \ldots, c_{999}$ such that cup c_0 contains no sugar, c_1 contains one grain of sugar, c_2 two grains, and so on. Since the difference in sweetness between two adjacent cups is too small to notice, users will prefer two adjacent cups equally, i.e., c_i and c_j are preferred equally if $i + 1 = j$. Since the difference in sweetness between c_0 and c_{999} is significant, the user will strictly prefer c_{999} to c_0. However, if the user's preferences between coffee cups has the property of transitivity of indifference, then the user must prefer c_0 and c_{999} equally.

The examples demonstrating that indifference is not always transitive concern exceptional situations. Therefore, this thesis assumes that typically indifference is transitive and indifference is only not transitive in exceptional situations. Transitivity of indifference can be informally captured using the following defeasible argument scheme.

Argument Scheme 1: Transitivity of Indifference

Element 1 is equally preferred to element 2 from perspective p, and
element 2 is equally preferred to element 3 from perspective p,

---

therefore, presumably element 1 is equally preferred to element 3 from perspective p.

If the application domain is such that indifference is typically not transitive, then this argument scheme should not be used. Because transitivity of indifference does not always hold, but does hold typically, this argument scheme could be modeled as a defeasible inference rule, as done in Chapter 4. If on the one hand an agent should prefer two elements equally because of the transitivity of indifference, but on the other hand the agent knows explicitly that these two elements are not preferred equally, then for both conclusions an argument could be constructed. If the argument concluding that the two elements are not preferred equally is stronger, then it will defeat the transitivity of indifference argument.

Because there may be exceptions, the following defeasible argument scheme accounts for strict preference - indifference transitivity, abbreviated as PI transitivity.

Argument Scheme 2: PI Transitivity

Element 1 is strictly preferred to element 2 from perspective p, and
element 2 is equally preferred to element 3 from perspective p,

---

therefore, presumably element 1 is strictly preferred to element 3 from perspective p.

Similarly, we need to account for indifference - strict preference transitivity, abbreviated as IP transitivity. Again, since there may be exceptions to this kind of transitivity, we will use a defeasible argument scheme to model it.

Argument Scheme 3: IP Transitivity
Element 1 is equally preferred to element 2 from perspective p, and
element 2 is strictly preferred to element 3 from perspective p,

---

therefore, presumably element 1 is strictly preferred to element 3 from perspective p.

Finally, transitivity of strict preference can be captured using the following argument scheme. Also this argument scheme needs to be defeasible.

Argument Scheme 4: Transitivity of Strict Preference
Element 1 is strictly preferred to element 2 from perspective p, and
element 2 is strictly preferred to element 3 from perspective p,

---

therefore, presumably element 1 is strictly preferred to element 3 from perspective p.

These four argument schemes can be used to construct arguments that conclude how elements compare in value from a perspective.

Example 3.3 (Transitivity of preferences) Suppose that agent α knows that he strictly prefers assignment 1 to assignment 2 and that he strictly prefers assignment 2 to assignment 3. However, agent α does not know whether he prefers assignment 1 to assignment 3. Argument Scheme 4 can be applied to construct the following argument.

Assignment 1 is strictly preferred to assignment 2 from the perspective of agent α,
assignment 2 is strictly preferred to assignment 3 from the perspective of agent α,

---

therefore, presumably assignment 1 is strictly preferred to assignment 3 from the perspective of α.

Now agent α has a reason to believe that he prefers outcome 1 to outcome 3. If agent α decides to choose an alternative that results in outcome 1 rather than an alternative resulting in outcome 3, then agent α can use this argument to justify his decision.

3.2.2 Value in Practical Reasoning

Research question 1c is concerned with how the conceptual framework that we propose in this chapter can be used for practical reasoning. Because agents typically do not know a priori how they prefer alternatives, they need to reason about what alternative they prefer most. Furthermore, agents may need to explain why they prefer a certain alternative to another alternative.

To determine an agent's preferences between alternatives, the agent should look at the possible outcomes of all alternatives and at the attribute values of these outcomes for the attributes that he cares about. Given an agent's preferences between attribute values of attributes he cares about, the agent could reason about what outcome he prefers most. Given his risk attitude, in what outcomes the alternatives may result, and how he prefers those outcomes, the agent could then determine how he prefers alternatives. For example, a risk averse agent prefers the alternative whose worst possible outcome he prefers most, whereas an optimistic agent prefers the alternative whose best possible outcome he prefers most.

Preferences between alternatives, outcomes, and attribute values of the attributes of outcomes are thus relevant to determine what alternative the agent should choose. Furthermore, preferences between alternatives are determined using preferences between outcomes, and preferences between outcomes are determined using preferences between the attribute values of outcomes. Consequently, we need to represent preference statements between alternatives, between outcomes, and between attribute values. Moreover, we need inference rules that infer preference between alternatives using preference between outcomes, and inference rules that infer preference between outcomes from preference between attribute values. Chapter 6 proposes several methods to determine an agent's preferences between alternatives using his preferences between outcomes. Subsection 3.2.3 proposes how to determine an agent's preferences between outcomes using his preferences between attribute values.

3.2.3 Generalizing Value Statements

Recall from Section 3.1 that for each partial assignment s, there is a number of complete assignments that satisfy s. If a statement expresses value of a partial assignment s, then does that statement also something about the value of all other assignments that satisfy s? For example, does the statement 'the genre comedy is strictly preferred to the genre drama from the perspective of fun' say something about the value of the assignments 'genre is comedy and length is 100 minutes' and 'genre is drama and length is 101 minutes'.

It is natural to interpret a preference statement concerning a partial assignment s as that that statement expresses value over all the assignments that satisfy s. Because such a statement thus compactly represents a number of value statements over all the satisfying assignments, value statements containing a partial assignment are called *generalizing value statements*. There are however different ways as to how a generalizing value statement can be interpreted.

A simple interpretation of the statement 'assignment s is weakly preferred to assignment t from perspective p' is that each assignment that satisfies s is weakly preferred to each assignment that satisfies t from perspective p. For example, the preference statement 'the genre comedy is preferred to the genre drama from the perspective of fun' is interpreted as that all assignments with the genre comedy are preferred to all assignments with the genre drama from the perspective of fun. However, if there are multiple preference statements from the same perspective, conflicts may arise as illustrated with the following example.

Example 3.4 (Genre and Actors) Suppose that I have stated that I strictly prefer drama to comedy and that I strictly prefer movies with Kevin Spacey to movies with Jim Carey. Furthermore, movie 1 is a drama movie featuring Jim Carey, whereas movie 2 is a comedy featuring Kevin Spacey. Given these two preference statements over genre and actors, do I prefer movie 1 or movie 2?

Brafman and Domshlak (2009) describe the following three different semantics for the generalizing statement 'assignment s is weakly preferred to assignment t from p'. These three semantics are in Table 3.1 and have been adapted for assignments.

In the totalitarian semantics the statement encodes the most comparisons, in the *ceteris paribus*[2] semantics it encodes the least comparisons, and in the defeasible semantics the

[2] *Ceteris paribus* means 'all else being equal'.

Table 3.1: *Different semantics for the generalizing value statement 'assignment s is weakly preferred to assignment t from perspective p'*

SEMANTICS	MEANING
TOTALITARIAN	if assignment s' satisfies assignment s and assignment t' satisfies assignment t, then s' is weakly preferred to t' from perspective p
DEFEASIBLE	if assignment s' satisfies assignment s and assignment t' satisfies assignment t, then *presumably* s' is weakly preferred to t' from perspective p
CETERIS PARIBUS	if assignment s' satisfies assignment s, assignment t' satisfies assignment t and s' and t' assign the same attribute values to all attributes not in s or t, then s' is weakly preferred to t' from perspective p

number of comparisons the statement encodes is in between totalitarian and *ceteris paribus* semantics. More specifically, the set of comparisons of the *ceteris paribus* semantics is a subset of the set of comparisons of the totalitarian semantics. By using the totalitarian semantics, inconsistent preferences quickly arise. Consider Example 3.4. Using totalitarian semantics, both $1 < 2$ and $2 < 1$ are interpreted, which is an inconsistent preference ordering. The *ceteris paribus* semantics is the safest semantics, but it is likely that there is information in the statements that is not used. Using *ceteris paribus* semantics in Example 3.4, nothing can be inferred about my preference between 1 and 2 from the preference statements.

Using defeasible semantics seems like a good approach to overcome the strictness of totalitarian semantics and the weakness of *ceteris paribus* semantics. Defeasible semantics require to specify what *tends to be preferred* means.

Recall from Section 3.1 that an assignment s' satisfies assignment s if s' makes all the attribute-assignments that s makes and possibly more. For example, the assignment 'genre is comedy and length is 100 minutes' satisfies the assignment 'genre is drama'. If assignment s' satisfies assignment s, assignment t' satisfies t, and s is preferred to t from perspective p, then s' should also be preferred to t'. The following argument scheme captures this intuition.

Argument Scheme 5: Generalizing strict preference

Assignment s is strictly preferred to assignment t from perspective p,
assignment s' satisfies assignment s,
assignment t' satisfies assignment t,

---

therefore, presumably s' is strictly preferred to t' from perspective p.

For example, if the assignment 'genre is comedy' is preferred to the assignment 'genre is drama' from the perspective of fun, then the intuition is that an assignment 'genre is comedy and length is 100 minutes' should also be preferred to the assignment 'genre is drama'. It is possible that there are exceptions to this argument scheme, e.g., 'genre is comedy and length is 500 minutes' is not preferred to 'genre is drama' from the perspective of fun, and therefore, the conclusion of this scheme contains 'presumably'.

Using this argument scheme, arguments can be constructed that have conflicting conclusions, e.g., one argument concludes that s' is strictly preferred to t' from perspective p and another argument concludes that t' is strictly preferred to s' from p. For such situations of conflict, argumentation frameworks can be used to determine what conclusions are justified. The reader is referred to Subsection 2.1.4 for details on argumentation frameworks.

Note that because the conclusion of Argument Scheme 5 contains the word 'presumably', this argument scheme is defeasible. This means that other arguments can rebut the conclusion of an argument using this argument scheme, which allows explicitly modeling exceptions. Namely, if the knowledge base contains that s' is not strictly preferred to t' from p, then an atomic argument can be constructed concluding this. This atomic argument and the argument applying this argument scheme then attack each other. Since atomic arguments are stronger than non-atomic arguments, the atomic argument defeats the non-atomic argument.

Example 3.5 (Continuing Example 3.4) Recall that the genre drama is strictly preferred to the genre comedy from my perspective and the actor Kevin Spacey is strictly preferred to the actor Jim Carey from my perspective. Furthermore, movie 1 is a drama movie in which Jim Carey acts, whereas movie 2 is a comedy in which Kevin Spacey acts.

Let assignment s 'genre is drama and lead actor is Jim Carey' be the outcome of watching movie 1 and assignment t 'genre is comedy and lead actor is Kevin Spacey' be the outcome of watching movie 2. Because I strictly prefer the genre drama to comedy, the following argument A can be constructed that applies Argument Scheme 5.

'genre is drama' is strictly preferred to 'genre is comedy' from my perspective,
assignment s satisfies 'genre is drama',
assignment t satisfies 'genre is comedy',

---

therefore, presumably s is strictly preferred to t from my perspective.

In a similar fashion, the following argument B can be constructed concluding that assignment t is strictly preferred to assignment s because K. Spacey is strictly preferred to J. Carey.

'lead is K. Spacey' is strictly preferred to 'lead is J. Carey' from my perspective,
assignment t satisfies 'lead is K. Spacey',
assignment s satisfies 'lead is J. Carey',

---

therefore, presumably t is strictly preferred to s from my perspective.

Because of the properties of strict preference as described in Subsection 3.2.1, the conclusions of A and B are contradictory and therefore A and B attack each other. Assuming that arguments A and B are equally strong, both arguments are defensible, but neither argument is justified, so no justified conclusions can be drawn concerning what assignment is preferred from my perspective.

If no such tradeoffs occur between a set of outcomes, defeasible semantics makes the same interpretations as totalitarian semantics. If there are conflicts, then defeasible semantics makes less interpretations than totalitarian semantics, but the interpretations that are made are consistent. Furthermore, all interpretations made by *ceteris paribus* semantics are also made by defeasible semantics because if two outcomes have the same attribute values, then no conflicting arguments can be constructed.

Example 3.6 (Continuing Example 3.4) Let movie 3 be a drama movie featuring Kevin Spacey and assignment u be the outcome of watching movie 3. Notice that movie 2 and 3 are the same except for their genre. Because movie 3 is a drama and movie 2 a comedy and because I prefer drama, the following argument can be constructed using Argument Scheme 5.

'genre is drama' is strictly preferred to 'genre is comedy' from my perspective,
assignment u satisfies 'genre is drama',
assignment t satisfies 'genre is comedy',

---

therefore, presumably u is strictly preferred to t from my perspective.

Note that the arguments constructed in the example interpret my preferences the same as would have been done using *ceteris paribus* semantics.

Attributes Measuring Value From A Perspective

Value from a perspective may depend only on a subset of all the attributes available. For example, for the perspective of 'fun', the attributes 'genre' and 'lead actor' matter, but the attributes 'length' and 'release date' do not. If value from a perspective p only depends on the attributes in the set X, then we say that *set of attributes X measures value from perspective p*.

If value from a perspective p does not depend on some attribute x, then the value from perspective p is not influenced in any way by what attribute values are assigned on x. Consequently, if two assignments assign the same attribute values on all the attributes that matter for value from perspective p, then it does not matter how much they differ in the other attributes, they should have an equal amount of value from perspective p. To ensure this, the following constraint must be satisfied.

Constraint 1: Dependent Attributes

Value from perspective p is measured by the set of attributes X, and
assignments s and t assign the same attribute values on each attribute in X
if and only if
s and t are equally preferred from perspective p.

Note that this constraint resembles the notion of preferential independence (see Definition 2.46), which states that a set of attributes X is preferentially independent from a set of attributes Y iff preferences over two outcomes with different values assigned on X but with the same values assigned on Y only depends on the values assigned on X and not on what values are assigned on Y.

Example 3.7 (Movies) Suppose that length, genre and lead actor are all attributes by which outcomes of watching a movie can be described. Furthermore, I do not care about how long a movie is and therefore value from my perspective is measured by the attributes 'genre' and 'lead actor', but not by 'length'.

It is possible that different attributes can be used to measure value from a perspective. For example, the perspective 'proximity to the city centre' can be measured by the attribute 'distance by foot in m', 'distance by public transport in m', 'distance as the crow flies in m' or 'time in minutes required to reach the city centre'. Depending on the situation, one attribute may be more accurate than another. For example, if the agent always walks to the city centre, then the attribute 'distance by foot in m' is more accurate than the attribute 'distance as the crow flies in m', even though the latter may be good estimation.

Example 3.8 (Accuracy of attributes) Suppose there are two attributes x and y that can be used to measure value from perspective p. Argument A concludes that outcome 1 is preferred to outcome 2 because the x-value of outcome 1 is preferred to the x-value of outcome 2 from perspective p. Furthermore, argument B concludes that outcome 2 is preferred to outcome 1 because the y-value of outcome 2 is preferred to the y-value of outcome 1 from perspective p. The conclusions of argument A and B conflict and consequently they attack each other. However, if attribute x is more accurate than attribute y for the measurement of value from perspective p, then argument A should be stronger than argument B and thus A should defeat B.

Although one attribute can be more accurate than another, the less accurate attribute can still be useful because the more accurate attribute values of outcomes may be unknown or expensive to find out.

Assume that attribute x is more accurate than attribute y for measuring value from perspective p. Also, alternative 1 results in an y-value that is preferred to the y-value in which alternative 2 results. Furthermore, alternative 1 results in x-value 1, but it is not known in which x-value alternative 2 results. Consequently, attribute x cannot be used to compare alternative 1 and 2, but attribute y can. Even though it is better to compare the outcomes using attribute x, comparing them using y is better than nothing.

The problem of justifying why one attribute is better suitable to measure value from a perspective is difficult and we will not address this in this thesis. As future work it is suggested to construct argument schemes to argue about what set of attributes is the most appropriate to measure value from a perspective. A starting point for this could be Keeney and Raiffa (1976), where a number of criteria are described to select the most appropriate attributes. Another aspect that needs to be considered is that attributes may be equivalent in the sense that there may exist a function that transforms one attribute into another. For example, the attribute 'price in Dollar' could be transformed into the attribute 'price in Euro'.

Errors

To support a user in making the best decision, the value statements the user makes should not be taken for granted, but should be critically examined. Especially generalizing value statements are prone to error. There are several reasons why the user may make errors. One possible kind of error is that the generalizations which users make are too strong or not based on sufficient or biased evidence. For example, a user may express the generalizing statement 'I prefer Kevin Spacey to Jim Carey' on the basis of watching one movie with Kevin Spacey and one movie with Jim Carey. This is not a strong reason to conclude that the user in general prefers Kevin Spacey to Jim Carey because the sample is very small. Because of the small sample, it is likely that the difference in preference is caused by chance. For example, the movie featuring Jim Carey is considered to be the worst movie he acted in, whereas the movie featuring Kevin Spacey is considered to be the best movie he acted in.

Another reason is that the user may have made wrong generalizations. For example, the user previously watched movie 1, lasting 120 minutes, and movie 2, lasting 130 minutes. Because the user preferred movie 1 to movie 2, the user now generalizes that he prefers movies that last 120 minutes to movies that last 130 minutes. However, in this case, the attribute duration is not the cause for the user preferring movie 1 to movie 2. By asking the

user for justification of a generalizing value statement, such errors can be detected and the system could argue that the user's generalization is not accurate.

3.3 Value Trees

If an agent is unfamiliar with some outcome, then he may not be able to judge how the value of that outcome compares to the value of other outcomes from a perspective. In that case, the agent could use argumentation to reason about how these outcomes compare in value from that perspective. In an opposite case, an agent may need to justify or explain to another agent how two outcomes compare in value from a perspective. In both cases, it is useful to be able to justify why an outcome has a certain value from a perspective and argumentation is particularly useful for this.

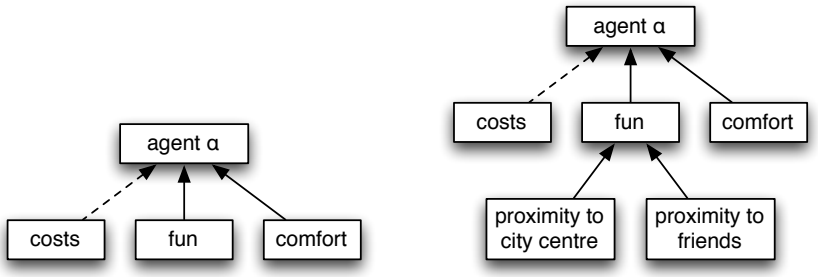
In such situations, it is often useful to structure these values in the form of a so-called *value tree* (Edwards, 1977; von Winterfeldt and Edwards, 1986), which is closely related to an 'objective hierarchy' (Keeney, 1992; Keeney and Raiffa, 1976) and to a 'decision hierarchy' (Saaty, 1986, 2008). A value tree hierarchically relates general areas of concern, intermediate objectives, and specific evaluation criteria defined on attributes. The purpose of a value tree is to explicate and operationalize higher level values. Because general areas of concern, intermediate objectives and specific evaluation criteria express value, each of them can be used to compare elements, which means that they each can be seen as perspectives.

Because an agent's preferences, general areas of concerns, intermediate objectives, and specific evaluation criteria can all be seen as perspectives, we will talk about decomposing a perspective p into other perspectives that influence p. In other words, instead of decomposing an agent's preferences into a number of general concerns, we will say that the perspective of an agent is decomposed into a number of perspectives that represent the agent's general areas of concerns. Similarly, instead of decomposing an intermediate objective into a number of specific evaluation criteria, we will decompose the perspective of an intermediate objective into a number of perspectives that represent specific evaluation criteria.

We thus talk about the value tree for a perspective. Assume that in the value tree for perspective p, perspective p is decomposed into perspectives q and r, but that q and r are both still too general. In that case, value trees could be constructed for both q and r. The value tree of perspective p thus contains other value trees of perspectives that influence p. If a specific evaluation criterion can be associated to a perspective p, then perspective p does not need to be decomposed further and perspective p is called a leaf node in the value true.

Example 3.9 (Decomposing an Agent's Preferences) Suppose that agent α is buying a house and does not know what house he prefers most. However, agent α does know that he cares about minimizing costs, maximizing fun, and maximizing comfort. This information can be represented by decomposing perspective α into the perspectives costs, fun, and comfort. The more costs, the less α prefers it, i.e., costs negatively influence α's preferences. The more fun, the more α prefers it, i.e., fun positively influences α, and so does comfort. Figure 3.1a sketches how α's perspective is decomposed. A box is a perspective and a normal arrow denotes positive influence and a dotted arrow denotes negative influence.

Ideally, a user is able to decompose a perspective p into all the perspectives that are important for p. Moreover, the user is able to determine value from p given value from all



(a) *Decomposition of α's perspective* **(b)** *Decomposition of fun included*

Figure 3.1: *Visualization of Decomposition in Example 3.9*

the perspectives that influence p, i.e., he is able to infer value upwards in the value tree. In multi-attribute utility theory for example, a value function could be calculated by adding the value functions of all attributes assuming that the attributes are additive independent.

There is a tradeoff between expressiveness and difficulty of elicitation. A fully quantified value function induces a complete ordering over elements, which is useful for decision making. However, obtaining a fully specified value function is difficult and sometimes impossible. In contrast, one could obtain what an agent values in an easy way using statements that users can express easily. In this case however, the values of the user are not fully specified, which results in the fact that some elements cannot be compared in value.

The main research question in the thesis concerns complex situations where users typically find it hard or are not able to specify a quantified value function. For this purpose, we introduce two general relations between perspectives that are easy to express for users: *positive influence* and *negative influence*. These two relations are based on the direction of how a perspective influences another the perspective, e.g., the perspective costs negatively influences the perspective of an agent. Because these relations express something that is typically true, e.g., in some cases a higher cost may be preferred by the agent, it must be possible to model exceptions.

3.3.1 Influence Between Perspectives

If *perspective p positively influences perspective q*, then p is an aspect of q and the more preferred an element is from perspective p, the more preferred it tends to be from perspective q. In other words, if p positively influences q, then 'element 1 is preferred to element 2 from p' is a reason to believe that 'element 1 is preferred to element 2 from perspective q' is true. The following argument scheme informally describes this intended meaning of how the influence between perspectives should be used to infer how elements compare in value.

Argument Scheme 6: Positive Influence
Perspective p positively influences perspective q,
element 1 is strictly preferred to element 2 from p,

therefore, presumably element 1 is strictly preferred to element 2 from q.

The conclusion of this argument scheme may not always be true for two reasons. Firstly, there may be other perspectives that influence perspective q from which it can be inferred that the elements are preferred differently from q. Secondly, because it is difficult to decompose a perspective, influence is something that holds in general. Therefore, there may exceptional situations where the conclusion of this argument scheme does not hold.

In contrast, if *perspective p negatively influences perspective q*, then p is an aspect of q such that the more preferred an element is from perspective p, the *less* preferred it tends to be from perspective q. In other words, if p negatively influences q, then 'element 1 is preferred to element 2 from p' is a reason to believe that 'element 2 is preferred to element 1 from perspective q'. Negative influence can also be used to argue about what outcome is preferred from a perspective. The following argument describes how to reason with negative influence.

Argument Scheme 7: Negative Influence

Perspective p negatively influences perspective q,
element 1 is strictly preferred to element 2 from p,

---

therefore, presumably element 2 is strictly preferred to element 1 from q.

For the same reasons as for positive influence, the conclusion of this argument scheme may not always be true. Therefore, this argument scheme must be modeled with a defeasible inference rule.

Assume that two elements have an equal amount of value from some perspective p, or in other words are equally preferred, and that p either positively or negatively influences perspective q. Then that is a reason to say that they are also equally preferred from perspective q. For example, if the perspective 'costs' negatively influence my preferences, then if two products cost the same amount, then that is a reason to infer that I should prefer both products equally. The following argument scheme describes this intuition. Again for the same reasons this scheme must be defeasible.

Argument Scheme 8: Equal Value and Influence

Perspective p influences perspective q,
element 1 and element 2 are equally preferred from p,

---

therefore, presumably element 1 and element 2 are equally preferred from q.

Perspectives have been given different names in the literature. If a perspective p influences the perspective of an agent α, then perspective p can be seen as a criterion that agent α use for making decisions. Perspective p could also be seen as a *point of view* in Steedman and Krause (1986), or a *measure of effectiveness / performance* in Keeney (1992). Note that there may also be perspectives that does not influence the perspective of an agent. In that case, the agent does not care about that perspective and does not use it as a criterion in making decisions.

More specific means easier to specify preferences

When determining the value tree of an agent, the question arises of how far or deep a value tree should be. In other words, how deep should an agent's perspective be decomposed? The purpose of decomposing a perspective is to identify and operationalize those aspects that matter to the perspective. A perspective is operational if either it is measured by an attribute

or if it is decomposed into a number of operational perspectives. An agent's perspective should be decomposed until it has been made operational.

On the one hand, one could only look at the agent's perspective and how attribute values compare from the agent's perspective. For example, only use value statements such as 'assignment 1 is strictly preferred to assignment 2 from the perspective of agent α'. Tradeoffs can now only be resolved at α's perspective. The advantage of decomposing α's perspective into several more concrete perspectives is that tradeoffs can be resolved at lower perspectives that are more concrete and therefore it is easier to resolve a conflict, i.e., it becomes easier to make a tradeoff.

The perspectives that influence a perspective q should never be more abstract than p. This also means that if p influences q, then value from p is measured by as much attributes as or fewer attributes than value from q. For example, the preferences of an agent concerning houses depend on all kinds of attributes such as size, price, location, and facilities. The perspective of the agent could be decomposed into perspectives likes 'costs', 'fun', and 'comfort'. For the perspective of costs only the attribute price matters, for the perspective of fun, both the location and facilities matter, and for the perspective of comfort only the facilities matter. Each of these perspectives depends on fewer attributes than the perspective of the agent.

However, because the perspective of the agent is decomposed in all these other perspectives, all the attributes that measure value from these perspectives influence what is valued from the agent's perspective. In other words, if a perspective p is measured by a set X of attributes and influences perspective q, then value from perspective q also depends on the attributes in X. The following constraint describes this intuition.

Constraint 2: Attributes Influence
Value from perspective p is measured by the set of attributes X, and
perspective p influences perspective q
if and only if
value from perspective q is measured by X and possibly more attributes.

In general, the fewer attributes that matter for a perspective, i.e., the fewer attributes measure value from a perspective, the easier it is to determine value from that perspective. Because each of the perspectives that influence a perspective p is measured by fewer attributes than p does, it is easier to determine value from those perspectives than it is to determine value from p. Because the leaf nodes in the value tree are not influenced by any perspective, they depend on the least amount of attributes. The root node of the value tree on the other hand, depends on all the attributes that measure value from the perspectives of the leaf nodes.

Using how elements compare from the leaf nodes in the value tree, value is inferred 'upwards' towards the root node. Because the intermediate and higher perspectives in the value tree 'aggregate' value from the lower perspectives, it is possible that tradeoffs arise where it is not clear how two assignments compare from a perspective.

How to use influence

The positive and negative influence relations between perspectives can be used to decompose a perspective into the various aspects it involves. Decomposing a perspectives had two main advantages. Firstly, it explicates what that perspective means to an agent. This is particularly

useful when communicating about a perspective with a common name or with subjective perspectives such as an agent's preferences or what is fun for an agent. For example, a perspective such as justice or safety has different meanings for different agents, but if an agent explicates what a particular perspective means for him, then another agent can understand and possibly support him in his preferences.

Secondly, by decomposing a perspective p into the perspectives that influence p, agents can reason about how elements compare from perspective p using how they compare from the perspective that influence p. If an agent has reasoned about why one element is preferred from a perspective, then the agent can use this line of reasoning to explain or justify this preference. Furthermore, decomposing perspective p makes explicit what aspects an agent uses to determine how elements compare from perspective p. Agents can then discuss whether any important aspect has been forgotten, or whether the agent made mistakes in his line of reasoning.

Example 3.10 (Continuing Example 3.9) Assume that outcome 1 is the result of agent α buying a house in the centre and outcome 2 is the result of α buying a house in a suburb. From the perspective of location, outcome 1 is strictly preferred to outcome 2. Using the argument scheme from positive influence, the following argument A can be constructed.

The perspective of location positively influences agent α's preferences,
outcome 1 is strictly preferred to outcome 2 from the perspective of location,

---

therefore, outcome 1 is strictly preferred to outcome 2 from the perspective of α.

Because the house in the suburb is big and therefore costs more, outcome 2 is strictly preferred to outcome 1 from the perspective of costs. Using the argument scheme from negative influence, the following argument B can be constructed.

The perspective of costs negatively influences agent α's preferences,
outcome 2 is strictly preferred to outcome 1 from the perspective of costs,

---

therefore, outcome 1 is strictly preferred to outcome 2 from the perspective of α.

Note that the argument A and B have the same conclusion and therefore do not conflict. However, the house in the suburb is more comfortable and agent α finds comfort important. Using the argument scheme of positive influence, the following argument C can be constructed.

The perspective of comfort positively influences agent α's preferences,
outcome 2 is strictly preferred to outcome 1 from the perspective of comfort,

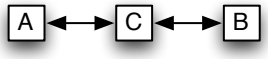
---

therefore, outcome 2 is strictly preferred to outcome 1 from the perspective of α.

Argument C's conclusion conflicts with both argument A's and argument B's conclusions. The attacks between A, B, and C can be visualized as in Figure 3.2. We now have two arguments that conclude that agent α should prefer outcome 1 and one argument concluding that outcome 2 should be preferred.

A tradeoff is a situation that involves losing one quality or aspect of something in return for gaining another quality or aspect. Assume that one argument concludes that agent α should prefer outcome 1 to outcome 2 because outcome 1 is better from perspective p and another argument argues that agent α should prefer outcome 2 because outcome 2 is better from

Figure 3.2: *The attacks between arguments A, B and C in Example 3.10*



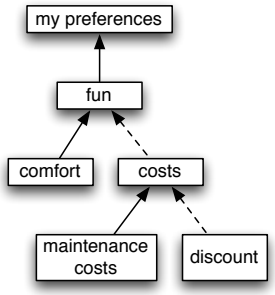


Figure 3.3: *Influence between perspectives in Example 3.11. The perspective of fun positively influences my perspective, i.e., the more fun, the more I prefer it. The perspective of costs negatively influences the perspective of fun, i.e., the more costs, the less fun, but the perspective of comfort positively influences fun. Also, consider the perspective of 'maintenance costs' denoting that element 1 is strictly preferred to element 2 from the perspective of maintenance costs if element 1 has higher maintenance costs than element 2. The perspective of maintenance positively influences the perspective of costs, i.e., the more maintenance costs, the more costs. Finally, the perspective of discount negatively influences the perspective of costs, i.e., the more discount, the less costs.*

perspective q. To choose an outcome, agent α has to choose between an outcome better with respect to perspective p and an outcome better with respect to perspective q. In other words, in such a situation an agent has to make a tradeoff between the various perspectives that are important to him. One way to make a tradeoff is to look at the importance of perspectives.

3.3.2 Transitivity of Influence

Because we distinguish two kinds of influence, i.e., positive and negative influence, there are four combinations of influences: positive positive, positive negative, negative positive, and negative negative. Transitivity of influence can be used to explain and justify influence between perspectives, which is useful when supporting constructing a value tree. The following example is used to provide an intuition for how transitivity between different kinds of influence is defined.

Example 3.11 Assume I am buying a house. I should choose the house that I prefer most, i.e., the house that is most preferred from my perspective. Figure 3.3 illustrates how my perspective is decomposed by showing the perspectives that influence my perspective.

Consider Example 3.11. Comfort positively influences fun and fun positively influences my preferences. If this is the case, then comfort positively influences my preferences. The following constraint ensures this desired behavior and is called Positive-Positive (PP) Transitivity.

Constraint 3: PP Transitivity

If perspective p positively influences perspective q and
perspective q positively influences perspective r
then perspective p positively influences perspective r.

In Example 3.11, costs negatively influence fun and fun positively influences my preferences. Costs should negatively influence my preferences. The following constraint makes

sure this is always the case and is called Negative-Positive (NP) Transitivity.

Constraint 4: NP Transitivity
If perspective p negatively influences perspective q and
perspective q positively influences perspective r
then perspective p negatively influences perspective r.

Similarly, the Positive-Negative (PN) Transitivity concerns a positive and negative influence. Consider in Example 3.11 that maintenance costs positively influence costs and that costs negatively influence fun. Because maintenance costs should negatively influence fun, we have the following constraint.

Constraint 5: PN Transitivity
If perspective p positively influences perspective q and
perspective q negatively influences perspective r
then perspective p negatively influences perspective r.

Finally, Negative-Negative (NN) Transitivity combines negative influence with negative influence. In Example 3.11, discount negatively influences costs, and costs negatively influence fun. Because discount should positively influence fun, the NN Transitivity constraint is defined as follows.

Constraint 6: NN Transitivity
If perspective p negatively influences perspective q and
perspective q negatively influences perspective r
then perspective p positively influences perspective r.

When supporting a (human) agent in making a decision, it can be useful to help him construct a value tree. When constructing a value tree, a user may state how some perspective p influences his preferences. It is possible that p indirectly influences his preferences. In that case, the user could be asked why p influences his preferences. The constraints concerning the transitivity of influences can be used to justify a certain influence. For example, when buying a house, the user may state that the perspective of 'distance to city centre' negatively influences his preferences (i.e., the more distance, the less he prefers it). The decision support system could then ask the reason behind this influence, which the user can answer by stating that 'distance to the city centre' positively influences the perspective 'time to city centre', which negatively influences his preferences.

3.3.3 Importance

Not all aspects of a perspective may be equally important for a perspective or in other words, the perspectives that influence a perspective may differ in importance. For example, suppose agent α's perspective is decomposed into costs, fun, and comfort, i.e., α cares about costs, fun, and comfort. However, agent α is poor and therefore, minimizing costs is more important than fun. Similarly, for the perspective of fun α may find proximity to friends more important than proximity to the centre. This thesis focuses on the following statements concerning the importance of perspectives.

- *"Perspective p is more important than perspective q for perspective r"*

- *"Perspective p is equally important as perspective q for perspective r"*

A perspective p is only important for another perspective q if p influences q either positively or negatively. If two perspectives p and p' both are unimportant for perspective q, then p and p' are equally important for q. The following example shows the intuition behind why this section proposes to use the importance of arguments to determine the strength of arguments.

Example 3.12 (Importance of Perspectives) Assume that agent α's perspective is decomposed into costs and fun and that α finds costs more important than fun. Furthermore, let outcome 1 be the result of buying house 1 and outcome 2 be the result of buying house 2.

Agent α starts reasoning about what outcome he prefers most and constructs the following arguments: argument A concludes that α strictly prefers outcome 1 to outcome 2 because outcome 1 costs less than outcome 2, and argument B concludes that α strictly prefers outcome 2 to outcome 1 because outcome 2 is more fun than outcome 1. Argument A and B have conflicting conclusions and therefore attack each other.

However, argument A should be a stronger argument than argument B because agent α finds costs more important than fun. If A is stronger than B, A defeats B resulting in that A becomes a justified argument and that α can conclude that he should strictly prefer outcome 1 to outcome 2.

The strict importance relation should be *irreflexive* because no perspective should be more important than itself for any perspective. Furthermore, the importance relation should also be *antisymmetric*, i.e., if perspective p is more important than perspective q for perspective r, then perspective q is not more important than perspective p for perspective r. We assume that importance is *transitive*, i.e., if p_3 is more important than p_2 for q and p_2 is more important than p_1 for q, then p_3 is more important than p_1 for q.

If a perspective p does not influence another perspective r, then p is not important for r at all. A perspective q that does influence perspective r is therefore always more important than p for r. In other words, if p does not influence r and q does influence r, then q is more important than p for r.

The consequence of that perspectives differ in importance is that the presumptions of applying Argument Scheme 6 and 7 differ in conclusive force / strength. By formalizing the importance of perspectives on a meta-level, it can be used to argue about what application of an argument scheme has more conclusive force, which is an interesting aspect to consider when determining the strength of an argument. Because there may be other reasons why one application of an argument scheme is stronger than another application, this idea is captured in the following defeasible argument scheme.

Argument Scheme 9: Importance

Perspective p is more important than perspective q for perspective r,

---

therefore, presumably inferring value from p to r is stronger than inferring value from q to r.

Note that because it refers to the application of object-level argument scheme, this is a meta-level argument scheme. In Section 5.2 we will formalize this scheme as a defeasible rule in a meta-level argumentation system.

Making Importance Precise

As argued by (Keeney, 1992, p. 147), it is difficult to interpret what it means that one perspective is more important than another. For example, assume agent α is buying a house and finds the perspective of costs more important than the perspective of location. Does this mean that agent α prefers a house that is 1 euro cheaper but in a horrible location to a house that is 1 euro more expensive but has the best possible location?

Our conceptual framework for value only allows making ordinal comparisons between elements. It is not possible to express that one element is only a little bit preferred to another from one perspective, but much more preferred from another perspective. Adding a notion of distance is left as future work, but we will sketch an approach how the notion of distance can be incorporated together with the notion of importance. The following argument scheme infers that one application of an argument scheme is stronger than another application because the difference in value is significantly bigger. Note that the conclusion is not necessarily true because Argument Scheme 9 may infer a conflicting conclusion. Therefore, the following scheme is defeasible.

Argument Scheme 10: Significant Difference
Assignments s and t differ significantly more in value from perspective p than they do from perspective q,
both p and q influence perspective r,

---

therefore, presumably inferring value from p to r is stronger than inferring value from q to r.

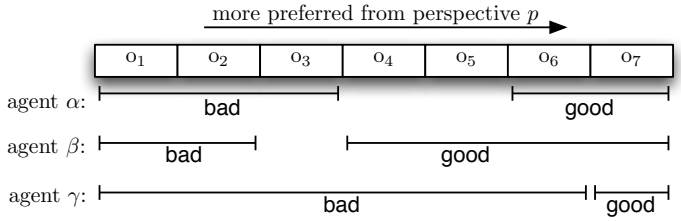
It is difficult to define what 'to differ significantly' is in a qualitative manner. Therefore, this argument scheme will not be used in the rest of this thesis but is left as future work.

3.4 Monadic / Classificatory Value

In informal discourse about what a person values, terms like 'good', 'bad', 'satisfactory', 'best', 'worst', 'beautiful', and 'cheap' are commonly used. A term like 'good' or 'bad' is a predicate that can be given to an alternative, an outcome, or an attribute of an outcome. Such predicates are called *monadic value predicates* (Hansson, 2001). Monadic value predicates can be used to evaluate alternatives, outcomes, and attribute values of outcomes. Evaluations by means of monadic value predicates are subjective. Namely, what one agent finds good, another finds bad. Moreover, evaluations are done from a perspective. For example, a car may be good from the perspective of speed, but bad from the perspective of costs.

Example 3.13 (Subjective Evaluation) Assume there are the following outcomes $\Omega = \{o_1, \ldots, o_7\}$ and that outcome o_i is preferred to outcome o_j from perspective p if $i > j$. Agents α, β, and γ evaluate the outcomes. Agent α evaluates outcomes o_6, and o_7 as 'good' and o_1, o_2, o_3 as 'bad', whereas agent β is easier to please and finds just o_1, and o_2 'bad' and o_4, o_5, o_6, o_7 'good'. In contrast, agent γ is very hard to please and evaluates only o_7 as good from perspective p and the other outcomes as 'bad'. Figure 3.4 visualizes how α, β, and γ evaluate the different outcomes.

Figure 3.4: *Perspective on outcomes from Example 3.13*



Many monadic value predicates are used in daily life, e.g., good, excellent, great, awesome, legendary, beautiful, cheap, fair, unfair, fast, and so on. The most general monadic value predicates are 'good', 'bad', 'best' and 'worst'. Some value predicates can only be used in combination with certain perspectives. For example, 'beautiful', and 'ugly' are predicates for the perspective of esthetics, 'fair' and 'unfair' from the perspective of justice, and 'fast' and 'slow' from the perspective of speed. Because this thesis explicitly represents the perspective from which a monadic value statement is made, we will focus on general value predicates such as 'good' and 'bad'. The translation of a statement like 'x is good from the perspective of esthetics' to the statement 'x is beautiful' and vice versa is left as future work.

For our purposes it is sufficient to focus just on 'good' and 'bad'. Hansson (2001) further distinguishes statements like 'fairly good', 'very bad' and so on. Such extensions are left for future work.

3.4.1 Properties of Monadic Value Predicates

Exclusionary value predicates

If monadic value predicates m and n are *exclusionary*, then no element can be evaluated as both m and n from any perspective. This means that the following constraint must be satisfied if two value predicates are exclusionary.

Constraint 7: Exclusionary Value Predicates
If monadic value predicate m is exclusionary with monadic value predicate n,
agent α evaluates assignment s as m from perspective p,
Then agent α should not evaluate assignment s as n from perspective p.

The monadic value predicates 'good' and 'bad' are exclusionary. Namely, if agent α evaluates element 1 as 'good' from some perspective p, then agent α should not also evaluate element 1 as 'bad' from the same perspective p. Note that if agent α evaluates an element as 'good' from perspective p, that he may evaluate the same element as 'bad' from some other perspective. Also note that if agent α evaluates an element as 'good' from perspective p, then another agent may evaluate that element as 'bad' from the same perspective p.

Positive and negative value predicates

What is better than good is itself good. The monadic value predicate 'good' has the property of being $\leq$-positive (Hansson, 2001). This means that if an agent evaluates element 1 as 'good' from perspective p, then all elements that are preferred to element 1 from perspective

p should also be evaluated as 'good' from perspective p. Similarly, what is worse than bad is itself bad. The monadic value predicate 'bad' has the property of being $\leq$-negative. This means that if an agent evaluates element 1 as 'bad' from perspective p, then all elements to which element 1 is preferred from p should also be evaluated as 'bad' from perspective p.

A first formalization could be: if the monadic value predicate m is positive and agent α classifies assignment s as m from perspective p, then any assignment s' that is preferred to s from p is also classified as m. For example, the monadic value predicate 'good' is positive and agent α classifies a profit of 1 million as 'good' from the perspective of profit, then any higher profit is also good. However, this formalization is incorrect because it does not consider how the perspective influences α's preferences. For example, let the perspective of 'costs' negatively influence α's preferences. Given the previous formalization, if 'good' is positive and α finds 1 million dollar good from the perspective of costs, then any cost higher than 1 million is also good. In that case, if the agent finds costs of 1 million 'good', he should also find costs of 100 billion 'good'. Clearly, how the perspective influences the agent's preferences should be taken into account.

If a monadic value predicate is positive, then the following constraint should be satisfied.

Constraint 8: Positive Value Predicate
If monadic value predicate m is positive,
agent α evaluates element 1 as m from perspective p,
perspective p positively influences α's preferences, and
element 2 is weakly preferred to element 1 from p,
Then agent α should evaluate element 2 as m from p.

Similarly, when an agent evaluates an element from a perspective p that negatively influences his preferences, then assignments that have less value from perspective p should be evaluated the same. For example, if the perspective 'costs' negatively influences agent α's perspective and α evaluates costs of 1 million as 'good', then α should evaluate costs less than a million as 'good' as well.

If a monadic value predicate is negative, then the following constraint should be satisfied.

Constraint 9: Negative Value Predicate
If monadic value predicate m is negative,
agent α evaluates element 1 as m from perspective p,
perspective p positively influences α's preferences, and
element 1 is weakly preferred to element 2 from p,
Then agent α should evaluate element 2 as m from p.

Note that these constraints can be used to reason about, justify and refute monadic evaluations. For example, given that the conditions of Constraint 9 are true, then the constraint can be used to justify that the agent should evaluate element 2 as m from p.

Continuous

Another property that Hansson considers is whether a monadic value predicate is *continuous*. If a monadic value predicate m is continuous, then the following constraint must be satisfied.

Constraint 10: Continuous monadic value predicates
If monadic value predicate m is continuous,

agent α evaluates assignments s and u as m from perspective p and
assignment t is in between s and t from perspective p,
Then agent α should evaluate assignment t as m from perspective p.

The value predicates 'good' and 'bad' are both continuous. For example, if elements 1 and 3 are good and element 2 is in-between 1 and 3, i.e., 2 is better than 1 but worse than 3, then element 2 is also good.

3.4.2 Justifying Monadic Value

In this section we discuss two approaches to justify why an element is given a certain monadic value predicate. The first approach uses the properties of a value predicate and the predicate given to another element to justify what predicate another element obtains. For example, because the monadic value predicate 'good' is so-called $\leq$-positive, element 1 is 'good' from perspective p and element 2 is preferred to element 1 from perspective p, element 2 should also be evaluated as 'good' from the perspective p. Properties such as $\leq$-positive, $\leq$-negative, $\leq$-continuous, and exclusivity can be used to justify an element obtaining a certain monadic value predicate.

The second approach justifies an element obtaining a certain monadic value predicate because of how that element compares to a certain *reference element*. For example, if agent α uses element e as reference element and element 1 is preferred to the reference element e from perspective p, then agent α should evaluate element 1 as good from perspective p. If an element is worse than the reference element, it is evaluated as bad.

There are different reference elements which an agent can choose to justify a monadic value statement. Figure 3.5 schematically explains the following approaches to justify monadic value.

Doing Nothing The agent could use the outcome of the alternative of doing nothing as reference element. In that case, outcomes are evaluated as good if they result in something better than the outcome of doing nothing.

Compared To Others If the decision context is such that other agents have to deal with the same decision context, then an agent could use as reference element the element that other agents on average choose. Assume the decision context is the $100m$ sprint, then to evaluate if a certain outcome is good, one could look at how other agents perform. If on average they sprint $100m$ in x seconds, then you could say that less than x seconds is good, and more than x seconds is bad.

Within Own Powers If the agent has an idea about what alternatives he can perform and in what outcomes and attribute values those alternatives may result, then the agent could evaluate elements from perspective p by how good they are and how likely they will result. If it is likely that the agent will achieve a certain outcome and unlikely that something better is achieved, then the agent could say that better than a likely outcome is good and worse than a likely outcome is bad. For example, agent α runs $100m$ in y seconds on average, then less than y seconds is good and more is bad.

What reference element an agent should use in what decision context is difficult. Evaluating outcomes as 'good' if they are better than a typical outcome is more ambitious than

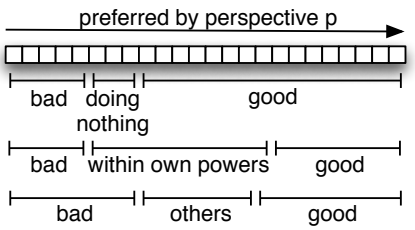


Figure 3.5: *Sketch of different approaches for evaluating the monadic value of outcomes. Outcomes are evaluated as 'good' if for the first they are better than the outcome of doing nothing, for the second if they are better than what is within the power of the agent, and for the third if they are better than what others are achieving.*

evaluating outcomes as 'good' if they are the outcome of doing nothing. To choose the level of ambition with respect to a perspective, an agent should look at how important he finds that perspective. If an agent does not find perspective p very important but does find perspective q very important, then he should be ambitious towards q and easy-going towards p.

Determining what reference element to use lies outside the scope of this thesis. Rather this thesis will assume that there is a reference element.

3.4.3 In Practical Reasoning

To deal with bounded resources in computation and memory, Simon (1955) proposes that agents should not try to find the best possible alternative, but that they should satisfice with an alternative that is satisfactory. If an agent tries to find an alternative that is satisfactory he can stop looking when he found one whereas if an agent tries to find the maximally preferred alternative, then he needs to consider every alternative.

Monadic value statements are particularly useful for an agent to justify and determine whether an alternative is satisfactory. If an agent assigns the monadic value predicate 'good' to an alternative, then that alternative is satisfactory. To infer whether an agent should evaluate an alternative as 'good' from his perspective, he could look at how he evaluates the possible outcomes of that alternative. Similarly, to infer whether an agent should evaluate an outcome as 'good' from his perspective, he could look at how he evaluates the attribute values of that outcome. Chapter 6 proposes how monadic value predicates can be used in practical reasoning. Namely, it is proposed that an agent can justify pursuing a goal to achieve certain attribute values if he evaluates those attribute values as 'good'.

Because the effects of alternatives are described in terms of attribute values, it is relatively easy for a planner or means-end reasoner to find an alternative that results in some given attribute values. Splitting attribute values into a good and bad part is a way to select an explicit set of attribute values that the agent wants to achieve. By interpreting attribute values evaluated as good as being satisfactory, monadic value statements are used to select a satisfactory level of achievement.

If an agent splits attribute values into a good and bad part and is committed to achieve a good attribute value, then we say that the agent has adopted the goal to achieve a good attribute value.

Example 3.14 (Continuing Example 3.13) Because agent α, β, and γ have different opinions about what outcome is 'good' from perspective p, they adopt different goals. Namely, agent α may adopt the goal to find an alternative that results in outcome o_5 or better; agent β to find an alternative resulting in o_3 or better, and agent γ adopts the goal to find an alternative

that results in o_7. For β it is the easiest to find an alternative that results in a good outcome and for γ it is the most difficult.

What attribute value is evaluated as 'good' is subjective. Several constraints can be given to make adopting a set of attribute values as a goal rational. For example, is it sensible for agents to adopt goals that are impossible to achieve? An ambitious agent may only evaluate the best achievable attribute value as 'good', even though such an attribute value may be difficult to achieve. Chapter 6 proposes how agents can justify adopting a goal.

3.5 Chapter Summary

This chapter has proposed a conceptual framework for value that is aimed to be used by artificial agents to determine what decision is the best. The underlying motivation is to use this in a decision support setting where an artificial agent supports a (human) agent in making a decision focusing on determining what is valued most. Because using arguments to justify and refute opinions is a natural and commonly used approach by people, we have taken an argumentation-based approach for decision support. This approach enables both the artificial agent and the human user to give arguments supporting or attacking a certain point of view concerning what decision is best.

The conceptual framework has proposed how an agent's preferences can be decomposed into the manageable aspects. The result of this decomposition is a value true, which is based on literature in decision analysis (Edwards, 1977; Keeney, 1992; Keeney and Raiffa, 1976; von Winterfeldt and Edwards, 1986). Furthermore, this chapter has proposed a number of argument schemes that can be used to argue about what an agent should prefer given a value tree of his perspective. Hansson (2001) assumes that preferences have already been constructed in a proper way (p.17: "the value statements under study have been subjected to sufficient (reflection-guided) adjustment to have attained consistency"). In Chapter 5 we will formalize the conceptual framework in the ASPIC+ framework. By constructing arguments and counterarguments and evaluating them in ASPIC+, the value statements under study are subjected to this sufficient adjustment and attain consistency.

Preference elicitation methods aim at efficiently obtaining a user's preferences (Brafman and Domshlak, 2009). In cases where a user does not know what to prefer, e.g., a novice who wants to buy a professional camera that does not know what features of a camera are important, preference elicitation techniques are insufficient because either the user does not know what to prefer or may express wrong preferences. By using argumentation in combination with the conceptual framework of this chapter, the decision maker's preferences can be discussed in a natural way, much like what a salesman would do in a shop where he would explain to the customer why some feature is more important than another and why the customer should thus pick one item. Moreover, where quantitative approaches can only justify a recommendation by showing the formulae and parameters used, the proposed approach can put forward arguments that justify a certain recommendation. Argumentation is much more natural and intuitive for people than showing the calculations of utility theory.

4 Perspective-Based Value Model

In this chapter, the PVM conceptual framework proposed in the previous chapter is formalized. In the PVM, the notions of an assignment and of a perspective are essential. These notions are formalized in Section 4.1 and Section 4.2 respectively. Next, the formalization of monadic evaluations is proposed in Section 4.3. To demonstrate the formalization that we introduce in this chapter, Section 4.4 applies it to the running example as described in the introduction. We end this chapter with a summary. The following chapter proposes an argumentation logic based on the formalization of this chapter. This allows us to construct arguments and counterarguments concerning the relative value of assignments.

4.1 Assignments

As explained in Section 3.1, we are interested in arguing about the value of assignments. To define assignments, the notions of 'attribute' and 'attribute value' as described in Section 2.4 are first introduced.

Definition 4.1 (Attributes and Attribute Values) *The set $\mathcal{A}$ denotes the set of all attributes and the set AV denotes the set of all attribute values any attribute can take. The function $\mathsf{dom} : \mathcal{A} \rightarrow 2^{\mathsf{AV}}$ maps an attribute to the set of attribute values it can take.*

Definition 4.2 (Assignment) An assignment s on a set of attributes $X \subseteq \mathcal{A}$ *is a partial function $s : \mathcal{A} \rightarrow \mathsf{AV}$ such that X is the domain of definition of s, denoted $\mathsf{dod}(s) = X$, and $s(x) \in \mathsf{dom}(x)$ for each $x \in X$. Furthermore, the set $\mathcal{S}_{\mathcal{A}}$ denotes the set of all assignments on all subsets of $\mathcal{A}$.*

If s is an assignment on the set of attributes $X \subset \mathcal{A}$, i.e., $\mathsf{dod}(s) = X$, and if attribute x is an attribute that is not in the domain of definition of s, i.e., $x \notin X$, then $s(x)$ is undefined. When an expression like $s(x) = t(x)$ is evaluated, then $s(x)$ and/or $t(x)$ being undefined results in the expression returning false.

We now introduce some notation. Since the set of attributes $\mathcal{A}$ will typically be fixed, we will write $\mathcal{S}$ instead of $\mathcal{S}_{\mathcal{A}}$. Further recall that a function $f : X \rightarrow Y$ is the set $\{(x, y) \mid f(x) = y\}$. Therefore, an assignment is a set of pairs of an attribute and an attribute value. We will thus also use the notation $s \subseteq t$ with s and t assignments.

Example 4.1 Let $\mathcal{A} = \{x, y\}$, $\mathsf{dom}(x) = \{x_1, x_2\}$ and $\mathsf{dom}(y) = \{y_1\}$. Then we have the following possible assignments.

- $\emptyset$ is the assignment on the empty set of attributes,
- assignments on $\{x\}$: $s_1 = \{(x, x_1)\}$ and $s_2 = \{(x, x_2)\}$,
- assignments on $\{y\}$: $t = \{(y, y_1)\}$, and
- assignments on $\{x, y\}$: $u_1 = \{(x, x_1), (y, y_1)\}$ and $u_2 = \{(x, x_2), (y, y_1)\}$

Consequently, $\mathcal{S}_\mathcal{A} = \{\emptyset, s_1, s_2, t, u_1, u_2\}$. Furthermore, $\mathsf{dod}(s_1) = \mathsf{dod}(s_2) = \{x\}$, $\mathsf{dod}(t) = \{y\}$, and $\mathsf{dod}(u_1) = \mathsf{dod}(u_2) = \{x, y\}$.

We will now show several properties of assignments to obtain some insight into the relations between assignments, their domain of definitions and the subset notation.

Proposition 4.1 *Let $s, t \in \mathcal{S}_\mathcal{A}$ be two assignments.*

1. *if $s(x) = t(x)$ for all $x \in \mathsf{dod}(s)$, then $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$*
2. *$s \subseteq t$ if and only if $s(x) = t(x)$ for all $x \in \mathsf{dod}(s)$*
3. *if $s \subseteq t$, then $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$*

Proof Point 1 by contradiction: assume that for all $x \in \mathsf{dod}(s)$ it is true that $s(x) = t(x)$, but $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$ is not true. Then there is an attribute $x \in \mathsf{dod}(s)$ such that $x \notin \mathsf{dod}(t)$ and $s(x) \in \mathsf{dom}(x)$. If $x \notin \mathsf{dod}(t)$, then $t(x)$ is undefined and thus $s(x)$ cannot be the same as $t(x)$.
Point 2 from left to right: assume that for all $x \in \mathsf{dod}(s)$ it is true that $s(x) = t(x)$ but not $s \subseteq t$. Then there is an element $(x, v) \in s$ such that $(x, v) \notin t$. In that case, s maps attribute x to attribute value v and $x \in \mathsf{dod}(s)$ by definition.
Point 2 from right to left. If $s \subseteq t$, then every $(x, v) \in s$ is also in t.
Point 3: If $s \subseteq t$, then for all $x \in \mathsf{dod}(s)$ it is true that $s(x) = t(x)$ because of point 2. If for all $x \in \mathsf{dod}(s)$ it is true that $s(x) = t(x)$, then $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$ by point 1. ■

It will be convenient to refer to what an assignment assigns to a given set of attributes. For this, the notion of *restriction* is useful. Formally, the restriction of a function $f : X \to Y$ is the same function but then defined on a subset of X. If $f : X \to Y$ is a function and $A \subset X$, then the restriction of f to A is written as $f|_A$.

Example 4.2 Some restrictions in Example 4.1 are: $s_1|_{\{x\}} = s_1$, $u_1|_{\{x\}} = \{(x, x_1)\} = s_1$ and $t|_{\{x\}} = \emptyset$.

Note that if $\mathsf{dod}(s)$ and X are disjoint, then $s|_X = \emptyset$ because the restriction is an assignment of their intersection.

Definition 4.3 (Compatibility) *Let $s, t \in \mathcal{S}$ be two assignments. We say that s* and *t* are compatible *(denoted* $\mathsf{compatible}(s, t)$*) if and only if $s|_{\mathsf{dod}(t)} = t|_{\mathsf{dod}(s)}$. Otherwise s and t are called* incompatible.

If assignments are incompatible, then they assign different attribute values on the same attribute. If two assignments have a disjoint domain of definition, then they do not assign different attribute values on the same attribute and thus they are compatible.

Proposition 4.2 *Let $s, t \in \mathcal{S}_{\mathcal{A}}$ be two assignments. If $\mathsf{dod}(s) \cap \mathsf{dod}(t) = \emptyset$, then s and t are compatible.*

Proof Because $\mathsf{dod}(s) \cap \mathsf{dod}(t) = \emptyset$, it is true that $s|_{\mathsf{dod}(t)} = \emptyset$. For the same reason it is true that $t|_{\mathsf{dod}(s)} = \emptyset$. Consequently, $s|_{\mathsf{dod}(t)} = t|_{\mathsf{dod}(s)}$ and thus s and t are compatible. ■

If an assignment s is a subset of or equal to another assignment t, i.e., $s \subseteq t$, then t contains all the assignments that s makes to attributes and some more. Because t thus does not assign a different attribute value than s to some attribute, s and t must be compatible.

Proposition 4.3 *Let $s, t \in \mathcal{S}_{\mathcal{A}}$ be two assignments. If $s \subseteq t$, then s and t are compatible.*

Proof If $s \subseteq t$, then $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$ and for every $x \in \mathsf{dod}(s)$ it is true that $s(x) = t(x)$. Because $\mathsf{dod}(s) \subseteq \mathsf{dod}(t)$, $s|_{\mathsf{dod}(t)} = s$. Because for every $x \in \mathcal{A}$ it is true that $s(x) = t(x)$, $t|_{\mathsf{dod}(s)} = s$. Consequently, $s|_{\mathsf{dod}(t)} = t|_{\mathsf{dod}(s)}$ and thus s and t are compatible. ■

Note that compatibility is not a transitive relation. For example, let s and u be assignments on the set of attributes $X \subset \mathcal{A}$ that are incompatible and t an assignment on the set of attributes $Y \subset \mathcal{A}$ such that X and Y are disjoint. Because $\mathsf{dod}(s) \cap \mathsf{dod}(t) = \emptyset$ and Proposition 4.2, s and t are compatible. Similarly, because $\mathsf{dod}(t) \cap \mathsf{dod}(u) = \emptyset$ and Proposition 4.2, t and u are compatible. If compatibility were transitive, then s and u should be compatible, but they are not. However, it is possible that if $\mathsf{compatible}(s, t)$ and $\mathsf{compatible}(t, u)$, then $\mathsf{compatible}(s, u)$. Namely, because each assignment is compatible with itself, i.e., $\mathsf{compatible}(s, s)$ for all assignments, if $s = t = u$, then compatibility is transitive.

Assignments In Goals

Chapter 6 proposes how goals can be justified. A goal will be defined as a set of assignments. It is therefore interesting to define what achieving a set of assignments means.

The property of compatibility is too weak to define achievement. Assume an agent has the goal to achieve assignment s in which the attribute 'color' has the attribute value 'green'. Let assignment t be the outcome of some action and in t nothing is assigned on the attribute 'color'. In that case, s and t are compatible, but t does not achieve the goal of 'color' being 'green'.

Definition 4.4 (Achievement) *Let $s \in \mathcal{S}$ be an assignment and $G \subseteq \mathcal{S}$ a set of assignments.* Assignment s achieves the set of assignments G *(denoted $\mathsf{achieves}(s, G)$) if and only if $\exists_{t \in G}[t \subseteq s]$.*

Note that if $t \subseteq s$, then t and s are compatible as explained in Proposition 4.3. Further note that if assignments s and t are compatible, then it is not necessarily the case that s achieves t or t achieves s.

If for each $s, t \in G$ it is true that s and t are compatible, then it is possible to find an assignment r such that for all $s \in G$: $s \subseteq r$. However, typically the assignments in goals will not be compatible. For example, an agent may pursue the goal to obtain attribute 'color' on either 'green', 'blue', or 'black'. Because it is not possible to obtain green, blue, and black, these assignments are incompatible. We will therefore look at sets of assignments that we call *congruous*.

Definition 4.5 (Congruous Sets of Assignments) *Let $G_1, \ldots, G_n$ be sets of assignments. We say that $G_1, \ldots, G_n$* are congruous *if and only if $\exists_{s \in \mathcal{S}} \forall_{1 \leq i \leq n} [\mathsf{achieves}(s, G_i)]$.*

The notions of compatible assignments and congruous sets of assignments are related as follows.

Proposition 4.4 *Let $G_1, \ldots, G_n$ be sets of assignments in $\mathcal{S}_\mathcal{A}$. If $G_1, \ldots, G_n$ are congruous, then there are $g_1 \in G_1, \ldots, g_n \in G_n$ such that for each $1 \leq i < j \leq n$ it is true that g_i and g_j are compatible.*

Proof Because $G_1, \ldots, G_n$ are congruous, there is an assignment that achieves each G_i. Let s be this assignment. By Definition 4.4, in each G_i there is a $t_i \in G_i$ such that $t_i \subseteq s$. By Proposition 4.3, t_i and s must be compatible. The question now arises whether each t_i and t_j are compatible.

Assume that t_i and t_j are not compatible. Then there is an attribute $x \in \mathcal{A}$ such that $t_i(x) \neq t_j(x)$. However, because $t_i \subseteq s$ and $t_j \subseteq s$, s assigns two different attribute values to attribute x, which is impossible because an assignment is a partial function and can thus only assign maximally one attribute value to each attribute. Thus we have a contradiction. Therefore it must be that t_i and t_j are compatible. ■

4.2 Perspectives

In Chapter 3 the notion of perspective was introduced. The value of assignments can be compared from different perspectives. For example, an assignment s may have more value than assignment t from perspective p, less from perspective q, as much from perspective r and so on. In value trees, several different kinds of perspectives are distinguished: (1) the perspectives of agents; (2) the perspectives that represent general areas of concern or basic values; (3) the perspectives that represent intermediate objectives to address a general area of concern; and, (4) the perspectives that represent specific evaluation criteria. We will now define a *perspective structure*, which is a tuple consisting of a set for each type of perspective.

Definition 4.6 (Perspective Structure) *A* perspective structure *is a tuple $\langle \mathcal{P}_a, \mathcal{P}_v, \mathcal{P}_o, \mathcal{P}_c \rangle$ with $\mathcal{P}_a$, $\mathcal{P}_v$, $\mathcal{P}_o$, and $\mathcal{P}_c$ disjoint sets of perspectives.*

The set $\mathcal{P}_a$ denotes the perspectives representing the perspectives of agents, $\mathcal{P}_v$ denotes the perspectives of basic social values, $\mathcal{P}_o$ denotes the perspectives of intermediary objectives and $\mathcal{P}_c$ denotes the set perspectives of specific evaluation criteria. In a number of cases below it does not matter what kind of perspective it is. If $\mathcal{P}$ is a perspective structure, then we will use $\mathcal{P}$ as the set of all perspectives, i.e., $\mathcal{P} = \mathcal{P}_a \cup \mathcal{P}_v \cup \mathcal{P}_o \cup \mathcal{P}_c$.

The set $\mathcal{S}$ of all assignments of which the value we want to compare is induced by the set of attributes and these attributes' attribute values. To compare the value of two assignments from a given perspective, a ternary relation $\leq \subseteq \mathcal{P} \times \mathcal{S} \times \mathcal{S}$ is used. Instead of $(p, s, t) \in \leq$ we will write $s \leq_p t$.

As explained in Subsection 3.2.3, value from a perspective may only depend on a subset of all attributes. In this case, we say that 'value from perspective p is measured by the set X of attributes'. If a perspective p is measured by a set X of attributes, then what attribute

values are assigned to attributes not in X has no effect on the value from perspective p. Consequently, if two assignments s and t assign the same attribute values to X, then s and t should have an equal amount of value from perspective p. Constraint 1 states that if value from p is measured by the set of attributes X and assignment s and t assign the same attribute values on X, then s and t have an equal amount of value from p. Constraint 1 is formalized by adding the second constraint to the following definition.

Definition 4.7 (Perspective-based Value Comparison Structure) *A* Perspective-based Value Comparison Structure *(PVCS) is a tuple* $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ *with* $\mathcal{A}$ *a set of attributes,* $\mathcal{S}$ *a set of assignments on* $\mathcal{A}$*,* $\mathcal{P}$ *a perspective structure,* msr *a partial function* $\mathsf{msr} : \mathcal{P} \rightarrow 2^{\mathcal{A}}$ *and* $\leq \subseteq \mathcal{P} \times \mathcal{S} \times \mathcal{S}$ *such that for all* $p \in \mathcal{P}$ *and for all* $s, t \in \mathcal{S}$*:*

- $s \leq_p s$*,*
- $s|_{\mathsf{msr}(p)} \leq_p t|_{\mathsf{msr}(p)}$ *if and only if* $s \leq_p t$*.*

We will use $s <_p t$ as an abbreviation for $s \leq_p t$ and $t \not\leq_p s$. Furthermore, $s \equiv_p t$ is used as an abbreviation for $s \leq_p t$ and $t \leq_p s$. If $\mathsf{msr}(p) = X$, then we say that value from perspective is measured by the set of attributes X.

Example 4.3 (PVCS) Let the set $\mathcal{A} = \{x_1, x_2\}$ of attributes be as in Example 4.1 and its corresponding set of assignments be denoted as $\mathcal{S}$. We consider two perspectives: $\mathcal{P} = \{p, q\}$, which are measured as follows: $\mathsf{msr}(p) = \{x_1\}$ and $\mathsf{msr}(q) = \{x_1, x_2\}$. Finally, the relation $\leq$ contains:

- reflexivity: $s \leq_p s$ and $s \leq_q s$ for all $s \in \mathcal{S}$
- $s_1 <_p s_2$. Because $\mathsf{msr}(p) = \{x_1\}$ and $u_1 |_{\{x_1\}} = s_1$ and $u_2 |_{\{x_1\}} = s_2$, $u_1 <_p u_2$. Because $s_1 |_{\{x_1\}} = s_1$, it must also be true that $s_1 <_p u_2$. Similarly, $u_1 <_p s_2$ because $s_2 |_{\{x_1\}} = s_2$. Note that t is only comparable to itself from p and not comparable to any other assignment.
- $u_2 <_q u_1$

Then $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ is a PVCS.

The way that $<$ and $\equiv$ are defined gives rise to the following common properties.

Proposition 4.5 *If* $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ *is a PVCS, then the following properties hold (for all* $s, t \in \mathcal{S}_{\mathcal{A}}$ *and all* $p \in \mathcal{P}$*):*

- *asymmetry of preference: if* $s <_p t$*, then not* $t <_p s$*,*
- *symmetry of indifference: if* $s \equiv_p t$*, then* $t \equiv_p s$*,*
- *reflexivity of indifference:* $s \equiv_p s$*, and*
- *incompatibility of preference and indifference: if* $s <_p t$*, then not* $s \equiv_p t$*.*

Proof The proofs are straightforward when using the unabbreviated versions of $<$ and $\equiv$. ■

There are several other properties of PVCSs that we want to highlight. The first is transitivity. Because value comparisons are made from a perspective, transitivity is also defined per perspective. As explained in Subsection 3.2.1, transitivity is an optional property of perspectives rather than a constraint.

Definition 4.8 (Transitivity) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS. We say that perspective $p \in \mathcal{P}$ is*

- I-transitive in δ *iff* $s \equiv_p t$ *and* $t \equiv_p u$ *implies* $s \equiv_p u$ *for all* $s, t, u \in \mathcal{S}_\mathcal{A}$,
- P-transitive in δ *iff* $s <_p t$ *and* $t <_p u$ *implies* $s <_p u$ *for all* $s, t, u \in \mathcal{S}_\mathcal{A}$,
- PI-transitive in δ *iff* $s <_p t$ *and* $t \equiv_p u$ *implies* $s <_p u$ *for all* $s, t, u \in \mathcal{S}_\mathcal{A}$,
- IP-transitive in δ *iff* $s \equiv_p t$ *and* $t <_p u$ *implies* $s <_p u$ *for all* $s, t, u \in \mathcal{S}_\mathcal{A}$.

The second property concerns whether the value of all assignments can be compared from a perspective. If the value of every two assignments can be compared from a perspective, the perspective is called complete.

Definition 4.9 (Completeness) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS. We say that perspective $p \in \mathcal{P}$* is complete in δ *iff either $s \leq_p t$ or $t \leq_p s$ for all $s, t \in \mathcal{S}_\mathcal{A}$.*

4.2.1 Influence Between Perspectives

As explained in Subsection 3.3, we use two relations between perspectives that express the direction of influence between perspectives: positive and negative influence. Given a set of perspectives, a perspective influence structure describes how all those perspectives influence each other. We restrict influence such that a perspective can never influence itself. Furthermore, if a perspective p influences perspective q and q influences another perspective r, then p also influences r. However, special attention must be paid to the direction of influence and therefore we obtain a special kind of transitivity between positive and negative influence.

Definition 4.10 (Perspective Influence Structure) *A* Perspective Influence Structure *is a triple $\langle \mathcal{P}, I_\uparrow, I_\downarrow \rangle$ with $\mathcal{P}$ a perspective structure, and $I_\uparrow$ and $I_\downarrow$ binary irreflexive relations over $\mathcal{P}$ such that:*

- *if* $(p, q), (q, r) \in I_\uparrow$ *then* $(p, r) \in I_\uparrow$
- *if* $(p, q) \in I_\uparrow$ *and* $(q, r) \in I_\downarrow$ *then* $(p, r) \in I_\downarrow$
- *if* $(p, q) \in I_\downarrow$ *and* $(q, r) \in I_\uparrow$ *then* $(p, r) \in I_\downarrow$
- *if* $(p, q), (q, r) \in I_\downarrow$ *then* $(p, r) \in I_\uparrow$

Instead of $(p, q) \in I_\uparrow$ we will also write $p \uparrow q$ and instead of $(p, q) \in I_\downarrow$ we will also write $p \downarrow q$. Furthermore, we will say that *perspective p influences perspective q* if $p \uparrow q$ or $p \downarrow q$. If perspectives are seen as vertices and the influences between perspectives as vertices, then a Perspective Influence Structure (PIS) can be seen as a graph. Because a PIS distinguishes between positive and negative influence, two different kinds of vertices are used to visualize the different influence.

Example 4.4 (Perspective Influence Structure) Let the set of $\mathcal{P}$ be $\{p_1, p_2, p_3, q_1, q_2, r_1, r_2\}$ and $I_\uparrow$ contain $p_1 \uparrow q_2$, $q_1 \uparrow r_1$ and $p_2 \uparrow q_2$. Also, $I_\downarrow$ contains $p_2 \downarrow q_1$, $p_2 \downarrow q_2$ and $q_2 \downarrow r_1$. Note that no perspective influences perspective r_2. The tuple $\langle \mathcal{P}, I_\uparrow, I_\downarrow \rangle$ is not a valid Perspective Influence Structure, because it does not follow transitivity of influence.

We will now extend $I_\uparrow$ and $I_\downarrow$ to follow transitivity of influence. Let $I'_\uparrow$ contain $I_\uparrow$ and $p_1 \uparrow r_1$ (because $p_1 \uparrow q_1$ and $q_1 \uparrow r_1$) and $p_2 \uparrow r_1$ (because $p_2 \downarrow q_1$ and $q_1 \uparrow r_1$). Furthermore,

let $I'_{\downarrow}$ contain $I_{\downarrow}$, $p_2 \downarrow r_1$ (because $p_2 \downarrow q_2$ and $q_2 \downarrow r_1$) and $p_3 \downarrow r_1$ (because $p_3 \uparrow q_2$ and $q_2 \downarrow r_1$). The tuple $\left\langle \mathcal{P}, I'_{\uparrow}, I'_{\downarrow} \right\rangle$ now is a Perspective Influence Structure and is visualized in the 'influence diagram' in Figure 4.1.

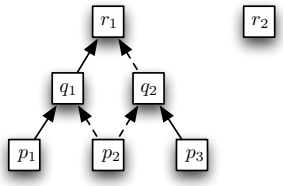


Figure 4.1: *Influence Diagram. Each node represents a perspective. The normal arrows denote positive influence and the dotted arrows denote negative influence. Indirect influence that follows from transitivity is not visualized explicitly. For example, p_1 influences r_1 because p_1 influences q_1 and because q_1 influences r_1. However, the the influence of p_1 on r_1 is indirect and therefore not visualized.*

Proposition 4.6 *Let $\langle \mathcal{P}, I_{\uparrow}, I_{\downarrow} \rangle$ be a PIS and $I = I_{\uparrow} \cup I_{\downarrow}$. Then I is transitive.*

Proof Let $p, q, r \in \mathcal{P}$ be such that $(p, q), (q, r) \in I$. There are four cases we can distinguish in which different constraints of Definition 4.10 apply: (1) if $(p, q), (q, r) \in I_{\uparrow}$, then the first constraint ensures $(p, r) \in I_{\uparrow}$; (2) if $(p, q) \in I_{\uparrow}$ and $(q, r) \in I_{\downarrow}$, then the second constraint ensures that $(p, r) \in I_{\downarrow}$; (3) if $(p, q) \in I_{\downarrow}$ and $(q, r) \in I_{\uparrow}$, then the third constraint ensures that $(p, r) \in I_{\downarrow}$; and, (4) if $(p, q), (q, r) \in I_{\downarrow}$, then the fourth constraint ensures that $(p, r) \in I_{\uparrow}$. In all four cases it is true that $(p, r) \in I$. Therefore, I is transitive. ■

Proposition 4.7 *Let $\langle \mathcal{P}, I_{\uparrow}, I_{\downarrow} \rangle$ be a PIS and $I = I_{\uparrow} \cup I_{\downarrow}$. There are no cycles in I.*

Proof Because $I_{\uparrow}$ and $I_{\downarrow}$ are irreflexive and because there is transitivity, if there is a cycle, then there is a perspective p such that p influences itself through a path of influences. Because of transitivity, p influences itself which contradicts with the irreflexivity property. ■

The msr function in a PVCS specifies what attributes are used to measure value from a perspective. If $\mathsf{msr}(p) = X$, then the value that assignments get from perspective p only depends on what attribute values are assigned to attributes in X. If a perspective q is influenced by another perspective p, then value from q also depends on value from p. Consequently, if value from p is measured by X, then value from q also depends on X. If p influences q, then the attributes that measure p should be contained in the attributes that measure q, i.e., $\mathsf{msr}(p) \subseteq \mathsf{msr}(q)$ if both are defined. If this is the case for all influences, we say that a PVCS is 'measures-consistent'. Otherwise the PVCS is 'measures-inconsistent'.

Definition 4.11 (Measures-Consistent) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS and $\iota = \langle \mathcal{P}, I_{\uparrow}, I_{\downarrow} \rangle$ a PIS. We say that δ* is measures-consistent under *ι if and only if for all perspectives p and q the following holds: if p influences q, value from p is measured by set attributes X and value from q is measured by set attributes Y, then $X \subseteq Y$.*

4.2.2 Relative Importance

As explained in Subsection 3.3.3, the perspectives that influence a perspective p may differ in their importance for p. For example, although multiple objectives influence the agent's perspective, one objective may be more important to the agent than another. Note that it

does not matter whether a perspective p positively or negatively influences perspective q for p's importance relative to the other perspectives that influence q. However, importance for a perspective is reflexive and transitive.

Definition 4.12 (Perspective Importance Ordering) *Let $\mathcal{P}$ be a perspective structure. An* perspective importance ordering of $\mathcal{P}$ *is a ternary relation over perspectives such that*

- *for all $p, q \in \mathcal{P}$ it is true that $p \trianglelefteq_q p$, and*
- *for all $p_1, p_2, p_3, q \in \mathcal{P}$ it is true that if $p_1 \trianglelefteq_q p_2$ and $p_2 \trianglelefteq_q p_3$, then $p_1 \trianglelefteq_q p_3$.*

Note that these two conditions state that importance for a perspective is reflexive and transitive respectively. We will use $p \triangleleft_r q$ to abbreviate $p \trianglelefteq_r q$ and not $q \trianglelefteq_r p$. Also, we use $p \bowtie_r q$ as an abbreviation of $p \trianglelefteq_r q$ and $q \trianglelefteq_r p$.

Example 4.5 (Importance Of Perspectives) Let ι be a PIS that is visualized in Figure 4.2 and let perspective q_3 be more important for r than q_2 and q_2 be equally important for r as q_1. Because importance is transitive, this means that q_3 is also more important for r than q_1. In other words, $q_1 \bowtie_r q_2$, $q_2 \triangleleft_r q_3$ and $q_1 \triangleleft_r q_3$. Perspective p_2 is more important for q_2 than p_1, i.e., $p_1 \triangleleft_{q_2} p_2$.

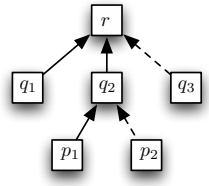


Figure 4.2: *Influence Diagram. Each node represents a perspective. The normal arrows denote positive influence and the dotted arrows denote negative influence. Note that p_1 and p_2 influence r indirectly through q_2.*

4.3 Monadic Evaluations

As explained in Subsection 3.4, terms like 'good', 'bad', 'satisfactory', 'best', 'worst', 'beautiful', and 'cheap' are commonly used in informal discourse about what a person values. A term like 'good' or 'bad' is a predicate that can be given to an alternative, outcome, or an attribute of an outcome. Hansson (2001) calls such predicates *monadic value predicates*. Monadic value predicates can be used to evaluate alternatives, outcomes, and attribute values of outcomes. Evaluations by means of monadic value predicates are subjective. Namely, what one agent finds good, another finds bad. Moreover, evaluations are done from a perspective. For example, a car may be good from the perspective of speed, but bad from the perspective of costs.

If value from p is measured by attributes X, then only assignments that have assigned X can be evaluated from p. Furthermore, if value from perspective p only depends on attributes in the set X, the monadic evaluations from p also only depend on attributes in X. Consequently, if two assignments assign the same attribute values on attributes in X, then they should be evaluated the same.

Definition 4.13 (Monadic Evaluations) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS with $\mathcal{P} = \langle \mathcal{P}_a, \mathcal{P}_v, \mathcal{P}_o, \mathcal{P}_c \rangle$ a perspective structure. A* Monadic Evaluation Structure based on δ *is a tuple $\langle \mathcal{M}, \mathsf{eval} \rangle$ where $\mathcal{M}$ is a set of monadic value predicates and* $\mathsf{eval} \subseteq \mathcal{P}_a \times \mathcal{S} \times \mathcal{M} \times \mathcal{P}$ *such that:*

- *if $s \equiv_p t$, then $\mathsf{eval}(\alpha, s, m, p)$ if and only if $\mathsf{eval}(\alpha, t, m, p)$*
- *if $\mathsf{msr}(p) = X$ and $\mathsf{eval}(\alpha, s|_X, m, p)$, then $\mathsf{eval}(\alpha, s, m, p)$.*

Example 4.6 (Monadic Evaluations) Let $\mathcal{A} = \{x, y\}$ such that both attributes have the domain of attribute values $\{0, 1\}$. Let s_0 be the assignment $\{(x, 0)\}$, s_1 the assignment $\{(x, 1)\}$, t_0 be the assignment $\{(y, 0)\}$, and t_1 the assignment $\{(y, 1)\}$. Value from perspective p is measured by attribute x. For convenience, we will use X to denote the set $\{x\}$. In that case, $\mathsf{msr}(p) = X$ and $t_0 \mid_X = t_1 \mid_X = \emptyset$. Finally, let $\langle \{m, n\}, \mathsf{eval} \rangle$ be a monadic evaluation structure.

Assume that assignments s_0 and s_1 have an equal amount of value from perspective p, i.e., $s_0 \equiv_p s_1$, and that agent α evaluates s_0 as m from perspective p, i.e., $\mathsf{eval}(\alpha, s_0, m, p)$. Because of $s_0 \equiv_p s_1$ being true and the first constraint in Definition 4.13, agent α should give assignment s_1 the same monadic value predicate from p as s_0, i.e., $\mathsf{eval}(\alpha, s_1, m, p)$.

Now assume that agent α evaluates the empty assignment as n from perspective p, i.e., $\mathsf{eval}(\alpha, \emptyset, n, p)$. Because $t_0 \mid_X = \emptyset$ and the second constraint in Definition 4.13, agent α should give assignment t_0 the same monadic value predicate from perspective p as the empty assignment, i.e., $\mathsf{eval}(\alpha, t_0, n, p)$. For the same reason $\mathsf{eval}(\alpha, t_1, n, p)$ should be true.

There are many monadic value predicates that are used in everyday life, e.g., good, bad, very good, the worst, mediocre, beautiful, fast, fair and so on. We will use the following properties that Hansson (2001) uses to describe monadic value predicates: positive, negative, and exclusive. Subsection 3.4 described these properties in more detail. If a monadic value predicate m is positive, then if assignment s is evaluated as m and assignment t is better than m, then t must also be evaluated as m. An example positive value predicate is 'good' because if something is good, then something that is better must also be good. We will now define a monadic value predicate description, which consists of a set of monadic value predicates and a description of them in terms of these properties.

Definition 4.14 (Monadic Value Predicate Description) *A* Monadic Value Predicate Description *(MVPD) is a tuple $\langle \mathcal{M}, \mathsf{exclusive}, \mathsf{pos}, \mathsf{neg} \rangle$ where $\mathcal{M}$ is a set of monadic value predicates,* exclusive *is an irreflexive symmetrical binary relation over $\mathcal{M}$ and* pos *and* neg *are disjoint subsets of $\mathcal{M}$.*

A monadic value predicate description specifies the properties of a set of monadic value predicates. If a set of monadic evaluations satisfies the properties of a MVPD, then we say that these evaluations are consistent with that MVPD. For example, if a MVPD states that value predicates m and n are exclusive and an assignment s is evaluated as both m and n from some perspective, then that evaluation is not consistent with that MVPD.

Subsection 3.4 describes the various properties of monadic value predicates in more detail. We will now define when a monadic evaluation structure is called consistent under a MVPD. This definition is based on the properties of monadic value predicates as described in

Subsection 3.4. An example property is the following: if a monadic value predicate m is positive, agent α evaluates assignment s as m from perspective p, and perspective p positively influences agent α's perspective, then agent α should also evaluate assignments preferred to s from p as m. Consider for example the monadic value predicate 'good', which is positive. If agent α evaluates a profit of a thousand euro as 'good' from the perspective of profit and the perspective profit positively influences the agent's perspective, then α should also evaluate higher profits as good from the perspective of profit.

Definition 4.15 (Consistent Monadic Evaluations) *A monadic evaluation structure* $\langle \mathcal{M}, \mathsf{eval} \rangle$ *is called* consistent under MVPD $\langle \mathcal{M}, \mathsf{exclusive}, \mathsf{pos}, \mathsf{neg} \rangle$ and PIS $\langle \mathcal{P}, I_\uparrow, I_\downarrow \rangle$ *if and only if*

- *exclusivity: if* $(m, n) \in \mathsf{exclusive}$*, then it is never the case that* $\mathsf{eval}(\alpha, s, m, p)$ *and* $\mathsf{eval}(\alpha, s, n, p)$ *are both true*
- *positivity: if* $m \in \mathsf{pos}$*, then:*
  - *if* $\mathsf{eval}(\alpha, s, m, p)$ *and* $(p, \alpha) \in I_\uparrow$ *then* $\mathsf{eval}(\alpha, t, m, p)$ *for every* t *such that* $s \leq_p t$
  - *if* $\mathsf{eval}(\alpha, s, m, p)$*,* $(p, \alpha) \in I_\downarrow$ *and* $t \leq_p s$*, then* $\mathsf{eval}(\alpha, t, m, p)$
- *negativity: if* $m \in \mathsf{neg}$*, then:*
  - *if* $\mathsf{eval}(\alpha, s, m, p)$*,* $(p, \alpha) \in I_\uparrow$ *and* $t \leq_p s$*, then* $\mathsf{eval}(\alpha, t, m, p)$
  - *if* $\mathsf{eval}(\alpha, s, m, p)$*,* $(p, \alpha) \in I_\downarrow$ *and* $s \leq_p t$*, then* $\mathsf{eval}(\alpha, t, m, p)$
- *If* $(p, \alpha) \notin I_\uparrow$ *and* $(p, \alpha) \notin I_\downarrow$*, then not* $\mathsf{eval}(\alpha, s, m, p)$ *for any* s *and* m*.*

In this thesis we will focus on the monadic value predicates 'good' and 'bad' (denoted good and bad), i.e., $\mathcal{M} = \{\mathsf{good}, \mathsf{bad}\}$. This means that the monadic value predicate description that we will use is $\langle \mathcal{M}, \mathsf{exclusive}, \mathsf{pos}, \mathsf{neg} \rangle$ where $(\mathsf{good}, \mathsf{bad}), (\mathsf{bad}, \mathsf{good}) \in \mathsf{exclusive}$, $\mathsf{good} \in \mathsf{pos}$ and $\mathsf{bad} \in \mathsf{neg}$.

Consider three assignments s, t, and u such that $s <_p t <_p u$. If both s and u are evaluated as m from perspective p, then it is interesting to investigate whether t is also evaluated as m from p. This property is called *continuity* and is proven for positive and negative value predicates as follows.

Proposition 4.8 (Positive Continuity) *Let* $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ *be a PVCS. For every monadic evaluation structure based on* δ *that is consistent under MVPD* μ *and PIS* $\langle \mathcal{P}, I_\uparrow, I_\downarrow \rangle$ *the following is true for all assignments* $s, t, u \in \mathcal{S}_\mathcal{A}$*: if* $m \in \mathsf{pos}$*,* $\mathsf{eval}(\alpha, s, m, p)$*,* $\mathsf{eval}(\alpha, u, m, p)$*,* $s \leq_p t$ *and* $t \leq_p u$*, then* $\mathsf{eval}(\alpha, t, m, p)$*.*

Proof Since $\mathsf{eval}(\alpha, s, m, p)$ is true, it must be true that p influences α by Definition 4.15. If p influences α, then either $p \uparrow \alpha$ or $p \downarrow \alpha$ is true.

First we consider $p \uparrow \alpha$. Because $\mathsf{eval}(\alpha, s, m, p)$ and $s \leq_p t$ are true, it must be true that $\mathsf{eval}(\alpha, t, m, p)$ by Definition 4.15. Next we consider $p \downarrow \alpha$. Because $\mathsf{eval}(\alpha, u, m, p)$ and $t \leq_p u$, it must be true that $\mathsf{eval}(\alpha, t, m, p)$ by Definition 4.15. ∎

Negative continuity can be proved in a similar fashion. Proposition 4.8 and a similar proposition for negative continuity prove that Constraint 10 holds. We will now show how the formalism that we have introduced in this chapter can be used in the running example.

4.4 Running Example

Recall the example described in Chapter 1, where a fire commander student is situated in a situation where a factory is on fire and there a number of victims inside the building. The student has to decide what course of action to take.

Attributes And Assignments

Various attributes can be used to express the outcome of decisions, but we will focus on the attributes $\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. Table 4.1 describes what each attribute in $\mathcal{A}$ denotes and what the domain of attribute values for each attribute is. For example, attribute x_1 denotes the number of minutes that the victims are inside the factory. Attribute x_1's domain of attribute values are the natural numbers.

Table 4.1: *Attributes of outcomes*

Attribute	Denotes	Attribute Values
x_1	Number of minutes that victims are inside	$\mathbb{N}$
x_2	Number of minutes that personnel is inside	$\mathbb{N}$
x_3	Amount of smoke	$\{\mathsf{none}, \mathsf{some}, \mathsf{lots}\}$
x_4	Amount of chemicals leaked into earth	$\{\mathsf{none}, \mathsf{some}, \mathsf{lots}\}$
x_5	Number of minutes that victims are near fire	$\mathbb{N}$
x_6	Number of minutes that personnel is near fire	$\mathbb{N}$

Assignments (as in Definition 4.2) are partial functions that map a subset of attributes on their corresponding attribute values. For example, the assignment $s = \{(x_1, 10), (x_4, \mathsf{lots})\}$ denotes that the victims are inside for ten minutes and that lots of chemicals leak into the earth. Note that it only assigns attribute values to attributes x_1 and x_4 and does not state what attribute values are assigned on the other attributes. Another assignment $t = \{(x_1, 10)\}$ denotes that the victims are inside for ten minutes. Note that s is a more specific assignment than t. The restriction of s to the set of attributes $\{x_1\}$ is $s_{\{x_1\}} = \{(x_1, 10)\}$ and thus the same as t. Because s and t assign the same attribute value on x_1, they are compatible (see Definition 4.3).

Different perspectives on assignments

We will assume that the student's value tree is as visualized in Figure 4.3. Recall from Section 2.5 that we will use the ten basic abstract values from Schwartz (1992). For the example, we will focus on the three basic abstract values of universalism, security, and benevolence. These three basic values are the most relevant values in this example and they will cause several interesting tradeoffs that we want to deal with. There is one objective of minimizing impact on the environment that promotes the value of universalism. Furthermore, the objectives of maximizing safety of the victims and maximizing the safety of the personnel both promote the value of security. Finally, the objective of maximizing the safety of the personnel promotes the value of benevolence. For each attribute in $\mathcal{A}$, a criterion perspective is introduced.

Figure 4.3: *Value Tree*

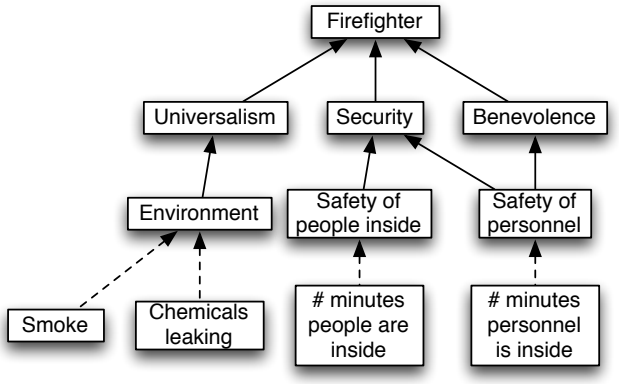


Table 4.2 describes the perspectives in the value tree visualized in Figure 4.3. First, the name of the perspective is given, then what the perspective denotes.

Table 4.2: *Perspectives*

Perspective	**Denotes**
α	The student's perspective
v_1	Universalism
v_2	Security
v_3	Benevolence
o_1	Impact on the environment
o_2	Safety of victims
o_3	Safety of personnel
c_1	Minutes victims are inside
c_2	Minutes personnel is inside
c_3	Amount of smoke
c_4	Amount of chemicals leaked into earth
c_5	Minutes victims are near fire
c_6	Minutes personnel is near fire

A perspective structure (as in Definition 4.6) organizes a set of perspectives into the following disjoint sets: (1) the perspectives of agents; (2) the perspectives of basic values; (3) the perspectives of objectives; and, (4) the perspectives of specific evaluation criteria. Table 4.2 describes the following perspective structure:

$$\mathcal{P} = \langle \{\alpha\}, \{v_1, v_2, v_3\}, \{o_1, o_2, o_3\}, \{c_1, c_2, c_3, c_4, c_5, c_6\} \rangle$$

A perspective-based value comparison structure (PVCS) is a tuple $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ consisting of a set of attributes, a set of assignments on those attributes, a perspective structure, a ternary relation $\leq$ and a partial function msr that maps a perspective on a set of attributes.

In the PVCS that corresponds to the example, we will use the attributes, assignments, and perspective structure as already described, but we still need to instantiate the $\leq$ relation and msr function. Each criterion perspective c_i is measured by the set of attributes $\{x_1\}$ (for $1 \leq i \leq 6$). The $\leq$ relation is instantiated for the criteria perspective of which the attribute has domain $\mathbb{N}$ as follows: $(c_i, s, t) \in \leq$ if and only if assignment t has an x_i-value as high as or higher than the x_i-value of assignment s. The remaining criteria perspectives that are not measured by $\mathbb{N}$ are instantiated as expected.

The PVCS that corresponds to the example is the tuple $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ where

- for $1 \leq i \leq 6$: $\mathsf{msr}(c_i) = \{x_i\}$
- for $i \in \{1, 2, 5, 6\}$ and for all $y_1, y_2 \in \mathsf{dom}(x_i)$: $y_1 \leq y_2$ iff $\{(x_i, y_1)\} \leq_{c_i} \{(x_i, y_2)\}$
- for $i \in \{3, 4\}$: $\{(x_i, \mathsf{none})\} <_{c_i} \{(x_i, \mathsf{some})\}$ and $\{(x_i, \mathsf{none})\} <_{c_i} \{(x_i, \mathsf{lots})\}$ and $\{(x_i, \mathsf{some})\} <_{c_i} \{(x_i, \mathsf{lots})\}$

For example, because 10 is a higher x_1-value than 0 it is true that $\{(x_1, 0)\} <_{c_1} \{(x_1, 10)\}$. Also, because lots is a higher x_3-value than none it is true in the PVCS that $\{(x_3, \mathsf{none})\} <_{c_3} \{(x_3, \mathsf{lots})\}$.

Further recall from Definition 4.7 that a PVCS satisfies the two conditions: (1) $s \leq_p s$; and, (2) $s \mid_{\mathsf{msr}(p)} \leq_p t \mid_{\mathsf{msr}(p)}$ if and only if $s \leq_p t$. To satisfy these two conditions the $\leq$ is further instantiated. For example, consider the assignments $s = \{(x_1, 0), (x_2, 10)\}$ and $t = \{(x_1, 1), (x_2, 10)\}$. Because $\mathsf{msr}(c_1) = \{x_1\}$ it is true that $s \mid_{\mathsf{msr}(c_1)} = \{(x_1, 0)\}$ and $t \mid_{\mathsf{msr}(c_1)} = \{(x_1, 1)\}$. Because 1 is a higher x_1-value than 0, it is true that $s \mid_{\mathsf{msr}(c_1)} <_{c_1} t \mid_{\mathsf{msr}(c_1)}$. Because of the second condition this means that $s <_{c_1} t$ is also true in the PVCS. On the other hand, because $\mathsf{msr}(c_2) = \{x_2\}$, it is true that $s \mid_{\mathsf{msr}(c_2)} = \{(x_2, 10)\}$ and $t \mid_{\mathsf{msr}(c_2)} = \{(x_2, 10)\}$. Consequently, $s \mid_{\mathsf{msr}(c_2)} \equiv_{c_1} t \mid_{\mathsf{msr}(c_2)}$ and thus $s \equiv_{c_1} t$ because of the second condition.

Influence Between Perspectives

Recall from Definition 4.10 that a perspective influence structure is a tuple $\langle \mathcal{P}, I_\uparrow, I_\downarrow \rangle$ with $\mathcal{P}$ a perspective structure and $I_\uparrow$ and $I_\downarrow$ binary irreflexive relations over $\mathcal{P}$ such that four transitivity-like conditions hold.

All the three values positively influence the student's perspective. Furthermore, the objectives in $\mathcal{P}$ influence the values as follows: (1) objective o_1 of maximizing environment positively influences the value v_1 of universalism; (2) objective o_2 of maximizing safety of victims positively influences the value v_2 of security; and, (3) objective o_3 of maximizing safety of personnel positively influences both value v_2 of security and value v_3 of benevolence. All criteria perspectives negatively influence the objectives of which they measure the performance. This means that lower attribute values are preferred to higher attribute values from the perspectives of objectives.

Because of the transitivity conditions and because the criteria perspectives negatively influence the objectives perspectives, $I_\uparrow$ and $I_\downarrow$ need to contain that every criterion c_i negatively influences the values that are influenced by the objectives that c_i influences. For example, because criterion c_3 negatively influences objective o_1 and o_1 positively influences value v_1, c_3 negatively influences v_1. For the same reasons, every criterion negatively influences the student's perspective α. Also because all objectives only positively influence the values and

the values positively influence the agent's perspective, the transitivity conditions state that every objective must positively influence the agent's perspective.

$$\begin{aligned} I_{\uparrow} =&\{(v_1,\alpha),(v_2,\alpha),(v_3,\alpha),(o_1,v_1),(o_2,v_2),(o_3,v_2),\\ &(o_3,v_3),(o_1,\alpha),(o_2,\alpha),(o_3,\alpha)\}\\ I_{\downarrow} =&\{(c_3,o_1),(c_4,o_1),(c_1,o_2),(c_5,o_2),(c_2,o_3),(c_6,o_3),\\ &(c_3,v_1),(c_4,v_1),(c_1,v_2),(c_5,v_2),(c_2,v_3),(c_6,v_3),\\ &(c_3,\alpha),(c_4,\alpha),(c_1,\alpha),(c_5,\alpha),(c_2,\alpha),(c_6,\alpha)\} \end{aligned}$$

The Perspective Influence Structure (PIS) for the example is thus $\langle \mathcal{P}, I_{\uparrow}, I_{\downarrow} \rangle$. Note that this PIS follows the four transitivity conditions of its definition.

Relative Importance Of Perspectives

The student does not care equally about every perspective that influences his perspective. Namely, the student finds his value perspectives important to different degrees: benevolence is more important than universalism and security is more important than both benevolence and universalism. Consequently, $v_1 \lhd_{\alpha} v_2$, $v_1 \lhd_{\alpha} v_3$ and $v_3 \lhd_{\alpha} v_2$ are true. Furthermore, criterion perspective c_5 is better suited than criterion perspective c_1 to measure objective perspective o_2. Also, criterion perspective c_6 is better suited than criterion perspective c_2 to measure objective perspective o_3. Consequently, $c_1 \lhd_{o_2} c_5$ and $c_2 \lhd_{o_3} c_6$ is true.

Monadic Evaluations

Using monadic evaluations, agents can express that assignments have a certain level of value from a perspective. Section 3.4 explains in detail what monadic statements are and Section 4.3 formalizes monadic evaluations. A monadic evaluation is done by an agent and expresses that an assignment obtains a certain monadic value predicate from a certain perspective. For example, when watching movies, agent α evaluates the assignment 'actor is Jim Carey' as 'good' from the perspective of fun.

As explained before, we focus on two monadic value predicates good and bad, i.e., $\mathcal{M} = \{\mathsf{good}, \mathsf{bad}\}$. The corresponding Monadic Value Predicate Description (MPVD) is $\langle \mathcal{M}, \mathsf{exclusive}, \mathsf{pos}, \mathsf{neg} \rangle$ with $\mathsf{exclusive} = \{(\mathsf{good}, \mathsf{bad}), (\mathsf{bad}, \mathsf{good})\}$, $\mathsf{pos} = \{\mathsf{good}\}$ and $\mathsf{neg} = \{\mathsf{bad}\}$.

Concerning the number of minutes that the victims are inside (attribute x_1), the student finds 10 minutes 'good' and 20 minutes 'bad' from the criterion perspective c_1. From the criterion perspective of how many minutes the personnel is inside, the student is critical and evaluates 0 'good' and 10 minutes 'bad' (w.r.t. attribute x_2). The student also cares about criterion perspective c_3 denoting 'amount of smoke' and evaluates 'no smoke' as good and 'lots of smoke' as bad. Concerning the amount of chemicals leaked into the earth (criterion perspective c_4), the student evaluates 'none' as good and 'lots' as bad. The student also cares about the number of minutes that the victims and the personnel are near fire (criterion perspectives c_5 and c_6 respectively). For c_5, he evaluates 10 minutes as good and 20 minutes as bad. For c_6, he evaluates 0 minutes as good and 10 minutes as bad.

These monadic evaluations are represented with the 4-ary relation eval, which contains the following.

- from perspective c_1, α evaluates the x_1 value of 10 minutes as good and of 20 minutes as bad: $(\alpha, \{(x_1, 10)\}, \mathsf{good}, c_1), (\alpha, \{(x_1, 20)\}, \mathsf{bad}, c_1) \in \mathsf{eval}$
- from perspective c_2, α evaluates the x_2 value of 0 minutes as good and of 10 minutes as bad: $(\alpha, \{(x_2, 0)\}, \mathsf{good}, c_2), (\alpha, \{(x_2, 10)\}, \mathsf{bad}, c_2) \in \mathsf{eval}$
- from perspective c_3, α evaluates the x_3 value of none as good and of lots as bad: $(\alpha, \{(x_3, \mathsf{none})\}, \mathsf{good}, c_3), (\alpha, \{(x_3, \mathsf{lots})\}, \mathsf{bad}, c_3) \in \mathsf{eval}$
- from perspective c_4, α evaluates the x_4 value of none as good and of lots as bad: $(\alpha, \{(x_4, \mathsf{none})\}, \mathsf{good}, c_4), (\alpha, \{(x_4, \mathsf{lots})\}, \mathsf{bad}, c_4) \in \mathsf{eval}$
- from perspective c_5, α evaluates the x_5 value of 10 minutes as good and of 20 minutes as bad: $(\alpha, \{(x_5, 10)\}, \mathsf{good}, c_5), (\alpha, \{(x_5, 20)\}, \mathsf{bad}, c_5) \in \mathsf{eval}$
- from perspective c_6, α evaluates the x_6 value of 0 minutes as good and of 10 minutes as bad: $(\alpha, \{(x_6, 0)\}, \mathsf{good}, c_6), (\alpha, \{(x_6, 10)\}, \mathsf{bad}, c_6) \in \mathsf{eval}$

From Definition 4.13 we may recall that if $\mathsf{msr}(p) = X$ and $\mathsf{eval}(\alpha, s\mid_X, m, p)$, then $\mathsf{eval}(\alpha, s, m, p)$. Further recall from the PVCS that each criterion perspective c_i is measured only by the attribute x_1, i.e., for $1 \leq i \leq 6$ it is true that $\mathsf{msr}(c_i) = \{x_i\}$. We will specify α's monadic evaluations from different perspectives using only assignments on the attributes that measure that perspective. For example, from perspective c_1 α evaluates the assignment $\{(x_1, 10)\}$ as good. This means that for c_1 every assignment is evaluated as good by α if its restriction to $\{x_1\}$ is $\{(x_1, 10)\}$.

Table 4.3: *Student α's monadic evaluations w.r.t. the criterion perspectives*

Perspective	Good	Bad
c_1	$t_1 = \{(x_1, 10)\}$	$t'_1 = \{(x_1, 20)\}$
c_2	$t_2 = \{(x_2, 0)\}$	$t'_2 = \{(x_2, 10)\}$
c_3	$t_3 = \{(x_3, \mathsf{none})\}$	$t'_3 = \{(x_3, \mathsf{lots})\}$
c_4	$t_4 = \{(x_4, \mathsf{none})\}$	$t'_4 = \{(x_4, \mathsf{lots})\}$
c_5	$t_5 = \{(x_5, 10)\}$	$t'_5 = \{(x_5, 20)\}$
c_6	$t_6 = \{(x_6, 0)\}$	$t'_6 = \{(x_6, 10)\}$

Recall from the Perspective Influence Structure (PIS) that every criterion negatively influences student α's perspective. In order to make MES $\langle \mathcal{M}, \mathsf{eval}\rangle$ consistent under the MPVD and the PIS that we have specified before, for all $1 \leq i \leq 6$ and all assignments $s, t \in \mathcal{S}$ the following two conditions must be true: (1) if $t \leq_{c_i} s$ and $\mathsf{eval}(\alpha, s, \mathsf{good}, c_i)$, then $\mathsf{eval}(\alpha, t, \mathsf{good}, c_i)$; and, (2) if $s \leq_{c_i} t$ and $\mathsf{eval}(\alpha, s, \mathsf{bad}, c_i)$, then $\mathsf{eval}(\alpha, t, \mathsf{bad}, c_i)$. Consider criterion perspective c_1 measured by attribute x_1 denoting the number of minutes that the victims are inside. These two conditions ensure that α evaluates every x_1-value between 0 and 10 is evaluated as good from c_1, and every x_1-value of 20 or higher is evaluated as bad from c_1. The x_1-values between 10 and 20 are not evaluated as good nor bad by α. Furthermore, only the x_2-value of 0 is evaluated as good from perspective c_2 by α.

4.5 Chapter Summary

The previous chapter proposed a conceptual framework for value, which describes and relates a number of concepts that are necessary to describe value in decision making. Research

question 1b asks how argumentation can be used to reason about value. In order to be able to use argumentation it was therefore necessary to formalize the conceptual framework of the previous chapter, which we have done in this chapter. The next chapter uses the formalization of this chapter to define an argumentation framework to reason about value.

We first formalized the notions of attribute, attribute value and assignment in Section 4.1, which are necessary to describe decisions. Because value is always seen from a perspective in the conceptual framework, we formalized the notion of a perspective in Section 4.2. Also influence between perspectives and relative importance between perspectives was formalized. Section 4.3 then formalized monadic evaluations. First a structure was defined that describes the properties of monadic value predicates. Then we defined when a set of monadic evaluations satisfy the properties of the monadic value predicates. Finally, we have demonstrated in Section 4.4 how this formalization can be used for the example in the introduction.

5 Argumentation about Perspective-Based Value

The previous chapter formalized the conceptual framework for value that we proposed in Chapter 3. To this end, Perspective-based Value Comparison Structures (PVCS) were defined. However, for our purposes, a PVCS is not a fully general representation of preferences. In a PVCS, each assignment is either strictly or equally preferred or incomparable to another assignment from each perspective. Lack of knowledge about and lack of determinateness of value from a perspective cannot be expressed in a PVCS while both are important to express for our purposes. Moreover, it is not straightforward to use the formalism of the previous chapter in an argumentation context. In this chapter we address research question 1b: 'how can argumentation be used to reason about, justify, and refute what a decision maker values' by proposing an argumentation logic based on the formalization of the previous chapter.

A common approach to increase expressive power is to use a set of sentences that describe the value of assignments from the different perspectives. A set of sentences is validated to see what preference relations correspond. Hansson (2001) introduces an object language consisting of sentences expressing preference. This object language is used to represent preference states, which are idealized states of mind with respect to value comparisons. Sets of sentences are then related to preference relations as follows: (1) start with a set S of sentences in a suitable object language (S is called the knowledge base); (2) use a standard first-order consequence relation to derive sentences S'; and finally, (3) determine all preference relations that satisfy S'. A set of sentences thus corresponds to a set of preference relations. If the set of corresponding preference relations is empty, then the set of sentences is said to be inconsistent. If there are multiple preference relations that correspond, then we speak of either lack of knowledge about preference or lack of determinateness of preferences.

In a decision situation there may be inconsistent knowledge, e.g., about what the current circumstances are. However, we want to be able to work with possibly inconsistent knowledge bases and we want to make use of defeasible inference rules. The standard first-order consequence relation is not suitable for both purposes. Namely, an inconsistent set of sentences always corresponds to no preference relations and defeasible inference rules cannot be formalized. The argumentation framework described in Section 2.1 is suitable for these purposes. Given the argumentation framework of an argumentation theory, every extension corresponds to a preference relation.

Section 5.1 proposes an Argumentation System for Perspective-based Value (ASPV) and a set of axioms to argue about how the value of assignments compare from different perspec-

tives. Section 5.2 then proposes a meta-argumentation system to reason about the relative strength of arguments in an ASPV. Section 5.3 uses the running example to demonstrate the formalism proposed in this chapter. Finally, Section 5.4 summarizes this chapter.

5.1 Object-Level Argumentation

In this section we propose how argumentation can be used to reason about value from different perspectives. For this purpose, we first propose an argumentation system in Subsection 5.1.1. As specified in Definition 2.1, an argumentation system consists of a language, a set of strict and defeasible inference rules, and a contrariness function. Next in Subsection 5.1.2, we will propose a class of argumentation theories that is suitable for reasoning about value from different perspectives.

5.1.1 Argumentation System for Perspective-Based Value

To represent value statements, a Language for Perspective-based Value (LPV) is proposed. An LPV is defined on the basis of a Perspective-based Value Comparison Structure (PVCS) as defined in Definition 4.7. A PVCS is a tuple consisting of a set of attributes, a set of assignments, a perspective structure, and a set of monadic value predicates. Each attribute, assignment, perspective and monadic value predicate is introduced as a constant in the language. To express relations between these elements, several predicates are used. Recall from Section 4.1 that we use a restriction function $|$ on assignments to refer to what an assignment assigns to a given set of attributes. Because of its convenience, we will introduce a function $|$ in the logical language that denotes the restriction as defined on assignments.

Definition 5.1 (Language for Perspective-based Value) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS and $\mathcal{M}$ a set of monadic value predicates. A* language for perspective-based value w.r.t. δ and $\mathcal{M}$ *is a first-order language with*

- *constants:*
  - *for each perspective p in $\mathcal{P}$ there is a constant $\underline{p}$,*
  - *for each attribute $x \in \mathcal{A}$, there is a constant $\underline{x}$,*
  - *for each assignment $s \in \mathcal{S}$, there is a constant $\underline{s}$,*
  - *for each monadic value predicate $m \in \mathcal{M}$, there is a constant $\underline{m}$.*
  - *for every subset X of $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{P}$, there is a constant $\underline{X}$*
- *function symbols: the binary function $|$ that denotes restriction*
- *predicates:*
  - *binary:* $\mathsf{satisfies}$, $\uparrow$, $\downarrow$, msr, *and* in
  - *ternary:* $\leq$
  - *quadruple:* eval

A language for perspective-based value contains no other formulae.

For ease of notation, if $\underline{x}$ is a constant, then we will write x. Furthermore, the constants $\underline{\mathcal{A}}$, $\underline{\mathcal{S}}$, $\underline{\mathcal{P}}$, $\underline{\mathcal{P}_{\mathsf{ag}}}$, $\underline{\mathcal{P}_{\mathsf{vl}}}$, $\underline{\mathcal{P}_{\mathsf{ob}}}$, $\underline{\mathcal{P}_{\mathsf{cr}}}$ and $\underline{\mathcal{M}}$ denote the sets $\mathcal{A}$, $\mathcal{S}$, $\mathcal{P}$, $\mathcal{P}_{\mathsf{ag}}$, $\mathcal{P}_{\mathsf{vl}}$, $\mathcal{P}_{\mathsf{ob}}$, $\mathcal{P}_{\mathsf{cr}}$ and $\mathcal{M}$

Table 5.1: *Abbreviations for Languages for Perspective-based Value*

Abbreviation	Of
$s <_p t$	$s \leq_p t \wedge \neg(t \leq_p s)$
$s \equiv_p t$	$s \leq_p t \wedge t \leq_p s$
$\mathsf{infl}(p, q)$	$p{\uparrow}q \vee p{\downarrow}q$
$\mathsf{good}(\alpha, s, p)$	$\mathsf{eval}(\alpha, s, \mathsf{good}, p)$
$\mathsf{bad}(\alpha, s, p)$	$\mathsf{eval}(\alpha, s, \mathsf{bad}, p)$
$\exists_{x \in X}[\phi]$	$\exists_x[\mathsf{in}(x, X) \wedge \phi]$
$\forall_{x \in X}[\phi]$	$\forall_x[\mathsf{in}(x, X) \supset \phi]$
$\exists_{x_1,\ldots,x_n \in X}[\phi]$	$\exists_{x_1,\ldots,x_n}[\mathsf{in}(x_1, X) \wedge \ldots \wedge \mathsf{in}(x_n, X) \wedge \phi]$
$\forall_{x_1,\ldots,x_n \in X}[\phi]$	$\forall_{x_1,\ldots,x_n}[\mathsf{in}(x_1, X) \wedge \ldots \wedge \mathsf{in}(x_n, X) \supset \phi]$
$X \subseteq X'$	$\forall_{x \in X}[\mathsf{in}(x, X')]$
$X \subset X'$	$X \subseteq X' \wedge \neg(X' \subseteq X)$

respectively. For example, instead of $\mathsf{in}(\underline{x}, \underline{\mathcal{A}})$ we will write $\mathsf{in}(x, \mathcal{A})$ and $\mathsf{in}(p, \mathcal{P}_{\mathsf{ob}})$ denotes that perspective p is in the set of perspectives representing intermediate objectives. Table 5.1 introduces several abbreviations to make notation easier.

The predicate $s \leq_p t$ denotes that the value of assignment t is as much or more than the value assignment s from perspective p. If $s \leq_p t$ is true, we will also say that t is weakly preferred to s from perspective p. The predicates $p \uparrow q$ and $p \downarrow q$ denote that perspective p positively / negatively influences perspective q respectively. The predicate $\mathsf{msr}(p, X)$ denotes that perspective p is measured by the set of attributes X. The predicate $\mathsf{eval}(\alpha, s, \mathsf{good}, p)$ denotes that agent α evaluates assignment s as 'good' from perspective p. Similarly, the predicate $\mathsf{eval}(\alpha, s, \mathsf{bad}, p)$ denotes that agent α evaluates assignment s as 'bad' from perspective p.

For example, let us consider $\forall_{p \in \mathcal{P}} \exists_{q \in \mathcal{P}}[\mathsf{infl}(p, q)]$, which denotes that every perspective positively or negatively influences at least one other perspective. First note that $\mathcal{P}$ is actually the constant $\underline{\mathcal{P}}$, which denotes the set of all perspectives. This formula is an abbreviation of $\forall_x[\mathsf{in}(x, \mathcal{P}) \supset \exists_y[\mathsf{in}(y, \mathcal{P}) \supset p{\uparrow}q \vee p{\downarrow}q]]$.

Example 5.1 (LPV) Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS with $\mathcal{A} = \{x, y\}$, $\mathcal{S} = \{s_1, \ldots, s_8\}$ and $\mathcal{P} = \langle \{\alpha\}, \{p, q\}, \emptyset, \emptyset \rangle$ a perspective structure and let μ be a MES for δ. If $\mathcal{L}_{\mathsf{pv}}$ is a language for perspective-based value for δ and μ, then the following expressions are some well-formed formulae of $\mathcal{L}_{\mathsf{pv}}$.

- $p \uparrow q \wedge q \downarrow \alpha$ denotes that perspective p positively influences perspective q and that perspective q negatively influences perspective α
- $\mathsf{in}(p, \mathcal{P})$ denotes that perspective p is a member of the set $\mathcal{P}$
- $s_1 <_p s_3$ denotes that assignment s_3 has more value from perspective p than assignment s_1
- $s_3 <_p s_1$ denotes that assignment s_1 has more value from perspective p than assignment s_3

We will now define a contrariness function (see Section 2.1) for languages for perspective-based value. Recall that if cf is a contrariness function, then formulae ϕ and ψ are called

contradictory if and only if $\phi \in \mathsf{cf}(\psi)$ and $\psi \in \mathsf{cf}(\phi)$. A contrariness function is used to determine the attacks between arguments.

Definition 5.2 (Contrariness Function for PV) *Let $\mathcal{L}_{\mathsf{pv}}$ be a language for perspective-based value. A* contrariness function for $\mathcal{L}_{\mathsf{pv}}$ *is a contrariness function* $\mathsf{cf}_{\mathsf{pv}} : \mathcal{L}_{\mathsf{pv}} \rightarrow 2^{\mathcal{L}_{\mathsf{pv}}}$ *with at least:*

- $s <_p t$ *is contradictory with* $t <_p s$, $t \leq_p s$, $s \equiv_p t$, *and* $t \equiv_p s$, *and*
- $\mathsf{good}(\alpha, s, p)$ *is contradictory with* $\mathsf{bad}(\alpha, s, p)$

If $\mathsf{cf}_{\mathsf{pv}}$ is a contrariness function for the LPV in Example 5.1, then $s_1 <_p s_3 \in \mathsf{cf}_{\mathsf{pv}}(s_3 <_p s_1)$ and $s_3 <_p s_1 \in \mathsf{cf}_{\mathsf{pv}}(s_1 <_p s_3)$. Because of the definitions of attack, an argument concluding $s_1 <_p s_3$ thus rebuts a defeasible argument concluding $s_3 <_p s_1$ on its conclusion.

Defeasible Rules to Infer Value

The defeasible inference rules in Table 5.2 formalize the argument schemes with respect to $\mathcal{L}_{\mathsf{pv}}$ that were proposed in the previous chapter. These defeasible inference rules can be used to reason about the value of assignments from different perspectives.

In this and the following chapters a number of defeasible inference rules will be introduced. Recall from Definition 2.4 that an inference rule is either strict or defeasible. A strict rule is of the form $s : \phi_1, \ldots, \phi_m \rightarrow \phi$ and a defeasible rule of the form $d : \phi_1, \ldots, \phi_m \Rightarrow \phi$. Here, s and d are called the rule names of the inference rules. Inference rules are usually formulated as rule schemas by the use of free variables. To refer to inference rules, we will use the following notation that is often used in nonmonotonic logic.

Notation 5.1 (Inference Rule Names) *Let $\phi_1, \ldots, \phi_m, \psi$ be formulae with $x_1, \ldots, x_n$ all free variables in them. The defeasible inference schema with antecedent $\phi_1, \ldots, \phi_m$ and conclusion ψ will have the rule name $d(x_1, \ldots, x_n)$. The defeasible inference rule that instantiates the schema with constants $c_1, \ldots, c_n$ will have the rule name $d(c_1, \ldots, c_n)$.*

If we say that an inference schema is in a set $\mathcal{R}$ inference rules, then we mean that all grounded instances of that schema are in $\mathcal{R}$.

Example 5.2 Consider the defeasible rule schema $d_n(x_1, x_2) : p_1(x_1), p_2(x_2) \Rightarrow p_3(x_1, x_2)$. Here, x_1 and x_2 are all free variables in the formulae in the schema. Consequently, the rule name is $d_n(x_1, x_2)$. For convenience, we will also refer to this schema as simply d_n. Defeasible schema d_n is a schema for all its grounded instances. For example, if c_1 and c_2 are constants, then this means that there is a defeasible inference rule $d_n(c_1, c_2) : p_1(c_1), p_2(c_2) \Rightarrow p_3(c_1, c_2)$.

Argument Scheme 6, 7, and 8 were proposed in Section 3.3.1 and use the influences between perspectives to propagate value. The first three defeasible inference schemas $d_{\uparrow}$, $d_{\downarrow}$ and $d_{\equiv}$ formalize these three argument schemes. For example, $d_{\uparrow}$ formalizes that if t has strictly more value than s from perspective p ($s <_p t$) and perspective p positively influences perspective q ($p \uparrow q$), then defeasibly it is true that t has strictly more value than s from perspective q ($s <_q t$).

In Argument Scheme 5, the structure of assignments is used to infer the relative value of two assignments from a perspective. Namely, if assignment t has strictly more value than assignment s from perspective p, then an assignment satisfies t also has strictly more value than an assignment that satisfies s from perspective p. The defeasible inference schema d_{gnlz} formalizes Argument Scheme 5. Four defeasible inference schemas are proposed that use transitivity to reason about value. The schemas d_{indf}, d_{si}, d_{is} and d_{strct} formalize Argument Schemes 1, 2, 3, and 4. These argument schemes are defeasible because transitivity is considered to be a controversial property as discussed in Subsection 3.2.1.

Table 5.2: *Defeasible Inference Schemas for Reasoning about Value*

Value from influence:	$d_{\uparrow}(p,q,s,t) : s <_p t, p \uparrow q \Rightarrow s <_q t$ $d_{\downarrow}(p,q,s,t) : s <_p t, p \downarrow q \Rightarrow t <_q s$ $d_{\equiv}(p,q,s,t) : s \equiv_p t, \mathsf{infl}(p,q) \Rightarrow s \equiv_q t$
Assignment structure:	$d_{gnlz}(p,s,t,s',t') : s <_p t, \mathsf{satisfies}(s',s), \mathsf{satisfies}(t',t) \Rightarrow s' <_p t'$
Transitivity:	$d_{indf}(p,s,t,u) : s \equiv_p t, t \equiv_p u \Rightarrow s \equiv_p u$ $d_{si}(p,s,t,u) : s <_p t, t \equiv_p u \Rightarrow s <_p u$ $d_{is}(p,s,t,u) : s \equiv_p t, t <_p u \Rightarrow s <_p u$ $d_{strct}(p,s,t,u) : s <_p t, t <_p u \Rightarrow s <_p u$

Given the language for perspective-based value, a suitable contrariness function, and the strict and defeasible inference rules proposed above, argumentation systems for perspective-based value are defined as follows.

Definition 5.3 (Argumentation System for PV) *Let δ be a PVCS. An* Argumentation System for Perspective-based Value (ASPV) for δ *is an argumentation system* $\langle \mathcal{L}_{\mathsf{pv}}, \mathcal{SR} \cup \mathcal{DR}_{\mathsf{pv}}, \mathsf{cf}_{\mathsf{pv}} \rangle$ *where*

- $\mathcal{L}_{\mathsf{pv}}$ *is an LPV for* δ*,*
- $\mathcal{SR}$ *is the set of all valid first-order inferences,*
- $\mathcal{DR}_{\mathsf{pv}}$ *contains the defeasible inference rules proposed in Table 5.2, and*
- $\mathsf{cf}_{\mathsf{pv}}$ *is a contrariness function for* $\mathcal{L}_{\mathsf{pv}}$*.*

Example 5.3 (Arguments in an ASPV) Let δ be the PVCS and $\mathcal{L}_{\mathsf{pv}}$ be the LPV of Example 5.1. Then $\mathcal{AS}_{\mathsf{pv}} = \langle \mathcal{L}_{\mathsf{pv}}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf}_{\mathsf{pv}} \rangle$ is a ASPV for δ. The following arguments A_1 and A_2 are arguments in $\mathcal{AS}_{\mathsf{pv}}$.

$$A_1 = \frac{s_1 <_p s_3 \quad p \uparrow q}{s_1 <_q s_3}\, d_{\uparrow} \qquad A_2 = \frac{s_1 <_q s_3 \quad \dfrac{s_3 \equiv_p s_4 \quad \dfrac{p \uparrow q}{\mathsf{infl}(p,q)}\, \vee I}{s_3 \equiv_q s_4}\, d_{\equiv}}{s_1 \equiv_q s_4}\, d_{si}$$

Argument A_1 applies defeasible inference rule $d_{\uparrow}$ to conclude that assignment s_3 has more value from perspective q than assignment s_1 because s_3 has more value from perspective p than s_1 and p positively influences q. Defeasible inference rules d_{si} and $d_{\equiv}$ and strict inference rule $\vee$ are applied in argument A_2. Recall that $\mathsf{infl}(p,q)$ abbreviates $p \uparrow q \vee p \downarrow q$.

5.1.2 Argumentation Theories

Arguments are constructed from a knowledge base. Furthermore, arguments may differ in strength and can thus be ordered by strength. An argumentation theory consists of an argumentation system, a knowledge base and an argument ordering. We will first define what knowledge bases should be used in argumentation theories for perspective-based value. A knowledge base consists of a set of necessary premises, a set of ordinary premises, and a set of assumption premises. Table 5.3 proposes several axioms that should be in the necessary premises in every knowledge base for perspective-based value.

Axioms $\mathtt{ax}_1$ and $\mathtt{ax}_2$ concern value comparisons. Axiom $\mathtt{ax}_1$ ensures that value comparisons are reflexive. Axiom $\mathtt{ax}_2$ formalizes Constraint 1 and ensures that value from a perspective only depends on the attributes that measure that perspective. Axioms $\mathtt{ax}_3$ till $\mathtt{ax}_8$ are introduced concerning influence between perspectives. Axiom $\mathtt{ax}_3$ ensures that influence is reflexive. Axiom $\mathtt{ax}_4$ formalizes Constraint 2, which ensures consistency between the attributes that measure value from perspectives and influence. Axioms $\mathtt{ax}_5$, $\mathtt{ax}_6$, $\mathtt{ax}_7$, and $\mathtt{ax}_8$ concern transitivity of influence and formalize Constraints 3, 4, 5, and 6, respectively.

Axioms $\mathtt{ax}_9$ till $\mathtt{ax}_{14}$ concern monadic evaluations. Axiom $\mathtt{ax}_9$ states that if a perspective does not influence value from an agent's perspective, i.e., the agent does not care about that perspective, then that agent cannot evaluate any assignment from that perspective. Axiom $\mathtt{ax}_{10}$ states that if two assignments have an equal amount of value from perspective p and an agent evaluates one as m from p, then that agent should also evaluate the other as m from p. Axioms $\mathtt{ax}_{11}$, $\mathtt{ax}_{12}$, $\mathtt{ax}_{13}$, and $\mathtt{ax}_{14}$ formalize Constraints 8 and 9 in Subsection 3.4.1. Note that four axioms are required to formalize these two constraint because each constraint needs to be formalized for the value predicates good and bad. Finally, it is not necessary to include an axiom for Constraint 10 concerning continuity because Proposition 4.8 proves that it holds for positive and negative value predicates and we do not use any other value predicates.

Table 5.3: *Axioms for Perspective-based Value*

Name	Axiom[a]
$\mathtt{ax}_1$	$\forall_{s\in\mathcal{S}}\forall_{p\in\mathcal{P}}[s \leq_p s]$
$\mathtt{ax}_2$	$\mathsf{msr}(p,X) \supset \forall_{s,t\in\mathcal{S}}[s\,|_X \leq_p t\,|_X \leftrightarrow s \leq_p t]$
$\mathtt{ax}_3$	$\forall_{p\in\mathcal{P}}[\neg\mathsf{infl}(p,p)]$
$\mathtt{ax}_4$	$\mathsf{infl}(p,q) \wedge \mathsf{msr}(p,X) \wedge \mathsf{msr}(q,Y) \supset X \subseteq Y$
$\mathtt{ax}_5$	$p{\uparrow}q \wedge q{\uparrow}r \supset p{\uparrow}r$
$\mathtt{ax}_6$	$p{\uparrow}q \wedge q{\downarrow}r \supset p{\downarrow}r$
$\mathtt{ax}_7$	$p{\downarrow}q \wedge q{\uparrow}r \supset p{\downarrow}r$
$\mathtt{ax}_8$	$p{\downarrow}q \wedge q{\downarrow}r \supset p{\uparrow}r$
$\mathtt{ax}_9$	$\forall_{p\in\mathcal{P}}\forall_{\alpha\in\mathcal{P}_{\mathsf{ag}}} : \neg\mathsf{infl}(p,\alpha) \supset \forall_{s\in\mathcal{S}}\forall_{m\in\mathcal{M}}[\neg\mathsf{eval}(\alpha,s,m,p)]$
$\mathtt{ax}_{10}$	$\mathsf{eval}(\alpha,s,m,p) \wedge s \equiv_p t \supset \mathsf{eval}(\alpha,t,m,p)$
$\mathtt{ax}_{11}$	$p{\uparrow}\alpha \wedge \mathsf{good}(\alpha,s,p) \wedge s \leq_p t \supset \mathsf{good}(\alpha,t,p)$
$\mathtt{ax}_{12}$	$p{\downarrow}\alpha \wedge \mathsf{good}(\alpha,s,p) \wedge t \leq_p s \supset \mathsf{good}(\alpha,t,p)$
$\mathtt{ax}_{13}$	$p{\uparrow}\alpha \wedge \mathsf{bad}(\alpha,s,p) \wedge t \leq_p s \supset \mathsf{bad}(\alpha,t,p)$
$\mathtt{ax}_{14}$	$p{\downarrow}\alpha \wedge \mathsf{bad}(\alpha,s,p) \wedge s \leq_p t \supset \mathsf{bad}(\alpha,t,p)$

[a] All formulae with free variables are implicitly universally quantified.

A language for PV is based on a PVCS. The language for PV has constants denoting

all elements of a PVCS and several predicates are used to describe relations between these constants. These predicates need to be instantiated in a knowledge base according with the PVCS. For example, an LPV has constants that denote assignments, constants that denote sets of assignments, and a predicate in that denotes membership of a set. A knowledge base for PV must be instantiated with when $\mathsf{in}(\underline{x}, \underline{X})$ is true. Namely, $\mathsf{in}(\underline{x}, \underline{X})$ is true iff $\underline{x}$ denotes element x, $\underline{X}$ denotes set X and $x \in X$. Furthermore, an LPV has the binary predicate satisfies. Recall from Section 4.1 that if s and t are assignments and $s \subseteq t$, then we say that s satisfies t. The knowledge base should contain $\mathsf{satisfies}(\underline{s}, \underline{t})$ iff $s \subseteq t$ is true for the assignments s and t.

Definition 5.4 (Knowledge Base for PV) *Let $\mathcal{AS}_{\mathsf{pv}}$ be an argumentation system for PV.* A knowledge base for PV in $\mathcal{AS}_{\mathsf{pv}}$ *is a knowledge base $\langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ in $\mathcal{AS}_{\mathsf{pv}}$ such that*

- $\mathcal{K}_{\mathsf{np}}$ *contains the axioms in Table 5.3,*
- $s, t \in \mathcal{S}$ *and* $s \subseteq t$ *iff* $\mathsf{satisfies}(\underline{s}, \underline{t}) \in \mathcal{K}_{\mathsf{op}}$*, and*
- $x \in X$ *iff* $\mathsf{in}(\underline{x}, \underline{X}) \in \mathcal{K}_{\mathsf{op}}$ *(with X a subset or equal to $\mathcal{A}$, $\mathcal{S}$, or $\mathcal{P}$).*

We will now look at when a knowledge base for PV corresponds to the structures in the Perspective-based Value Model that we have proposed in Chapter 4. Let $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ be a knowledge base for PV. Then $\mathcal{K}$ corresponds to

- PVCS $\langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ iff the following two conditions hold:
  1. $(\underline{s} \leq_{\underline{p}} \underline{t}) \in \mathcal{K}_{\mathsf{op}}$ iff $(p, s, t) \in \leq$, and,
  2. $\mathsf{msr}(\underline{p}, \underline{X}) \in \mathcal{K}_{\mathsf{op}}$ iff $\mathsf{msr}(p) = X$.
- Perspective Influence Structure $\langle \mathcal{P}, I_{\uparrow}, I_{\downarrow} \rangle$ iff the following two conditions hold:
  1. $p{\uparrow}q \in \mathcal{K}_{\mathsf{op}}$ iff $(p, q) \in I_{\uparrow}$, and,
  2. $p{\downarrow}q \in \mathcal{K}_{\mathsf{op}}$ iff $(p, q) \in I_{\downarrow}$.
- MES $\langle \mathcal{M}, \mathsf{eval} \rangle$ iff the following condition holds: $\mathsf{eval}(\underline{\alpha}, \underline{s}, \underline{m}, \underline{p}) \in \mathcal{K}_{\mathsf{op}}$ iff $(\alpha, s, m, p) \in \mathsf{eval}$.

To satisfy the rationality postulates described in Subsection 2.1.5, the closure of the set of necessary premises $\mathcal{K}_{\mathsf{np}}$ under strict rule application needs to be consistent.

Proposition 5.1 *Let $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ be a knowledge base for PV. The closure of $\mathcal{K}_{\mathsf{np}}$ under strict rule application is consistent.*

Proof To prove that the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent we need to find a model in which all formulae in $\mathcal{K}'_{\mathsf{np}}$ are true and to show that applying strict rules does not lead to inconsistencies. We will now construct such a model $M = \langle \langle D, R, F \rangle, I \rangle$ with I the interpretation function, $D = \{o_\alpha, o_s, o_{x_1}, o_{\mathcal{S}}, o_{\mathcal{P}}, o_{\mathcal{A}}, o_{\emptyset}, o_{\mathsf{good}}, o_{\mathsf{bad}}\}$ the set of objects, $R = \{\uparrow, \downarrow, \mathsf{msr}, \mathsf{in}, \mathsf{satisfies}, \leq, \mathsf{eval}\}$ the set of relations, and the restriction function $|$ the only function in F.

We will use PVCS $\delta = \langle \{o_{x_1}\}, \{o_s\}, \mathcal{P}, \leq', \mathsf{msr}' \rangle$ with $\mathcal{P} = \langle \{o_\alpha\}, \emptyset, \emptyset, \emptyset \rangle$, $\mathsf{msr}'(o_\alpha) = \{o_{x_1}\}$, and $(o_s, o_s, o_\alpha) \in \leq'$. In the LPV w.r.t. PVCS δ and no monadic value predicates, there are 'normal' constants and constants that denote sets. The normal constants are interpreted as $I(s) = o_s$, $I(x_1) = o_{x_1}$, $I(\alpha) = o_\alpha$, $I(\mathsf{good}) = o_{\mathsf{good}}$, and $I(\mathsf{bad}) = o_{\mathsf{bad}}$. The constants

denoting sets are interpreted as follows. Because the set of assignments only consists of assignment s, the constant $\mathcal{S}$ is interpreted the same as the constant $\{s\}$: $I(\mathcal{S}) = I(\{s\}) = o_{\mathcal{S}}$. Similarly, because there is only one attribute x_1 and one perspective α, we have the following interpretations: $I(\mathcal{A}) = I(\{x_1\}) = o_{\mathcal{A}}$ and $I(\mathcal{P}) = I(\{\alpha\}) = o_{\mathcal{P}}$. Finally, there is a constant $\underline{\emptyset}$ denoting the empty set: $I(\underline{\emptyset}) = o_{\emptyset}$. The LPV has the predicates $\uparrow$, $\downarrow$, msr, in, $\mathsf{satisfies}$, $\leq$ and eval, which are interpreted as follows: (1) both $\uparrow$ and $\downarrow$ are empty, i.e., $I(\uparrow) = I(\downarrow) = \emptyset$; (2) perspective α is measured by all attributes, i.e., $I(\mathsf{msr}) = \{(o_\alpha, o_{\mathcal{A}})\}$; (3) $I(\mathsf{in}) = \{(o_s, o_{\mathcal{S}}), (o_{x_1}, o_{\mathcal{A}}), (o_\alpha, o_{\mathcal{P}})\}$; (4) $I(\mathsf{satisfies}) = \emptyset$; (5) $I(\leq) = \{(o_\alpha, o_s, o_s)\}$; and, (6) $I(\mathsf{eval}) = \emptyset$. Finally, the interpretation of the 2-ary restriction function $|$ maps $(o_s, o_{\mathcal{A}})$ to o_s.

We will now show that all formulae in $\mathcal{K}_{\mathsf{np}}$ (i.e., all axioms in Table 5.3) are true in M. Axiom $\mathtt{ax}_1$ is true in M because there is only one assignment s and one perspective α and $s \leq_\alpha s$ is true. Axiom $\mathtt{ax}_2$ is also true in M because $\mathsf{msr}(\alpha, \mathcal{A})$ is true and because $s|_{\mathcal{A}} = s$ and thus $s \leq_\alpha s \leftrightarrow s \leq_\alpha s$ is true. Because infl is never true in M, axiom $\mathtt{ax}_3$ is also true in M. Axioms $\mathtt{ax}_4$, $\mathtt{ax}_5$, $\mathtt{ax}_6$, $\mathtt{ax}_7$, and $\mathtt{ax}_8$ are all true because none of their antecedents can be made true. Axiom $\mathtt{ax}_9$ is true in M because eval is never true in M. Finally, axioms $\mathtt{ax}_{10}$, $\mathtt{ax}_{11}$, $\mathtt{ax}_{12}$, $\mathtt{ax}_{13}$, and $\mathtt{ax}_{14}$ are all true because none of their antecedents can be made true.

Because all formulae in $\mathcal{K}'_{\mathsf{np}}$ are universally quantified, the inference rule universal generalization can be applied freely without adding (semantically) new formulae. Also material implication can be applied freely. Namely, the only formula with a material implication in $\mathcal{K}_{\mathsf{np}}$ whose antecedent is true is $\mathtt{ax}_2$. Because $\mathsf{msr}(\alpha, \mathcal{A})$ is true, s is the only assignment and $s \leq_\alpha s$ is true, the right side of $\mathtt{ax}_2$ does not introduce inconsistencies. Because we can find a model for $\mathcal{K}_{\mathsf{np}}$ and strict rule application does not lead to inconsistencies, the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent.

Now we have defined all elements of an argumentation theory for perspective-based value (PV).

Definition 5.5 (Argumentation Theory for PV) *Let δ be a PVCS, ι a PIS, and μ a MES. An* Argumentation Theory for Perspective-based Value (ATPV) w.r.t. δ and μ *is an argumentation theory $\langle \mathcal{AS}_{\mathsf{pv}}, \mathcal{K}, \leq \rangle$ where*

- *$\mathcal{AS}_{\mathsf{pv}}$ is an ASPV for δ and μ,*
- *$\mathcal{K}$ is a knowledge base for PV in $\mathcal{AS}_{\mathsf{pv}}$ that corresponds to δ, ι, and μ, and*
- *$\leq$ is an argument ordering over arguments in $\mathcal{AS}_{\mathsf{pv}}$ and satisfies the last-link or weakest-link principle.*

Recall from Subsection 2.1.5 that an argumentation theory $\mathcal{AT}$ satisfies the rationality postulates iff the following conditions hold: (1) $\mathcal{AT}$ is well-formed as in Definition 2.27; (2) $\mathcal{SR}$ is closed under transposition or contraposition; (3) $\leq$ follows the last-link or weakest-link principle; and, (4) the closure of $\mathcal{K}_{\mathsf{np}}$ under strict-rule application is consistent. We will now show that argumentation theories for PV satisfy the rationality postulates.

Proposition 5.2 *Let $\mathcal{AT}$ be an argumentation theory for PV. Then $\mathcal{AT}$ satisfies the rationality postulates in Subsection 2.1.5.*

Proof Because there are no contraries in the contrariness function for PV, $\mathcal{AT}$ is well-formed. By Definition 5.3 of an ASPV, the strict rules are the first-order strict rules, which

satisfy the second condition. By Definition 5.5, $\leq$ satisfies the last-link or weakest-link principle. Finally, by Proposition 5.1, the closure of $\mathcal{K}_{\mathsf{np}}$ under strict-rule application is consistent. Consequently, argumentation theories for PV satisfy the rationality postulates in Subsection 2.1.5. ■

The relative importances of perspectives that influence a perspective, e.g., value v_1 is more important to agent α than value v_2, can be used to reason about the strength of arguments. Section 2.3 proposed a meta-level argumentation framework to reason about the relative strength of object-level arguments. Evaluation of the meta-level arguments is used to determine the grounded argument ordering over object-level arguments as defined in Definition 2.41. In the next section, the meta-level argumentation framework of Section 2.3 is used in combination with the relative importance of perspectives to determine the relative strength of object-level arguments. If this meta-level approach is used, then the grounded argument ordering of the meta-level argumentation framework should be used as the argument ordering of the object-level argumentation theory for PV. In other words, if $\langle \mathcal{AS}_{\mathsf{pv}}, \mathcal{K}, \leq \rangle$ is an argumentation theory for PV, then $\leq$ should be the grounded argument ordering induced by the meta-level argumentation framework.

Because an argumentation theory for PV is a normal argumentation theory, an argumentation framework for it can be built in the normal way as described in Definition 2.21. This AF can be used to determine which arguments are justified, defensible and overruled.

Proposition 5.3 *Let $\mathcal{AT}$ be the ATPV w.r.t. PVCS $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ and MES μ. If we have $\delta' = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq', \mathsf{msr}' \rangle$ with*

- *$\underline{s} \leq_{\underline{p}} \underline{t}$ is a justified conclusion of $\mathcal{AT}$ iff $(p, s, t) \in \leq'$, and*
- *$\mathsf{msr}(\underline{p}, \underline{X})$ is a justified conclusion of $\mathcal{AT}$ iff $\mathsf{msr}'(p) = X$.*

then δ' is a PVCS.

Proof To show that δ' is a PVCS, we need to show that for all $s, t \in \mathcal{S}$ and $p \in \mathcal{P}$: (1) $s \leq'_p s$; and, (2) $s|_{\mathsf{msr}(p)} \leq'_p t|_{\mathsf{msr}(p)}$ iff $s \leq'_p t$.

Condition 1: because $\mathcal{AT}$ satisfies the rationality postulates, every strict and firm argument is justified. Using axiom $\mathtt{ax}_1$, a strict and firm argument can be constructed concluding $s \leq_p s$ for every assignment in $\mathcal{S}$ and every perspective p. Consequently, $s \leq_p s$ is a justified conclusion and thus it is true that $s \leq'_p t$.

Condition 2: if $\mathsf{msr}(p, X)$ is a justified conclusion, then $\mathtt{ax}_2$ can be used to construct a justified argument concluding that $s \mid_X \leq_p t \mid_X \leftrightarrow s \leq_p t$ for all assignments s and t. If $\mathsf{msr}(p, X)$ is a justified conclusion, then $\mathsf{msr}'(p) = X$ in δ'. Consequently, the second condition is true. ■

Note that if $\mathcal{AT}$ is an argumentation theory w.r.t. PVCS δ, then the PVCS δ' based on $\mathcal{AT}$ is not necessarily the same as δ. Suppose that δ is a PVCS with two assignments s and t and two perspectives p and q such that $s <_p t$, but it is not known how s and t compare from perspective q. Furthermore, let perspective p positively influence q in the argumentation theory corresponding to δ. Then defeasible rule $d_{\uparrow}$ can be used to infer that $s <_q t$, which is a justified conclusion in the PVCS based on $\mathcal{AT}$. Consequently, it was not known how s and t compare from q in PVCS δ, but in the PVCS based on the argumentation theory

corresponding to δ, it was inferred that $s <_q t$. Summarizing, using the argumentation framework it is possible to infer new value comparisons.

When more than one perspective influences some perspective q it is possible that several arguments can be constructed that conclude $s <_q t$ for some assignments s and t. In the same way a number of arguments can be constructed concluding $t <_q s$. In such scenarios it is useful to accrue arguments for the same conclusion. To do this, argumentation theories for PV should be transformed into an accrual argumentation theories as described in Definition 2.35.

5.2 Meta-Level Argumentation

Since a perspective p can be influenced by multiple perspectives, it is not uncommon that there are several arguments concluding $s <_p t$ because of how the value of s and t compare from the perspectives that influence p. For the same reason there may also be several arguments that conclude $t <_p s$. Note that these kinds of arguments rebut each other and that they do not differ in strength. Therefore, this conflict cannot be resolved. Consequently, the arguments concluding $s <_p t$ and $t <_p s$ will not be justified or overruled, but defensible.

As explained in Subsection 3.3.3, not all perspectives that influence a perspective p are equally important for p. The relative importance of perspectives for a perspective can be used to determine the relative strength of arguments, which can be used to determine what attacks are successful. If some attacks are successful and others are unsuccessful, then it is likely that either the arguments concluding $s <_p t$ or the those concluding $t <_p s$ become justified.

Reasoning about the relative strengths of arguments is reasoning *about* arguments. This kind of reasoning is therefore done on a meta-level w.r.t. those arguments. In this section we will use the meta-level argumentation framework described in Section 2.3 to use the relative importance of perspectives to reason about the relative strength of object-level arguments in an ASPV. To do so, we first need to describe the meta-level language, contrariness function and defeasible rules that we will use. Next, Subsection 5.2.2 proposes several additional meta-level axioms and defines meta-level knowledge bases and argumentation theories. With this machinery, the meta-level argumentation framework for PV can be used to determine the grounded argument ordering of object-level arguments in the ASPV.

5.2.1 Meta-Argumentation System for PV

A meta-language for PV is a meta-language as defined in Definition 2.36 that is extended with constants for perspectives and a ternary predicate $\trianglelefteq$ denoting the relative importance of perspectives for a perspective. To separate the object-level clearly from the meta-level, we will use an apostrophe to denote that an element is on the meta-level. For example, if $\mathcal{L}_{\mathsf{pv}}$ is the object-language, then $\mathcal{L}'_{\mathsf{pv}}$ is the language on the meta-level w.r.t. $\mathcal{L}_{\mathsf{pv}}$.

Definition 5.6 (Meta-Language for PV) *Let $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$ be a PVCS, μ a MES and $\mathcal{AS}_{\mathsf{pv}}$ be an ASPV for δ and μ. A* meta-language for PV w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ *is a meta-language w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ extended with the following elements:*

- *for each perspective p in $\mathcal{P}$ there is a constant $\underline{p}$,*
- *there is a constant $\underline{\mathcal{P}}$, and*
- *ternary predicate $\trianglelefteq$*

As in previous chapters, if $\underline{x}$ is a constant, then we will write simply x. Furthermore, the constant $\underline{\mathcal{P}}$ denotes the set of all perspectives. For example, instead of $\mathsf{in}(\underline{p}, \underline{\mathcal{P}})$, we will write $\mathsf{in}(p, \mathcal{P})$. We will write $p \trianglelefteq_r q$, which denotes that p is weakly more important for r than q. Furthermore, $\mathsf{in}(\underline{x}, \underline{X})$ denotes that x is a member of the set X. The abbreviations used for meta-languages are also used in meta-languages for PV. These abbreviations can be found in Table 2.3 in Section 2.3.1. Additionally, we will use $p \triangleleft_r q$ to abbreviate $p \trianglelefteq_r q \wedge \neg(q \trianglelefteq_r p)$. Recall from Subsection 2.3.1 that the meta-language contains function symbols for each function defined on object-arguments. For example, this means that $\phi \in \mathsf{premises}(A)$ and $\mathsf{rules}(B) \subset \mathsf{rules}(A)$ are expressions in the meta-language.

Because the meta-language for PV contains a new predicate $\trianglelefteq$, the contrariness function is adapted as follows.

Definition 5.7 (Contrariness Function for Meta-LPV) *Let $\mathcal{L}'_{\mathsf{pv}}$ be a meta-language for PV. A* contrariness function for meta PV *is a meta-contrariness function $\mathsf{cf}'_{\mathsf{pv}}$ such that for all p, q, r: $p \triangleleft_r q \in \mathsf{cf}'_{\mathsf{pv}}(q \triangleleft_r p)$.*

We will now introduce meta-argumentation systems for PV to reason about the strength of object-level arguments. Table 5.4 proposes a number of meta-level defeasible inference rules that will be used to reason about the relative strength of arguments. Subsection 3.3.3 explains that the influence of perspectives on a perspective can differ in importance. Assume that agent α's perspective is influenced by the perspectives 'costs' and 'fun' and that α cares more about fun than about costs. This means that inferring how α prefers s and t from how s and t compare from the perspective of 'fun' creates more conclusive force than inferring this from how s and t compare from the perspective of 'costs'. This intuition is captured in Argument Scheme 9, which says that if perspective p is more important than perspective q for perspective r, then inferring value from p to r is stronger than inferring value from q to r. Because there are four combinations of how p and q can influence r, Argument Scheme 9 is formalized by the first four meta-level defeasible inference schemas[1] $d'_{\uparrow\uparrow}$, $d'_{\uparrow\downarrow}$, $d'_{\downarrow\uparrow}$ and $d'_{\downarrow\downarrow}$. These schemas defeasibly infer the relative strength of applications of the object-level defeasible inference rules $d_\uparrow$ and $d_\downarrow$.

Table 5.4: *Defeasible Inference Schemas for Meta-Argumentation*

Influence	$d'_{\uparrow\uparrow}(p,q,r,s,t) : p \triangleleft_r q$	$\Rightarrow$	$\{d_\uparrow(p,r,s,t)\} \prec_{\mathcal{R}} \{d_\uparrow(q,r,s,t)\}$
	$d'_{\uparrow\downarrow}(p,q,r,s,t) : p \triangleleft_r q$	$\Rightarrow$	$\{d_\uparrow(p,r,s,t)\} \prec_{\mathcal{R}} \{d_\downarrow(q,r,s,t)\}$
	$d'_{\downarrow\uparrow}(p,q,r,s,t) : p \triangleleft_r q$	$\Rightarrow$	$\{d_\downarrow(p,r,s,t)\} \prec_{\mathcal{R}} \{d_\uparrow(q,r,s,t)\}$
	$d'_{\downarrow\downarrow}(p,q,r,s,t) : p \triangleleft_r q$	$\Rightarrow$	$\{d_\downarrow(p,r,s,t)\} \prec_{\mathcal{R}} \{d_\downarrow(q,r,s,t)\}$

Given the meta-language for PV, a contrariness function for PV and the defeasible inference rules described in Table 5.4, meta-argumentation systems for PV are defined as follows.

Definition 5.8 (Meta-Argumentation System for PV) *Let $\mathcal{AS}_{\mathsf{pv}}$ be an argumentation system for perspective-based value. A* meta-argumentation system for PV w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ *is a meta-argumentation system $\langle \mathcal{L}'_{\mathsf{pv}}, \mathcal{SR} \cup \mathcal{DR}'_{\mathsf{pv}}, \mathsf{cf}'_{\mathsf{pv}} \rangle$ with:*

- *$\mathcal{L}'_{\mathsf{pv}}$ is a meta-language w.r.t. $\mathcal{AS}_{\mathsf{pv}}$,*

[1] Recall from Notation 5.1 the notation that we use for defeasible inference schemas and rules.

- $\mathsf{cf}'_{\mathsf{pv}}$ *is a contrariness function for* $\mathcal{L}'_{\mathsf{pv}}$,
- $\mathcal{SR}$ *is the set of all valid first-order inferences, and*
- $\mathcal{DR}'_{\mathsf{pv}}$ *contains the defeasible inference rules in Table 5.4.*

5.2.2 Meta-Argumentation Theories

Relative importance of perspective orders perspectives from the point of view of a perspective. To ensure that the $\trianglelefteq$ predicate in the meta-language behaves correctly, Table 5.5 contains two axioms for meta-argumentation for perspective-based value. Axioms $\mathtt{ax}'_1$ and $\mathtt{ax}'_2$ ensure that relative importance is reflexive and transitive respectively. If a perspective p does not influence another perspective q, then p is of no importance to q.

Table 5.5: *Axioms for Meta-Argumentation for Perspective-based Value*

Name	Axiom[a]
$\mathtt{ax}'_1$	$p_1 \trianglelefteq_q p_2 \wedge p_2 \trianglelefteq_q p_3 \supset p_1 \trianglelefteq_q p_3$
$\mathtt{ax}'_2$	$\forall_{p,q \in \mathcal{P}}[p \trianglelefteq_q p]$

[a] All formulae with free variables are universally quantified.

In a particular domain it may be so that inferring value using the defeasible rules w.r.t. transitivity creates more conclusive force than inferring value using the defeasible rules w.r.t. value and influence. For example, $\{d_{\uparrow}(p,q,s,u)\} \prec_{\mathcal{R}} \{d_{strct}(p,s,t,u)\}$ could be used as an axiom to formalize that inferring value comparison using strict transitivity creates more conclusive force than inferring value comparison using how perspectives influence each other. However, because these preferences may differ in different domains, we will not enforce axioms that ensure this. If meta-argumentation theories are used in a specific domain, the designers should consider adding such axioms.

Definition 5.9 (Meta-Knowledge Base for PV) *A* meta-knowledge base for PV *(meta-KBPV) is a meta-knowledge base* $\langle \mathcal{K}'_{\mathsf{np}}, \mathcal{K}'_{\mathsf{op}}, \mathcal{K}'_{\mathsf{as}} \rangle$ *such that* $\mathcal{K}'_{\mathsf{np}}$ *contains all axioms in Table 2.4 and Table 5.5.*

A meta-KBPV $\langle \mathcal{K}'_{\mathsf{np}}, \mathcal{K}'_{\mathsf{op}}, \mathcal{K}'_{\mathsf{as}} \rangle$ corresponds to an Influence Importance Ordering $\trianglelefteq$ (as defined in Definition 4.12) iff the following condition holds: $(p, p', q) \in \trianglelefteq$ iff $(p \trianglelefteq_q p') \in \mathcal{K}'_{\mathsf{op}}$. To make sure that the conclusions originating from a meta-knowledge base for PV follow the rationality postulates that are described in Section 2.1.5, we need to show that the following property holds.

Proposition 5.4 *Let* $\langle \mathcal{K}'_{\mathsf{np}}, \mathcal{K}'_{\mathsf{op}}, \mathcal{K}'_{\mathsf{as}} \rangle$ *be a meta-KBPV. The closure of* $\mathcal{K}'_{\mathsf{np}}$ *under strict rule application is consistent.*

Proof To prove that the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent we need to find a model in which all formulae in $\mathcal{K}'_{\mathsf{np}}$ are true and to show that applying strict rules does not lead to inconsistencies. Because meta-KBPVs extend 'normal' meta-knowledge-bases as in Definition 2.39, we will now extend the model $M = \langle \langle D, R, F \rangle, I \rangle$ in Proposition 2.1. Let $M' = \langle \langle D', R', F' \rangle, I' \rangle$ be a model with $D' = D \cup \{o_p, o_{\mathcal{P}}\}$, $R' = R \cup \{\preceq\}$, and

I' the interpretation function that extends I with $I'(p) = o_p$, $I'(\mathcal{P}) = o_{\mathcal{P}}$, and $I'(\trianglelefteq) = \{(o_p, o_p, o_p)\}$.

In M', $\mathtt{ax}'_1$ is true because its antecedent cannot be made true and $\mathtt{ax}'_2$ is also true because $(o_p, o_p, o_p) \in I'(\trianglelefteq)$. The axioms in Table 2.4 are true in M' because M' is extended with two constants and a relation that are not used in the axioms in Table 2.4.

Because all formulae in $\mathcal{K}'_{\mathsf{np}}$ are universally quantified, the inference rule universal generalization can be applied freely without adding (semantically) new formulae. Also material implication can be applied freely (and vacuously for members of $\mathcal{K}'_{\mathsf{np}}$, for which the antecedent of all material implications is always false). Consequently, the closure of $\mathcal{K}'_{\mathsf{np}}$ under strict rule application is consistent. ■

Definition 5.10 (Meta-Argumentation Theory for PV) *Let $\mathcal{AS}_{\mathsf{pv}}$ be an argumentation system for PV and $\mathcal{K}$ a knowledge base in $\mathcal{AS}_{\mathsf{pv}}$.* A meta-argumentation theory for PV w.r.t. $\mathcal{AS}$ and $\mathcal{K}$ *is a meta-argumentation theory $\langle \mathcal{AS}'_{\mathsf{pv}}, \mathcal{K}', \leq \rangle$ w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ and $\mathcal{K}$ such that*

- *$\mathcal{AS}'_{\mathsf{pv}}$ is a meta-argumentation system for PV w.r.t. $\mathcal{AS}_{\mathsf{pv}}$, and*
- *$\mathcal{K}'$ is a meta-KBPV, and*
- *$\leq$ is an argument ordering that satisfies the last-link or weakest-link principle.*

Note that because the contrariness in a meta-argumentation theory for PV (meta-ATPV) has no contraries, meta-ATPVs are well-formed. Consequently, meta-ATPVs satisfy the rationality postulates in Section 2.1.5. This means that the grounded argument ordering of a meta-ATPV satisfies the last-link principle. If the grounded argument ordering of a meta-ATPV is thus used in an object-level ATPV, then the rationality postulates are also satisfied at the object-level.

5.3 Running Example

In this section we will demonstrate how the argumentation mechanism of this chapter can be used to reason about what assignment to prefer in the context of the running example from Chapter 1. Recall that in the running example a fire commander student is in a situation where a factory is on fire and there is a number of victims inside the building. The student has to decide what course of action to take.

We will first look into the object-level argumentation system and knowledge base that correspond to the example's PVCS δ and MES μ as described in Section 4.4. We will then show several interesting arguments that can be constructed from this knowledge base and see what arguments attack each other. To determine what arguments are acceptable, we need the relative strengths of the arguments. For this purpose, we will investigate the meta-argumentation system and meta-knowledge base that correspond to the example's PVCS and relative importance between perspectives. From this, several meta-arguments can be constructed that induce an object-level argument ordering. This ordering is then used to determine what object-level attacks are successful and to determine what conclusions are acceptable.

Argumentation about Perspective-based Value

In Section 4.4, the running example was formalized in the PVCS $\delta = \langle \mathcal{A}, \mathcal{S}, \mathcal{P}, \leq, \mathsf{msr} \rangle$. Tables 5.6a and 5.6b recall from the previous chapter what every attribute and perspective

denotes. The set of attributes $\mathcal{A}$ by which outcomes of decisions are described and the perspective structure $\mathcal{P}$ are thus as follows.

$$\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$
$$\mathcal{P} = \langle\ \{\alpha\},\ \{v_1, v_2, v_3\},\ \{o_1, o_2, o_3\},\ \{c_1, c_2, c_3, c_4, c_5, c_6\}\ \rangle$$

Also recall that the set of monadic value predicates $\mathcal{M}$ is $\{\mathsf{good}, \mathsf{bad}\}$.

Table 5.6: *What the attributes and perspectives denote (as in Table 4.1 and 4.2)*

(a) *Attributes*

	Denotes
x_1	Number of minutes that victims are inside
x_2	Number of minutes that personnel is inside
x_3	Amount of smoke
x_4	Amount of chemicals leaked into earth
x_5	Number of minutes that victims are near fire
x_6	Number of minutes that personnel is near fire

(b) *Perspectives*

	Denotes
α	The student's perspective
v_1	Universalism
v_2	Security
v_3	Benevolence
o_1	Impact on the environment
o_2	Safety of victims
o_3	Safety of personnel
c_1	Minutes victims are inside
c_2	Minutes personnel is inside
c_3	Amount of smoke
c_4	Amount of chemicals leaked into earth
c_5	Minutes victims are near fire
c_6	Minutes personnel is near fire

To be able to argue about what decision to take, we will now show how the argumentation framework introduced in this chapter (which is based on δ) can be used in the running example. We will first introduce the object-level logical language, argumentation system, and knowledge base. We will then introduce the meta-level argumentation system and meta-level knowledge base, which is used to argue about the relative strengths of object-level arguments. Given the outcome of this meta-level argumentation, we will determine what object-level arguments are acceptable.

Following Definition 5.1, the language for perspective-based value w.r.t. δ and $\mathcal{M}$ is the first-order language $\mathcal{L}_{\mathsf{pv}}$. This means that $\mathcal{L}_{\mathsf{pv}}$ contains a constant for each perspective, attribute, assignment, monadic value predicate, and a constant for each set of perspectives, attributes, assignments, and monadic value predicates. Using this language $\mathcal{L}_{\mathsf{pv}}$ for LPV, the argumentation system for PV is then the tuple $\langle \mathcal{L}_{\mathsf{pv}}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf} \rangle$ (as in Definition 5.3).

Now that we have instantiated the ASPV, we will instantiate the knowledge base that corresponds to the structures that we defined in the previous chapter as described in Definition 5.4. Namely, we will use the knowledge base $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ that correspond to PVCS δ, to MES μ and to IS ι as described in Section 4.4. This means that $\mathcal{K}_{\mathsf{op}}$ contains the knowledge about the value tree (i.e., the influence relations between the perspectives), the monadic evaluations, what attributes measure what perspectives and how various assignments compare from the criterion perspectives.

Arguments And Attacks Between Them

Using the ASPV, we can consider all the arguments that can be constructed from the KBPV $\mathcal{K}$. The accrual mechanism of Section 2.2 will not be demonstrated here, but will be demonstrated in the next chapter. We will now highlight seven interesting arguments that can be constructed from $\mathcal{K}$.

$$A_1 = \frac{s_d <_{c_3} s_a \quad c_3{\downarrow}o_1}{s_a <_{o_1} s_d}\, d_{\downarrow} \qquad A_2 = \frac{s_a <_{c_4} s_d \quad c_4{\downarrow}o_1}{s_d <_{o_1} s_a}\, d_{\downarrow}$$

Note that arguments A_1 and A_2 rebut each other. Because the objective o_1 positively influences the value v_1, A_1 and A_2 can both be used to infer value from o_1 to v_1. We will show this for A_2 in A_3. Similarly, A_3 can then be used to infer value from value v_1 to the student's perspective α, which is done in argument A_4.

$$A_3 = \frac{A_2 \quad o_1{\uparrow}v_1}{s_d <_{v_1} s_a}\, d_{\uparrow} \qquad A_4 = \frac{A_3 \quad v_1{\uparrow}\alpha}{s_d <_{\alpha} s_a}\, d_{\uparrow}$$

Criterion c_1 is measured by attribute x_1, which denotes the number of minutes that the victims are inside the factory. Because s_a's x_1 value is 30 and s_d's x_1 value is 20, it is true that $(s_d <_{c_1} s_a) \in \mathcal{K}_{\mathsf{op}}$. Using this knowledge, an argument can be constructed that propagates value from criterion c_1 to the agent's preferences as follows:

$$A_5 = \frac{s_d <_{c_1} s_a \quad c_1{\downarrow}o_2}{s_a <_{o_2} s_d}\, d_{\downarrow} \qquad A_6 = \frac{A_5 \quad o_2{\uparrow}v_2}{s_a <_{v_2} s_d}\, d_{\uparrow} \qquad A_7 = \frac{A_6 \quad v_2{\uparrow}\alpha}{s_a <_{\alpha} s_d}\, d_{\uparrow}$$

In words, because in s_d the victims are 20 minutes inside instead of s_a's 30 minutes, s_d is better from the perspective of safety for victims and therefore s_d is also better from the perspective of security and thus also from the agent's perspective.

Meta-Argumentation

The meta-language for PV w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ is the first-order language $\mathcal{L}'_{\mathsf{pv}}$ such that for each formula, inference rule and argument in $\mathcal{AS}_{\mathsf{pv}}$ there are constants in $\mathcal{L}'_{\mathsf{pv}}$. Also for each set of formulae, inference rules and arguments in $\mathcal{AS}_{\mathsf{pv}}$ there are constants in $\mathcal{L}'_{\mathsf{pv}}$.

The meta-knowledge base is instantiated with the Perspective Influence Structure ι, which contains the relative importances between perspectives. Consequently, the meta-level ordinary premises knowledge base $\mathcal{K}'_{\mathsf{op}}$ contains $c_2 \lhd_{o_3} c_6$ and $c_1 \lhd_{o_2} c_5$.

Because the amount of chemicals that escapes into the environment is more important for the environment than the amount of smoke that is produced, $c_3 \lhd_{o_1} c_4$ is in $\mathcal{K}'_{\mathsf{op}}$. Recall from Table 5.4 that the (meta-level) defeasible rule $d'_{\uparrow\uparrow}$ defeasibly infers from $p \lhd_r q$ that applying the (object-level) defeasible rule $d_{\downarrow}$ on q to r is stronger than applying it on p to r. Consequently, the following argument can be constructed, which concludes that inferring value from c_3 to o_1 (i.e., $d_{\downarrow}(c_3, o_1, s_a, s_d)$) is less strong than inferring value from c_4 to o_1 (i.e., $d_{\downarrow}(c_4, o_1, s_a, s_d)$).

$$B_1 = \frac{c_3 \lhd_{o_1} c_4}{\{d_{\downarrow}(c_3, o_1, s_a, s_d)\} \prec_{\mathcal{R}} \{d_{\downarrow}(c_4, o_1, s_a, s_d)\}}\, d'_{\downarrow\downarrow}$$

We will now use r_1 to denote $d_{\downarrow}(c_3, o_1, s_a, s_d)$ and r_2 to denote $d_{\downarrow}(c_4, o_1, s_a, s_d)$. Because $\mathsf{lastDef}(A_1) = \{r_1\}$, it is true that $\mathsf{lastDef}(A_1) \neq \emptyset$. Then we can apply the strict rule of logical or introduction to construct the following strict and firm argument.

$$B_2 = \frac{\mathsf{lastDef}(A_1) \neq \emptyset}{\mathsf{lastDef}(A_1) \neq \emptyset \ \vee \ \mathsf{lastDef}(A_2) \neq \emptyset} \ \vee I$$

Using arguments B_1 and B_2 we can now apply the strict rule of logical and introduction to construct the following argument. Note that $\mathsf{lastDef}(A_2) = \{r_2\}$.

$$B_3 = \frac{B_2 \qquad B_1}{(\mathsf{lastDef}(A_1) \neq \emptyset \vee \mathsf{lastDef}(A_2) \neq \emptyset) \quad \wedge \quad \mathsf{lastDef}(A_1) \prec_{\mathcal{R}} \mathsf{lastDef}(A_2)} \ \wedge I$$

Note that B_3 is defeasible and plausible because its subargument B_1 applied a defeasible rule and because the premise $c_3 \lhd_{o_1} c_4$ is not a necessary premise. Now we can use the last-link axiom $\mathtt{lstLnk}'_1$ to infer that A_2 is a stronger argument than A_1 because the last applied defeasible rules of A_2 are stronger than those of A_1:

$$B_4 = \frac{B_3 \quad \mathtt{lstLnk}'_1}{A_1 \prec_{\mathsf{Args}} A_2} \ \mathsf{MP}$$

The argument ordering over meta-level arguments represents their relative strength and must be an admissible ordering. Furthermore, because B_2 is a strict and firm argument and B_1 is a defeasible argument, it is true that B_2 is stronger than B_1. Because B_3 applies a strict rule on B_1 and B_2, argument orderings (see Definition 2.12) enforce that B_3 is not stronger than its subarguments and equally strong as one of its subarguments. Since B_2 is stronger than B_1, argument B_3 must be as strong as B_1 and less strong than B_2. Consequently, the meta-level arguments are ordered as: $B_3 < B_2$ and $B_3 \equiv B_1$.

Using the meta-level argumentation framework, we will now determine the grounded argument ordering (as in Definition 2.41) over object-level arguments. Because of the axioms in a meta-knowledge base, the grounded argument ordering $\leq$ contains:

- reflexive: for all object-level arguments A it is true that $A \leq A$,
- transitive: for all object-level arguments A_1, A_2 and A_3 it is true that if $A_1 \leq A_2$ and $A_2 \leq A_3$, then $A_1 \leq A_3$.

Furthermore, because meta-argument B_4 is a justified argument, $\leq$ also states that $A_1 < A_2$. Given the grounded argument ordering $\leq$ resulting from the meta-level argumentation mechanism, the object-level argumentation theory is the tuple $\mathcal{AT} = \langle \mathcal{AS}_{\mathsf{pv}}, \mathcal{K}, \leq \rangle$. The argumentation framework corresponding to $\mathcal{AT}$ is the tuple $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$ where Args is the set of object-level arguments that can be constructed from $\mathcal{K}$ and Attack the attack relation between them. Thus Args contains arguments A_1 till A_7. Recall that A_1 and A_2 preference-dependent attack each other. Because in the grounded argument it is true that $A_1 < A_2$, the attack of A_1 on A_2 is unsuccessful, whereas A_2's attack on A_1 is successful. This means that $(A_2, A_1) \in \mathsf{Attack}$.

5.4 Chapter Summary

In complex situations it is often not clear to a decision maker what outcome he values the most. To make a decision that can be justified it is necessary to explain why a certain out-

come was valued in a certain way. Research question 1a asked what concepts are required to reason about what a decision maker values. Chapter 3 proposed a conceptual framework for value in which a number of value concepts and argument schemes to reason with value were proposed. Research question 1b asked how argumentation can be used for reasoning about what a decision maker values. To answer this question it is necessary to formalize the conceptual framework proposed in Chapter 3 and this was done in the previous chapter. To argue about what to value, this chapter proposed an argumentation framework for this purpose that is based on this conceptual framework and its formalization.

In the conceptual framework, what a decision maker values is decomposed into a value tree. For example, the decision maker's perspective may be decomposed into a number of abstract values, which are then further decomposed into a number of criteria that influence what is better from the perspective of a certain abstract value. Several argument schemes can then be used that use the decision maker's value tree to justify that one outcome is valued more than another. Because not all perspectives that influence some perspective p may be equally important to perspective p, these arguments differ in strength. To model this it is required to distinguish between object-level argumentation and meta-level argumentation. Meta-level argumentation is used to justify that one argument is stronger than another. Section 5.1 proposed the object-level argumentation system and Section 5.2 the meta-level argumentation system. In these two argumentation systems, the concepts and argument schemes of Chapter 3 were formalized. Section 5.3 then illustrated this argumentation framework on the running example of the introduction.

6 Practical Reasoning

Practical reasoning is reasoning about what to do and is often contrasted with theoretical reasoning, i.e., reasoning about what is true. In the previous chapters we have proposed the Perspective-based Value Model (PVM), a formal model to represent value from different perspectives, and an argumentation logic to reason about what an agent cares about. What an agent cares about is decomposed into the general areas of concern, i.e., the values that the agent pursues. General areas are further decomposed into intermediate objectives for which specific evaluation criteria are used to measure their performance. An agent should perform the alternative that results in the outcome that is valued the most from his perspective, i.e., the outcome that he prefers most. Using the PVM to reason about this is thus a way to do practical reasoning. However, because there may be many alternatives and each alternative may result in a number of outcomes, determining an agent's preferences over these outcomes can require a large amount of computation. We will illustrate this with the following example.

Example 6.1 (Using PVM for Making Decisions) An agent's value tree consists of several layers: first a layer of basic values, then a layer of objectives, and finally a layer of evaluation criteria. As explained in Section 2.5, there are ten basic values representing the general areas of concern of an agent. For each basic value, there may be a number of intermediate objectives that further the pursuit of that basic value. Finally, for each intermediate objective there may be a number of specific evaluation criteria that measure the performance of that objective. Suppose that on average 3 intermediate objectives are used per basic value and that on average each objective's performance is measured by 2 specific evaluation criteria. In this case, there are $10 \times 3 \times 2 = 60$ specific evaluation criteria that the agent uses to determine what assignment he values the most.

How assignments compare on these 60 evaluation criteria is used to argue about how assignments compare from the perspectives of the 30 objectives the agent cares about. It is likely that one assignment is better than another on some criteria, but worse on other criteria. In such cases, the arguments will conflict that conclude how assignments compare on the objectives that are measured by these criteria. If a conflict can be resolved, then there is no problem, but it is possible that conflicts cannot be resolved. Similarly, how assignments compare in value from the perspectives of these 30 objectives is used to argue about how these assignments compare from the perspectives of the ten basic values. Finally, how assignments compare from the perspectives of the ten basic values is used to determine how the value of

the assignments compares from the agent's perspective.

Suppose there are n possible outcomes of the alternatives an agent can perform. Ideally, the agent should choose the alternative with the most preferred outcome. This means that the agent needs to do at least $n - 1$ comparisons. Comparing the value of two assignments from an agent's perspective requires comparing them on the 60 specific evaluation criteria.

As illustrated in the example, using the PVM for practical reasoning will require a large amount of computation in normal decision situations. Simon (1957) argues that the amount of resources that both human and artificial agents have is bounded. Consequently, they cannot compute an arbitrarily large number of computations in a constant time. In the literature, approaches have been proposed for resource-bounded practical reasoning. For example, Raz (1978) and Pollock (1998) propose to do practical reasoning in two steps: first an agent should identify what goals he wants to achieve, next, he should determine the means that achieve these ends and select the best means. The first step of identifying what goals should be achieved is often called *deliberation*[1] (Wooldridge, 2000). The second step is often called *means-end reasoning*, which involves finding suitable plans of actions, intermediary goals, and sub-plans and also weighing the different means in order to select the best one.

To check easily whether an outcome achieves a goal, goals need to be specific. The specific evaluation criteria in an agent's value tree describe what he cares about in a specific way. In this chapter we therefore propose that an agent's goals are expressed in terms of specific evaluation criteria. Instead of comparing each outcome on each criterion, we will therefore propose to generate a goal with respect to each criterion and then check what goals are achieved by each outcome. Because goals are specific, it is easy to check whether an outcome achieves a goal.

This chapter extends the argumentation framework for perspective-based value proposed in the previous chapter such that it is possible to do practical reasoning taking into account resource-bounds. First, Section 6.1 introduces achievement and avoidance goals and describes informally how an agent can justify them using his PVM by proposing a number of argument schemes. Then, Section 6.2 describes how alternatives can be found that achieve these goals at least partly and proposes several argument schemes that an agent can use to justify choosing a certain alternative. Section 6.3 then formalizes the proposed argument schemes by extending the object-level and meta-level argumentation systems for perspective-based value. We will then demonstrate the extended formalism on the running example in Section 6.4 and compare our approach to existing approaches for using argumentation in practical reasoning in Section 6.5. The chapter is concluded with a summary.

6.1 Deliberation

In practical reasoning, identifying what goals an agent should try to achieve is called deliberation (Wooldridge, 2000). In the perspective-based value model, the outcomes of performing an alternative are expressed using assignments. Therefore, we will also represent goals in terms of assignments. In this section we will distinguish two kinds of goals and propose several argument schemes that justify an agent having a certain goal using the agent's value

[1] Note 'deliberation' is used differently here than in Walton and Krabbe (1995a), where deliberation is a kind of dialogue with the collective goal to decide what course of action is the best.

tree and his monadic evaluations. Because a value tree may be large and there may be limited time and resources to make a decision, an agent may have to focus on part of his value tree. Subsection 6.1.1 introduces the notion of a *decision context* to focus on part of an agent's value tree. We will then discuss two kinds of goals: achievement goals and avoidance goals. Subsection 6.1.2 proposes how achievement goals can be justified and Section 6.1.3 proposes how avoidance goals can be justified. The next section of the chapter, Section 6.2, describes how an agent can do means-end reasoning using the goals and rejections that he has justified. Finally, Section 6.3 formalizes the argument schemes that are proposed in this Section.

6.1.1 Decision Context

The context in which a decision is made determines what perspectives matter to an agent. For example, when deciding what movie to watch, an agent may care about the perspective of fun, but when deciding how to help victims trapped in a burning house, an agent does not care about the perspective of fun and does care about the perspective 'health of victims'. In Keeney (1992), this is called the *decision context*.

The time that is available to determine what decision is the best has a significant effect on what perspectives an agent can consider in that decision context. If little time is available to make a decision, e.g., because the more time is taken, the more likely a victim gets hurt, then the decision-maker only has time to focus on the most important perspective. In contrast, if much time is available, e.g., if it has to be decided what scientific theory will be used, then the various alternatives can be compared from all the different perspectives an agent cares about.

Determining how much time is available for a decision and thus how many perspectives can be considered is a form of meta-reasoning about what decision-making method the agent should use. This is a complex, but important issue, which is outside the scope of this thesis. Instead we will assume there is information concerning what perspectives in the agent's value tree are important in the current decision context. To denote this, we say that 'perspective p is important for agent α in the current decision context'.

6.1.2 Justification of Achievement Goals

The notion of a *goal* is a motivational construct that is often used in argumentation to justify making a certain decision, e.g., Amgoud and Prade (2009); Atkinson et al. (2006); Bench-Capon et al. (2009); Walton (1996). A goal is typically represented as a proposition or literal that is either true or false. A goal can thus be seen as a partition of possible states of the world into those that achieve the goal (i.e., the proposition is true) and those that do not. For example, in Wellman and Doyle (1991), a goal is a proposition representing a subset of the space of all outcomes. If an agent has goal G, then this means that the states that satisfy G are desirable to the agent in some way.

In the PVM, what an agent values is represented using perspectives. Instead of using a literal that is either true or false to describe what an agent values, perspectives order assignments by their amount of value from that perspective, which is a clear representation of the degrees to which agents desire assignments. For example, the perspective 'profit' represents that 1 euro profit is valued more (or preferred) than 0 euro, that 2 euro profit is valued more than 1, and so on. Representing that higher amounts of profit are more valuable with literals

is a cumbersome task. However, for practical reasoning it is convenient to have these literals that are either true or false to describe what an agent wants. For this, we will use goals.

To take into account the resource-bounds of agents, Simon (1955) proposes to simplify value functions by mapping outcomes to 1, 0 or -1 denoting a satisfactory, indifferent and unsatisfactory level of value respectively. We will use a similar approach that splits the ordering from a perspective into three parts. Following Amgoud and Prade (2009), we distinguish between *achievement goals* and *avoidance goals* (see Subsection 6.1.3). In our framework, an achievement goal states that a certain set of assignments has a satisfactory level of value for the agent from a certain perspective. For example, an agent may have the achievement goal from the perspective of 'profit' to achieve a profit of at least a thousand euro, because this amount has a satisfactory level for the agent, even though he would be more satisfied with an even higher profit. Note that this does not mean that all assignments in an achievement goal have an equal amount of value from its perspective. In contrast, an avoidance goal states that a certain set of assignments has such a low level of value from a certain perspective that the agent should try to avoid achieving those assignments. For example, an agent may have the avoidance goal from the perspective of 'profit' to achieve a profit of a hundred euro or less because this would be a really bad profit. Achievement and avoidance goals in our framework express that their associated assignments have a certain level of value. If an agent has a certain achievement or avoidance goal, it does not mean that the agent has a certain intention.

We propose that an agent can justify having a certain achievement goal from perspective p using his evaluation of what assignments are 'good' from p. Because the agent evaluates those assignments as good, the agent expresses a certain level of satisfaction. Furthermore, in the decision situations in which we are interested, there is a limited amount of time. Therefore, an achievement goal w.r.t. perspective p is only justified if p is a perspective that is important in the current decision context. This intuition is captured informally in the following argument scheme.

Argument Scheme 11: Achievement Goal

In the current decision context, perspective p is important for agent α,
agent α evaluates all assignments in G as good from perspective p,

---

therefore, presumably, agent α has the achievement goal from perspective p to achieve an assignment in G.

Note that using this argument scheme, an agent can justify having different achievement goals from different perspectives that conflict. For example, an agent may have the achievement goal to live close to the city centre from the perspective of fun, but could also have the goal to not live in the centre from the perspective of quietness. In our framework, it is not a problem that an agent has achievement goals that cannot be achieved simultaneously.

However, it is a problem if an agent has multiple achievement goals from the same perspective. To prevent this, this argument scheme should not be applied on a set assignments G if it already has been applied on a superset of G. For this purpose the critical question *Does α have an achievement goal from p to achieve an assignment in a superset of G?* is associated to Argument Scheme 11. If the critical question is answered positively, then the scheme cannot be applied. Because we need a critical question to ensure this problem does not occur, Argument Scheme 11 needs to be defeasible.

Further note that this argument scheme is liberal in the sense that the set of assignments

that is adopted as a goal does not need to be the set of all assignments that are good from a perspective. If a maximal goal set is mandatory, then agents need to evaluate every assignment. If there are many assignments, this takes up many resources. For that reason, even small sets of assignments can be adopted as achievement goals. In this way, an agent can start with a small set of assignments as an achievement goal, try to find an alternative that achieves that goal and if he finds one he can stop looking. If he cannot find an alternative, then he can try to make his goal broader.

6.1.3 Justification of Avoidance Goals

Recent evidence suggests that humans have different motivational systems for positive and negative stimuli (Cacioppo et al., 1997). Inspired by this research, Amgoud and Prade (2009) model agents with *goals* and so-called *rejections*, which we call *achievement goals* and *avoidance goals* respectively. Whereas an agent's achievement goal explicates something he wants to achieve (and therefore can be seen as a positive stimulus), an agent's avoidance goal explicates something he wants to avoid (which can be seen as a negative stimulus). We say that *assignment s achieves agent α's avoidance goal w.r.t. perspective p* if and only if agent α has the avoidance goal w.r.t. perspective p to avoid assignments in G and s does not achieve G. In other words, if an assignment s achieves an avoidance goal, then s avoids what the agent wanted to avoid.

Although an avoidance goal is logically similar to a goal, it is important to distinguish between achievement and avoidance goals because evidence suggests that humans treat them differently. The use of either achievement or avoidance goals may also have a significant effect on how a decision can be framed. By distinguishing between achievement and avoidance goals, an agent can have three attitudes towards an outcome w.r.t. a perspective: negative (i.e., the agent's avoidance goal w.r.t. p is not achieved), neutral (i.e., the agent's avoidance goal w.r.t. p is achieved but his achievement goal is not achieved), or positive (i.e., both the agent's avoidance and achievement goal w.r.t. p are achieved).

Similar to the argument scheme that justifies an agent having an achievement goal, we propose the following argument scheme that justifies an agent having a certain avoidance goal, i.e., wanting to avoid certain assignments. Because an agent should only have maximally one avoidance goal per perspective, this argument scheme should be defeasible. Otherwise it is possible that multiple avoidance goals are inferred for one perspective.

Argument Scheme 12: Avoidance Goals

In the current decision context, perspective p is relevant for agent α,
agent α evaluates all assignments in G as bad from perspective p,

---

therefore, presumably, agent α has the avoidance goal from perspective p to avoid assignments in G.

For the same reason as with Argument Scheme 11, we want to prevent that agents have multiple avoidance goals from the same perspective. For this purpose the critical question *Does α have an avoidance goal from p to avoid an assignment in a superset of G?* is associated to Argument Scheme 12. If the critical question is answered positively, then the scheme cannot be applied. Assume that Argument Scheme 11 is used to justify avoiding a movie of the genre comedy (avoidance goal 1), but also to justify avoiding a movie of either the

genre comedy or drama (avoidance goal 2). Because avoidance goal 2 is more general than avoidance goal 1, avoidance goal 1 should not be used.

What is good from one perspective may be bad from another perspective. It is therefore possible that an agent wants to avoid achieving an assignment s because s is bad from one perspective, but has the goal to achieve s because s is good from another perspective. This is a desirable and unproblematic feature. Because achievement and avoidance goals w.r.t. different perspectives may differ in importance to an agent, he may decide that it is more desirable to achieve an achievement goal w.r.t. perspective p even though he does not achieve an avoidance goal w.r.t. some other perspective.

Note that an assignment cannot be evaluated as both good and bad from a perspective (see Subsection 3.4.1 for more details). An argument that justifies an agent having the goal to achieve assignment s from perspective p therefore attacks an argument that uses this scheme to justify an agent wanting to avoid s.

6.2 Means-End Reasoning

The first step in practical reasoning is deliberation, in which an agent identifies what achievement and avoidance goals he adopts. Given the goals of an agent, the second step in practical reasoning is called means-end reasoning. In this step, the agent tries to find alternatives that achieve at least some of his achievement goals, compares the possible alternatives and selects the alternative[2] that he likes most. Note that Wooldridge (2000, p.84) considers the process of selecting an alternative as part of deliberation, whereas we see this as part of means-end reasoning. In either way, the same steps need to be taken, so whether it is called deliberation or means-end reasoning is not an important difference.

First, Subsection 6.2.1 discusses how we will represent that an alternative may result in an outcome. Then Subsection 6.2.2 proposes several argument schemes that justify an agent's decision to perform a certain alternative based on what goals and rejections that alternative achieves. Finally, Subsection 6.2.3 proposes argument schemes to justify the relative strengths of arguments in favour and against decisions. The argument schemes in this section will then be formalized in Section 6.3.

6.2.1 Outcomes of Alternatives

The decision of an agent to perform an alternative results in an outcome. We will use assignments to express in what outcome an agent executing an alternative may result. Let Alt be the set of alternatives that can be executed by agents, $\mathcal{A}$ be the set of attributes used to describe the outcomes of alternatives, and $\mathcal{S}$ be the set of all assignments on attributes in $\mathcal{A}$.

Definition 6.1 (Outcome Mapping) *An* outcome mapping *is a function* $\mathsf{outcome} : \mathsf{Alt} \rightarrow 2^{\mathcal{S}}$ *such that for all* $a \in \mathsf{Alt}$*: if* $s \in \mathsf{outcome}(a)$ *and* t *satisfies* s*, then* $t \in \mathsf{outcome}(a)$.

Recall from Section 4.1 that assignment t satisfies assignment s iff t assigns the same attribute values as s to all attributes that s assigns an attribute value to. To illustrate the constraint on outcome mappings, consider the following example. Suppose that when deciding

[2]Recall from Section 3.2 that agents can only select one alternative.

what car to buy, choosing alternative a may result in assignment s, i.e., $s \in \mathsf{outcome}(a)$, and that s denotes buying a Ford of a certain price, but that s does not assign an attribute value on the attribute 'annual maintenance costs'. In other words, when buying the Ford it is not known how much the annual maintenance costs will be. Because of the constraint on outcome mappings, this means that every annual maintenance cost is possible. However, if it is known that certain maintenance costs are impossible, e.g., it will certainly not be more than a thousand euro, then assignment s should not be in $\mathsf{outcome}(a)$, but $\mathsf{outcome}(a)$ should contain all the assignments that satisfy assignment s and assign a possible attribute value on the attribute of maintenance costs.

An alternative a is called *deterministic* if and only it may result in exactly one assignment, i.e., $|\mathsf{outcome}(a)| = 1$. If alternative a does not result in any assignments, i.e., $\mathsf{outcome}(a) = \emptyset$, then we say that performing a is *impossible*. Otherwise, alternative a is called *non-deterministic*. This means that if alternative a is deterministic, then a results in a single assignment on all attributes.

6.2.2 Justification of Decisions

The achievement and avoidance goals that an alternative achieves provide justification for an agent to decide to perform that alternative. This subsection proposes several argument schemes that informally describe several intuitions with respect to achievement and avoidance goals and deciding what alternative should be performed. These argument schemes will be used to provide a single reason for choosing or not choosing an alternative. Because it is possible that there are reasons for choosing an alternative, but also reasons not to choose that alternative, all the argument schemes in this section must be defeasible.

The underlying idea is that achievement goals can only provide positive stimuli and avoidance goals can only provide negative stimuli. This means that achieving an achievement goal provides a reason to perform a certain alternative (i.e., a positive stimulus), but not achieving an achievement goal does not provide a reason not to perform an alternative. Similarly, achieving an avoidance goal does not provide a reason to perform an alternative, but not achieving an avoidance goal does provide a reason not to perform an alternative (i.e., a negative stimulus).

If an alternative certainly achieves an achievement goal of an agent, then that is a reason for the agent to decide to perform that alternative. Namely, the agent wanting to achieve a certain goal and some alternatives doing this is a reason to perform that alternative. The following argument scheme describes this intuition informally.

Argument Scheme 13: Surely Achieved Achievement Goal

Alternative a certainly achieves agent α's achievement goal w.r.t. perspective p,

---

therefore, presumably, agent α should decide to perform alternative a.

In contrast, if an alternative certainly does not achieve an avoidance goal of an agent, i.e., an alternative certainly achieves what the agent wanted to avoid, then that is a reason for the agent to decide not to perform that alternative. This intuition is captured in the following argument scheme.

Argument Scheme 14: Surely Achieved Avoidance Goal

Alternative a certainly does not achieve agent α's avoidance goal w.r.t. perspective p,

therefore, presumably, agent α should decide not to perform alternative a.

Note that because we assume that alternatives are exclusionary (see Section 3.2), i.e., an agent can only decide to perform exactly one alternative, an argument concluding 'perform alternative a' and an argument concluding 'perform alternative b' attack each other.

If an alternative is non-deterministic, then it may result in a number of different outcomes. If there is an outcome of an alternative that achieves a goal of an agent, but not all outcomes achieve this goal, then we say that the alternative possibly achieves that goal. If an alternative possibly achieves a goal of an agent, then that is a reason for the agent to decide to perform that alternative. The following argument scheme informally describes this intuition.

Argument Scheme 15: Possibly Achieved Achievement Goal

Alternative a possibly achieves agent α's achievement goal w.r.t. perspective p,

therefore, presumably, agent α should decide to perform alternative a.

In contrast, if an alternative possibly does not achieve an avoidance goal of an agent, then that is a reason for the agent to decide not to perform that alternative. This intuition is captured in the following argument scheme.

Argument Scheme 16: Possibly Achieved Avoidance Goal

Alternative a possibly does not achieve agent α's avoidance goal w.r.t. perspective p,

therefore, presumably, agent α should decide not to perform alternative a.

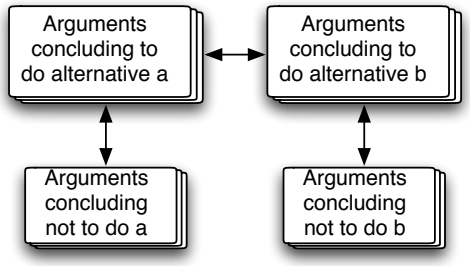
Note that applying Argument Schemes 15 and 16 provides weaker reasons to perform an alternative than applying Argument Scheme 13 and 14. Moreover, because we do not distinguish between different levels of how likely performing an alternative results in an outcome, this argument scheme should be used with care as it may create very weak reasons to choose alternatives. For example, when there is much uncertainty, performing an outcome may result in a large number of different outcomes. It is therefore possible that such an alternative possibly achieves a goal even though it is very unlikely. In this case, these argument schemes can be used to justify performing that alternative. Ideally, our framework should be extended such that the likelihood of an alternative resulting in an outcome is taken into account.

All the proposed argument schemes conclude that an agent should decide to perform or decide not to perform a certain alternative. Applying these schemes will thus result in a number of arguments, some of which have the same conclusion, and some of which attack each other. To determine what decision should be taken, the agent should reason about the relative strengths of arguments and accrue them.

In our framework, alternatives are exclusionary, which means that an agent can only choose to perform a single alternative. Consequently, an argument concluding 'perform alternative a' conflicts with an argument concluding 'perform alternative b'. Argument Schemes 13 and 15 both conclude that some agent should choose a particular alternative. Two arguments that use these schemes to conclude that an agent should choose different alternatives thus conflict. In contrast, argument Schemes 14 and 16 conclude that an agent should not choose a certain alternative. An argument concluding 'perform alternative a' and an argument concluding 'do not perform alternative a' also conflict.

Arguments concluding that an alternative should be performed can thus be attacked by two kinds of arguments: arguments concluding that another alternative should be performed

Figure 6.1: *Attacks between arguments about what decision to make*



and arguments concluding that the alternative should not be performed. Figure 6.1 visualizes the attacks between possible arguments in favour and against two alternatives a and b.

Because of this special structure between arguments concerning what alternative to perform, several approaches have been proposed to compare what decision is preferred. Subsection 6.5.2 discusses such approaches in detail, but we will give an example here to give an idea. Namely, Amgoud and Prade (2009) propose several types of so-called 'decision principles' to compare decisions. Unipolar decision principles compare decisions by only looking at the arguments in favour of or against decisions, e.g., alternative a is better than alternative b iff a achieves more achievement goals than b. Bipolar decision principles use both arguments in favour and against decisions to compare what decision is better.

In our framework, the various arguments in favour or against a decision can be accrued into an accrual argument. This results in one accrual argument concluding 'perform alternative a', one accrual argument for 'do not perform alternative a', one accrual argument for 'perform alternative b' and one accrual argument for 'do not perform alternative b'. To determine what arguments are acceptable we need to establish the relative strengths of these accrual arguments. However, this is a complicated matter, especially when taking into account that the uncertainty that an alternative achieves a goal. In this thesis we will not give a full account on how to compare the strength of accrual arguments, but we will discuss the relative strength of applying the argument schemes that we have proposed. This relative strength can be used to determine the strength of accrual arguments.

6.2.3 Strength of Arguments Schemes

The argument schemes proposed in the previous subsection can be used to construct arguments pro and con decisions. For each decision, this may result in several arguments pro and con. However, not every application of these argument schemes creates as much conclusive force.

If one alternative a surely achieves an achievement goal w.r.t. perspective p and another alternative b only possibly achieves that goal, then there is a stronger reason w.r.t. p to perform a than to perform b. Consequently, applying Argument Scheme 13 on perspective p has more conclusive force than applying Argument Scheme 15 on perspective p. Similarly, if alternative a certainly achieves what the agent wants to avoid from perspective p and alternative b may or may not achieve what the agent wants to avoid from p, then applying Argument Scheme 14 has more conclusive force than applying Argument Scheme 16.

If a perspective p is more important for agent α than perspective q, then that an alternative a achieves a goal w.r.t. p creates a stronger reason to perform a than when a achieves a goal w.r.t. q. For example, if the perspective of safety is more important to agent α than the perspective of fun, then achieving a goal w.r.t. safety creates a stronger reason to perform an alternative than achieving a goal w.r.t. fun. The following argument scheme captures the intuition that if an agent finds perspective q more important than perspective p, then applying Argument Scheme 13 on a goal w.r.t. q creates more conclusive force than applying it on a goal w.r.t. p. Because we want to allow that other argument schemes can be introduced to reason about the relative strength of applying these argument schemes, the following argument scheme is defeasible.

Argument Scheme 17: Goals and Perspective Importance

Perspective p is more important for agent α than perspective q,

---

therefore, presumably, applying Argument Schemes 13-16 on a goal w.r.t. p to justify a decision for α creates a stronger reason than applying them on a goal w.r.t. q.

Because this argument scheme is concerned with the relative strengths of applications of Argument Schemes 13-16, it can be seen as a meta-level argument scheme. In the next section, we will formalize Argument Scheme 17 in a meta-level argumentation system.

6.3 Formalization

In this chapter we have proposed how achievement and avoidance goals and decisions can be justified using argumentation and the perspective-based value model. In this section we will formalize these ideas by extending the argumentation mechanism proposed in Chapter 5. Definition 5.3 defined Argumentation Systems for Perspective-based Value (ASPV), which are used to reason about value from different perspectives. In addition, Definition 5.8 defined Meta-level Argumentation Systems for Perspective-based Value (MASPV) to reason on a meta-level about the strength of (object-level) arguments in the ASPV. This section extends the ASPV and MASPV to make them suitable for practical reasoning.

Subsection 6.3.1 proposes an argumentation system that extends the ASPV with predicates and defeasible inference rules to justify an agent's achievement and avoidance goals using his value tree and rules to justify decisions using the goals they achieve. Subsection 6.3.2 proposes a meta-level argumentation system that extends the MASPV to reason about the strengths of arguments.

6.3.1 Perspective-Based Argumentation for Practical Reasoning

To formalize the argument schemes for practical reasoning described in this chapter, the logical language is extended with several constants and predicates. Consequently, the contrariness function also needs to be extended. An additional set of defeasible inference rules is then used to formalize the new argument schemes in this chapter. We will start with the language for perspective-based practical reasoning, which extends the language for perspective-based value (see Definition 5.1) with several constants and predicates.

Definition 6.2 (Language for Perspective-Based Practical Reasoning) *Let* Agents *be a set of agents,* Alt *a set of alternatives that the agents in* Agents *can execute and δ a PVCS. A* Language for Perspective-based Practical Reasoning *is a language for perspective-based value for δ extended with*

- *constants:*
  - *for each alternative $a \in$ Alt there is a constant $\underline{a}$*
  - *for each agent $\alpha \in$ Agents there is a constant $\underline{\alpha}$*
- *predicates:*
  - *binary predicates* results, cntxt, do
  - *ternary predicates* goal, avoid

Because the Language for Perspective-based Practical Reasoning (LPPR) extends the Language for Perspective-based Value (LPV), all the constants, functions, and predicates that are in the LPV are also in the LPPR. See Definition 5.1 in Subsection 5.1.1 for the details. Similar to the previous languages that were proposed, we will write x instead of $\underline{x}$ for constants. The binary predicate $\mathsf{results}(a, s)$ denotes that assignment s is a possible outcome of alternative a. The predicate $\mathsf{cntxt}(\alpha, p)$ denotes that perspective p is important for agent α in the current decision context. The predicate $\mathsf{do}(\alpha, a)$ denotes that agent α should decide to perform alternative a. The ternary predicates $\mathsf{goal}(\alpha, p, G)$ and $\mathsf{avoid}(\alpha, p, G)$ denote that agent α has the achievement / avoidance goal from perspective p to achieve the set of assignments G respectively.

Example 6.2 (Movies) Consider assignments s_1, s_2 and s_3, where s_1 denotes 'genre is comedy', s_2 denotes 'genre is drama' and s_3 denotes 'genre is horror'. Furthermore, assignment t_1 denotes 'length is 100 minutes' and assignment t_2 denotes 'length is 120 minutes'. Finally, u_{ij} is the assignment that combines s_i with t_j. For example, u_{21} denotes 'genre is drama and length is 100 minutes'.

An alternative that can be chosen is a_1, which denotes watching the movie 'Eternal Sunshine of the Spotless Mind'. Because this movie is a drama and takes 100 minutes, alternative a_1 results in assignment u_{21}. This can be represented with $\mathsf{results}(a_1, u_{21})$ being true.

Agent α wants to have a relaxing evening and therefore the perspective of fun (denoted with fun) is important in the current decision context. This can be represented by the statement $\mathsf{cntxt}(\alpha, \mathsf{fun})$. What movie is fun is determined by the genre of the movie. Agent α has the achievement goal from the perspective of fun to either watch a comedy or a drama movie. However, α has the avoidance goal from the perspective of fun to watch a horror movie. This can be represented by the statements $\mathsf{goal}(\alpha, \mathsf{fun}, \{s_1, s_2\})$ and $\mathsf{avoid}(\alpha, \mathsf{fun}, \{s_3\})$.

In addition to the abbreviations in Table 5.1 that are used in the LPV, we will also use the abbreviations in Table 6.1. Recall from the previous abbreviations that, for example, $\exists_{t \in G}[\mathsf{satisfies}(t, s)]$ is an abbreviation of $\exists_t[\mathsf{in}(t, G) \wedge \mathsf{satisfies}(t, s)]$.

We will now define the contrariness function for perspective-based practical reasoning. Because the LPPR extends the LPV, we will extend the contrariness function for perspective-based value by copying all contraries and adding new ones. Since alternatives are exclusionary, agents can only choose one alternative. Therefore, $\mathsf{do}(\alpha, a)$ and $\mathsf{do}(\alpha, b)$ conflict with each other if alternative a and b are different. Lastly, we want to ensure that it conflicts to

Table 6.1: *Abbreviations in the Language for Perspective-Based Practical Reasoning*

Abbreviation	Of
$\mathsf{achieves}(s, G)$	$\exists_{t \in G}[\mathsf{satisfies}(t, s)]$
$\mathsf{alwAch}(a, G)$	$\forall_{s \in \mathcal{S}}[\mathsf{results}(a, s) \supset \mathsf{achieves}(s, G)]$
$\mathsf{mayAch}(a, G)$	$\neg\mathsf{alwAch}(a, G) \wedge \exists_{s \in \mathcal{S}}[\mathsf{results}(a, s) \wedge \mathsf{achieves}(s, G)]$

have an assignment in both an achievement and avoidance goal from the same perspective. Consequently, if two sets of assignments G_1 and G_2 share an assignment (i.e., $G_1 \cap G_2 \neq \emptyset$), then having an achievement goal of G_1 conflicts with having an avoidance goal of G_2 from the same perspective.

Definition 6.3 (Contrariness Function for LPPR) *Let $\mathcal{L}_{\mathsf{ppr}}$ be a language for perspective-based practical reasoning w.r.t. $\mathcal{L}_{\mathsf{pv}}$ and let $\mathsf{cf}_{\mathsf{pv}}$ be a contrariness function for $\mathcal{L}_{\mathsf{pv}}$. A* contrariness function for $\mathcal{L}_{\mathsf{ppr}}$ *is a contrariness function $\mathsf{cf}_{\mathsf{ppr}} : \mathcal{L}_{\mathsf{ppr}} \rightarrow 2^{\mathcal{L}_{\mathsf{ppr}}}$ such that*

- *if $\phi \in \mathsf{cf}_{\mathsf{pv}}(\psi)$, then $\phi \in \mathsf{cf}_{\mathsf{ppr}}(\psi)$,*
- *for all agents α: if $a \neq b$, then $\mathsf{do}(\alpha, a) \in \mathsf{cf}_{\mathsf{ppr}}(\mathsf{do}(\alpha, b))$,*
- *for all agents α and perspectives p: if G_1 and G_2 have a non-empty intersection, then $\mathsf{goal}(\alpha, p, G_1) \in \mathsf{cf}_{\mathsf{ppr}}(\mathsf{avoid}(\alpha, p, G_2))$ and $\mathsf{avoid}(\alpha, p, G_2) \in \mathsf{cf}_{\mathsf{ppr}}(\mathsf{goal}(\alpha, p, G_1))$*

Note that $\mathsf{goal}(\alpha, p, G_1)$ and $\mathsf{goal}(\alpha, p, G_2)$ never conflict, even when $G_1 \cap G_2 = \emptyset$. For example, the achievement goal to watch a movie with the genre 'comedy' does not conflict with the achievement goal to watch a movie with the genre 'drama'. Both goals express that their associated assignments have a satisfactory level of value from perspective p to the agent. They do not express that their associated assignments are the only assignments with a satisfactory level of value. For that reason, they do not conflict. However, to ensure that an agent maximally has one achievement goal for a perspective, we will introduce an undercutter such that an agent can only justify goals with maximal sets of assignment.

Further it is important to recall that the contrariness function for PV does not have any contraries and note that there are no new contraries introduced in the contrariness function for LPPR.

Defeasible Inference Rules

Section 6.1 proposed a number of argument schemes to justify an agent's achievement and avoidance goals based using the agent's value tree. Section 6.2 then proposed argument schemes to justify decisions. Table 6.2 formalizes the proposed argument schemes to do practical reasoning. The defeasible inference schemas[3] d_{goal} and d_{avoid} formalize Argument Scheme 11 and Argument Scheme 12 respectively. Argument Scheme 11 says that if an agent cares about perspective p and evaluates the set of assignments G as 'good', then his achievement goal of G from p is justified. However, if the agent already has an achievement goal G' w.r.t. p and G' is more general, i.e., $G \subset G'$, then this argument scheme cannot be used. The critical question associated to this argument scheme questions this. In a similar fashion,

[3]Recall from Notation 5.1 the notation that we use for defeasible inference schemas and rules.

Argument Scheme 12 can be used to justify that an agent has an avoidance goal and a similar critical question is associated. In this way, an agent can only justify one achievement goal and one avoidance goal per perspective that he cares about. The defeasible inference schemas d_{gUC} and d_{rUC} formalize the critical questions for justifying achievement and avoidance goals respectively. They conclude that the defeasible rules d_{goal} and d_{avoid} respectively cannot be applied in this specific case. Finally, the defeasible inference schemas d_{alwG}, d_{mayG}, d_{alwR} and d_{mayR} formalize Argument Scheme 13, 15, 14, and 16 respectively.

Table 6.2: *Defeasible Inference Schemas for Practical Reasoning*

Goals	$d_{\mathsf{goal}}(\alpha, p, G) : \mathsf{infl}(p, \alpha), \mathsf{cntxt}(\alpha, p), \forall_{s \in G}[\mathsf{good}(\alpha, s, p)] \Rightarrow \mathsf{goal}(\alpha, p, G)$ $d_{\mathsf{avoid}}(\alpha, p, R) : \mathsf{infl}(p, \alpha), \mathsf{cntxt}(\alpha, p), \forall_{s \in R}[\mathsf{bad}(\alpha, s, p)] \Rightarrow \mathsf{avoid}(\alpha, p, R)$ $d_{\mathsf{gUC}}(\alpha, p, G, G') : G \subset G', \mathsf{goal}(\alpha, p, G') \Rightarrow \neg\mathsf{appl}(d_{\mathsf{goal}}(\alpha, p, G))$ $d_{\mathsf{rUC}}(\alpha, p, G, G') : G \subset G', \mathsf{avoid}(\alpha, p, G') \Rightarrow \neg\mathsf{appl}(d_{\mathsf{avoid}}(\alpha, p, G))$
Decisions	$d_{alwG}(\alpha, a, p, G) : \mathsf{goal}(\alpha, p, G), \mathsf{alwAch}(a, G) \Rightarrow \mathsf{do}(\alpha, a)$ $d_{mayG}(\alpha, a, p, G) : \mathsf{goal}(\alpha, p, G), \mathsf{mayAch}(a, G) \Rightarrow \mathsf{do}(\alpha, a)$ $d_{alwR}(\alpha, a, p, R) : \mathsf{avoid}(\alpha, p, R), \mathsf{alwAch}(a, R) \Rightarrow \neg\mathsf{do}(\alpha, a)$ $d_{mayR}(\alpha, a, p, R) : \mathsf{avoid}(\alpha, p, R), \mathsf{mayAch}(a, R) \Rightarrow \neg\mathsf{do}(\alpha, a)$

Now that we have defined a language, contrariness function and a set of defeasible inference rules for PPR, we can define argumentation systems for PPR.

Definition 6.4 (Argumentation System for PPR) *An argumentation system* $\langle \mathcal{L}_{\mathsf{ppr}}, \mathcal{SR} \cup \mathcal{DR}_{\mathsf{ppr}}, \mathsf{cf}_{\mathsf{ppr}} \rangle$ *is called an* Argumentation System for Perspective-based Practical Reasoning *(ASPPR) if*

- $\mathcal{L}_{\mathsf{ppr}}$ *is a language for perspective-based practical reasoning,*
- $\mathcal{SR}$ *is the set of all valid first-order inferences,*
- $\mathcal{DR}_{\mathsf{ppr}}$ *is a set of defeasible inference rules containing the rules in Table 5.2 and Table 6.2, and*
- $\mathsf{cf}_{\mathsf{ppr}}$ *is a contrariness function for PPR.*

Because argumentation systems for PPR extend argumentation systems for PV, the axioms in Table 5.3 are necessary in knowledge bases for PPR. However, no extra axioms are necessary and therefore the knowledge base for PV can be used in argumentation theories for PV.

Definition 6.5 (Argumentation Theory for PPR) *An argumentation theory* $\langle \mathcal{AS}_{\mathsf{ppr}}, \mathcal{K}, \leq \rangle$ *is called an* Argumentation Theory for Perspective-based Practical Reasoning *if*

- $\mathcal{AS}_{\mathsf{ppr}}$ *an argumentation system for PPR as in Definition 6.4,*
- $\mathcal{K}$ *a knowledge base for PV as in Definition 5.4, and*
- $\leq$ *an argument ordering over arguments in* $\mathcal{AS}_{\mathsf{ppr}}$ *that satisfies the last-link or weakest-link principle.*

Note that the knowledge base in an argumentation theory for perspective-based practical reasoning (ATPPR) has the same necessary premises as a knowledge base for PV. Because

the closure of the necessary premises in a KBPV is consistent under strict rule application, it is straightforward to see that the closure of the necessary premises in a knowledge base of an ATPPR is also consistent under strict rule application. Further note that ATPPRs are well-formed because the contrariness function for PPR does not have any contraries. Consequently, ATPPRs satisfy the rationality postulates in Subsection 2.1.5.

Definition 6.1 specifies outcome mapping functions, which map an alternative to the set of assignments in which performing that alternative can result. We say that a knowledge base $\langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ corresponds to outcome mapping $\mathsf{outcome}$ if and only if $s \in \mathsf{outcome}(a)$ iff $\mathsf{results}(a, s) \in \mathcal{K}_{\mathsf{op}}$.

Example 6.3 (Arguing what movie to watch) This example extends Example 6.2. Agent α is going to reason about what movie to watch for which he uses the argumentation theory for PPR $\mathcal{AT} = \langle \mathcal{AS}_{\mathsf{ppr}}, \mathcal{K}, \leq \rangle$ with $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \emptyset \rangle$. Agent α's knowledge base $\mathcal{K}$ contains the following. Because α cares about the perspectives of fun and length, $\mathsf{infl}(\mathsf{fun}, \alpha)$ and $\mathsf{infl}(\mathsf{length}, \alpha)$ are both in $\mathcal{K}_{\mathsf{op}}$. Both fun and length are important for α in the current decision context, so $\mathsf{cntxt}(\alpha, p)$ and $\mathsf{cntxt}(\alpha, \mathsf{length})$ are also in $\mathcal{K}_{\mathsf{op}}$. Furthermore, because agent α evaluates the genres comedy and drama (assignments s_1 and s_2) as 'good' from perspective fun and a length of 100 minutes (assignment t_1) as 'good' from perspective length, the statements $\mathsf{good}(\alpha, s_1, \mathsf{fun})$, $\mathsf{good}(\alpha, s_2, \mathsf{fun})$ and $\mathsf{good}(\alpha, t_1, \mathsf{length})$ are in $\mathcal{K}_{\mathsf{op}}$. Finally, because choosing alternative a_1 (watching 'Eternal Sunshine of the Spotless Mind') results in assignment u_{21}, it is true that $\mathsf{results}(a_1, u_{21})$ is in $\mathcal{K}_{\mathsf{op}}$.

From agent α's knowledge base it is then possible to justify that agent α should have the achievement goal from perspective fun to achieve assignment s_1. This is done in the following argument, which applies the defeasible rule d_{goal}.

$$A_1 = \frac{\mathsf{infl}(\mathsf{fun}, \alpha) \quad \mathsf{cntxt}(\alpha, \mathsf{fun}) \quad \forall_{s \in \{s_1\}}[\mathsf{good}(\alpha, s_1, \mathsf{fun})]}{\mathsf{goal}(\alpha, \mathsf{fun}, \{s_1\})}\ d_{\mathsf{goal}}$$

Furthermore, because α finds both s_1 and s_2 'good' from fun, the following argument can also be constructed from $\mathcal{K}$.

$$A_2 = \frac{\mathsf{infl}(\mathsf{fun}, \alpha) \quad \mathsf{cntxt}(\alpha, \mathsf{fun}) \quad \forall_{s \in \{s_1, s_2\}}[\mathsf{good}(\alpha, s, \mathsf{fun})]}{\mathsf{goal}(\alpha, \mathsf{fun}, \{s_1, s_2\})}\ d_{\mathsf{goal}}$$

Note that both A_1 and A_2 conclude what achievement goal agent α should have from perspective fun. However, since different achievement goals for the same agent from the same perspective do not conflict, argument A_1 and A_2 do not attack each other. Further note that the achievement goal concluded by A_2 is more general than the one by A_1 (i.e., it contains more assignments). This means that the defeasible rule d_{gUC} can be used as follows to undercut the application of d_{goal} in A_1.

$$A_3 = \frac{\{s_1\} \subset \{s_1, s_2\} \qquad A_2}{\neg\mathsf{appl}(d_{\mathsf{goal}}(\alpha, \mathsf{fun}, \{s_1\}))}\ d_{\mathsf{gUC}}$$

Argument A_3 concludes defeasible rule d_{goal} cannot be applied on just $\{s_1\}$ because it can be applied on a larger set of assignments. Since A_1 applies d_{goal} in this way, argument A_3 undercuts argument A_1.

Recall from Definition 6.5 that a knowledge base in a argumentation theory for PPR is a knowledge base for PV as in Definition 5.4. This means that $\mathsf{satisfies}(s,t)$ is in $\mathcal{K}_{\mathsf{op}}$ if and only if assignment s satisfies assignment t, i.e., s makes the same assignments as t and possibly more. Further recall from Table 6.1 that $\mathsf{achieves}(s,G)$ abbreviates $\exists_{t\in G}[\mathsf{satisfies}(t,s)]$. Consequently, $\mathsf{achieves}(s_1,\{s_1,s_2\})$ and $\mathsf{achieves}(s_2,\{s_1,s_2\})$ are true. Again recall from Table 6.1 that $\mathsf{alwAch}(a,G)$ abbreviates $\forall_{s\in\mathcal{S}}[\mathsf{results}(a,s) \supset \mathsf{achieves}(s,G)]$ or in other words, every outcome in which alternative a may result achieves the set of assignments G. Because the only result of a_1 is u_{21} and u_{21} satisfies s_2, $\mathsf{alwAch}(a_1,\{s_1,s_2\})$ is thus true. In other words, watching the comedy movie 'Eternal Sunshine of the Spotless Mind' always achieves the goal to either watch a comedy or a drama.

We can now construct the following argument that justifies agent α performing alternative a because a always achieves the achievement goal of α that was concluded in argument A_2. It uses the defeasible inference rule d_{alwG} from Table 6.2.

$$A_4 = \frac{A_2 \qquad \mathsf{alwAch}(a_1,\{s_1,s_2\})}{\mathsf{do}(\alpha,a_1)}\ d_{alwG}$$

Because α also cares about perspective length and evaluates assignment t_1 as good from length, α should have the goal to achieve t_1. This is justified in the following argument.

$$A_5 = \frac{\mathsf{infl}(\mathsf{length},\alpha) \quad \mathsf{cntxt}(\alpha,\mathsf{length}) \quad \forall_{s\in\{t_1\}}[\mathsf{good}(\alpha,s,\mathsf{length})]}{\mathsf{goal}(\alpha,\mathsf{length},\{t_1\})}\ d_{\mathsf{goal}}$$

Because alternative a_1 always results in u_{11} and u_{11} satisfies t_1, it is true that $\mathsf{alwAch}(a_1,\{t_1\})$. The following argument can thus be constructed.

$$A_6 = \frac{A_5 \quad \mathsf{alwAch}(a_1,\{t_1\})}{\mathsf{do}(\alpha,a_1)}\ d_{alwG}$$

Both arguments A_4 and A_6 conclude that α should watch Eternal Sunshine of the Spotless Mind (i.e., perform alternative a_1), but for different reasons. Namely, A_4 argues that alternative a_1 is good from the perspective of fun and A_6 that a_1 is good from the perspective of length.

Recall from Section 2.2 that we can transform an argumentation theory into an accrual argumentation theory. By transforming an argumentation theory for PPR into an accrual argumentation theory, it is possible to accrue different reasons for doing a certain alternative.

6.3.2 Meta-Argumentation

In the previous subsection, we proposed an Argumentation System for Perspective-based Practical Reasoning (ASPPR) that extends the argumentation system for perspective-based value introduced in Section 5.1. In an ASPPR it is possible to construct arguments that justify an agent having a goal given that agent's value tree and monadic evaluations of specific assignments. Using means-end reasoning, alternatives can be found that achieve goals of the agent. For each of those alternatives, it is possible to construct arguments in favor and against deciding to do that alternative. It is thus possible that for each decision there are multiple arguments in favor and against. These arguments can be accrued, but in order to decide what

alternative α is justified to do, α should compare the accruals to see which accrual is the strongest.

To argue about the strength of object-level arguments in a ASPPR, we will use the meta-language for PV as the meta-language for perspective-based practical reasoning. We will add several defeasible inference schemas for this purpose. Subsection 6.2.3 informally describes how to justify relative strength of arguments concerning decisions. The defeasible inference schemas in Table 6.3 formalize these intuitions. Argument Scheme 17 describes that if perspective q is more important to agent α than perspective p, then justifying a goal with q creates more conclusive force than justifying a goal with p. Because of the four combinations of achievement and avoidance goals, Argument Scheme 17 is formalized with defeasible inference schemas d'_{gg}, d'_{rr}, d'_{rg} and d'_{gr}.

Table 6.3: *Defeasible Inference Schemas for Meta-level PPR*

Name	Defeasible Inference Schema
Goals	$d'_{gg}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{\mathsf{goal}}(\alpha,p,G)\} \prec_{\mathcal{R}} \{d_{\mathsf{goal}}(\alpha,q,G')\}$ $d'_{rr}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{\mathsf{avoid}}(\alpha,p,R)\} \prec_{\mathcal{R}} \{d_{\mathsf{avoid}}(\alpha,q,R')\}$ $d'_{gr}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{\mathsf{goal}}(\alpha,p,G)\} \prec_{\mathcal{R}} \{d_{\mathsf{avoid}}(\alpha,q,R)\}$ $d'_{rg}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{\mathsf{avoid}}(\alpha,p,R)\} \prec_{\mathcal{R}} \{d_{\mathsf{goal}}(\alpha,q,G)\}$
Decisions	$d_{aG}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{alwG}(\alpha,a,p,G)\} \prec_{\mathcal{R}} \{d_{alwG}(\alpha,b,q,G')\}$ $d_{aR}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{alwR}(\alpha,a,p,G)\} \prec_{\mathcal{R}} \{d_{alwR}(\alpha,b,q,G')\}$ $d_{mG}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{mayG}(\alpha,a,p,G)\} \prec_{\mathcal{R}} \{d_{mayG}(\alpha,b,q,G')\}$ $d_{mR}(\alpha,p,q,G,G') : p \lhd_\alpha q \;\Rightarrow\; \{d_{mayR}(\alpha,a,p,G)\} \prec_{\mathcal{R}} \{d_{mayR}(\alpha,b,q,G')\}$

To argue about the relative strength of (object-level) arguments in an ASPPR (see Definition 6.4), we will now define meta-argumentation systems for perspective-based practical reasoning (meta-ASPPRs) as meta-ASPVs extended with the defeasible rules in Table 6.3.

Definition 6.6 (Meta-Argumentation System for PPR) *Let $\mathcal{AS}_{\mathsf{ppr}}$ an ASPPR. A* meta-argumentation system for PPR *is a meta-argumentation system for PV $\langle \mathcal{L}_{\mathsf{ppr}}, \mathcal{SR} \cup \mathcal{DR}, \mathsf{cf}_{\mathsf{ppr}} \rangle$ w.r.t. $\mathcal{AS}_{\mathsf{ppr}}$ such that*

- *$\mathcal{L}'_{\mathsf{pv}}$ a meta-language for PV w.r.t. $\mathcal{AS}_{\mathsf{ppr}}$,*
- *$\mathcal{SR}$ is the set of all valid first-order inferences,*
- *$\mathcal{DR}$ a set of defeasible inference rules containing those in Table 6.3, and*
- *$\mathsf{cf}_{\mathsf{pv}}$ a contrariness function for $\mathcal{L}'_{\mathsf{pv}}$.*

Object-level defeasible inference rules d_{mayG} and d_{alwG} both conclude $\mathsf{do}(\alpha, a)$, which denotes that agent α should perform alternative a. However, d_{mayG} concludes this because a may achieve an achievement goal of α while d_{alwG} concludes this because a surely achieves an achievement goal of α. Therefore, applying d_{alwG} has more conclusive force than applying d_{mayG}. Premise prem_1 in Table 6.4 formalizes this intuition and should be added as an ordinary premise to every knowledge base. The object-level rules d_{mayR} and d_{alwR} are similar but then with respect to avoidance instead of achievement goals. Therefore, applying d_{alwR} creates more conclusive force not to do a certain alternative than applying d_{mayR}. Premise prem_2 in Table 6.4 formalizes this and should be included as an ordinary premise.

Table 6.4: *Premises for Meta-Level Perspective-based Practical Reasoning*

Name	Premise[a]
prem_1	$\{d_{mayG}(\alpha, a, p, G)\} \prec_{\mathcal{R}} \{d_{alwG}(\alpha, b, p, G)\}$
prem_2	$\{d_{mayR}(\alpha, a, p, G)\} \prec_{\mathcal{R}} \{d_{alwR}(\alpha, a, p, G)\}$

[a] All formulae with free variables are universally quantified.

A meta-knowledge base for PPR is now defined as a meta-knowledge base for PV that includes the ordinary premises in Table 6.4.

Definition 6.7 (Meta-Knowledge Base for PPR) A knowledge base for meta PPR *(meta-KBPPR) is a meta-knowledge base* $\langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ *where* $\mathcal{K}_{\mathsf{np}}$ *contains the axioms Table 5.5 and* $\mathcal{K}_{\mathsf{op}}$ *contains the premises in Table 6.4.*

Meta-KBPPRs extend meta-KBPVs (as in Definition 5.9) by adding the two ordinary premises in Table 6.4. Meta-argumentation theories for PPR are defined as follows.

Definition 6.8 (Meta-Argumentation Theory for PPR) *An argumentation theory* $\langle \mathcal{AS}, \mathcal{K}, \leq \rangle$ *is called a* meta-argumentation theory for PPR *(meta-ATPPR) if*

- $\mathcal{AS}$ *is a meta-argumentation system for PPR,*
- $\mathcal{K}$ *is a meta-knowledge base for PPR, and*
- $\leq$ *is an argument ordering over arguments in* $\mathcal{AS}$ *that satisfies the last-link or weakest-link principle.*

Constructing and evaluating an argumentation framework for a meta-argumentation theory for PPR is straightforward. Because the contrariness function for PPR does not contain any contraries, meta-ATPPRs are well-formed. This also means that meta-ATPPRs satisfy the rationality postulates in Subsection 2.1.5. Grounded argument orderings on the basis of a meta-ATPPR thus follow the last-link principle and can be used in the (object-level) argumentation theory for PPR.

6.4 Running Example

Alternatives

In the running example, there are multiple alternatives that the student can decide to perform. This example focuses on three alternatives $\mathsf{alt}_{\mathsf{re}}$, $\mathsf{alt}_{\mathsf{er}}$ and $\mathsf{alt}_{\mathsf{ere}}$. Recall that alternatives can be single atomic actions or plans and that only one alternative can be chosen. Table 6.5 describes what each alternative denotes and describes the outcome in terms of the attributes in $\mathcal{A}$. In alternative $\mathsf{alt}_{\mathsf{er}}$ the student first extinguishes the entire fire and then rescues the victims[4]. Extinguishing the fire completely takes thirty minutes and if the fire is extinguished fully, then rescuing the victims only takes five minutes. With alternative $\mathsf{alt}_{\mathsf{er}}$, the victims are thus inside for thirty-five minutes. Because the fire is extinguished gradually, after twenty

[4] Because in alternative $\mathsf{alt}_{\mathsf{er}}$ the firefighters first Rescue and then Extinguish, its subscript is re.

minutes the victims are not near the fire. Although $\mathsf{alt}_{\mathsf{er}}$ does result in some smoke, it prevents chemicals from escaping into the environment.

Alternative $\mathsf{alt}_{\mathsf{re}}$ is that the student first rescues the victims and then extinguishes the fire. If the fire is not extinguished at all, then rescuing the victims takes at least ten minutes and possibly even twenty-five minutes. Consequently, with alternative $\mathsf{alt}_{\mathsf{re}}$ the victims are inside for anything between ten and twenty-five minutes, so they are also near the fire for anything between ten and twenty-five minutes just like the firefighters are. Because the firefighters can only start extinguishing when they return rescuing the victims, they will start after ten to twenty-five minutes, resulting in a large amount of smoke and a large amount of chemicals escaping.

In alternative $\mathsf{alt}_{\mathsf{ere}}$ the student first extinguishes the fire near the victims, then gets the victims out and then extinguishes the rest of the fire. Only extinguishing the fire near the victims takes five minutes and then getting them out takes another ten minutes. With alternative $\mathsf{alt}_{\mathsf{ere}}$, the victims are inside the building for fifteen minutes out of which only five are near fire. To rescue the victims, the firefighters need to go near the fire for five minutes.

Table 6.5: *The available alternatives and their outcomes*

Alt.	Denotes	x_1 [a]	x_2	x_3	x_4	x_5	x_6
$\mathsf{alt}_{\mathsf{re}}$	First rescue victims, then extinguish fire	10-25	10-25	lots	lots	10-25	10-25
$\mathsf{alt}_{\mathsf{er}}$	First extinguish fire, then rescue victims	30	5	some	none	15-20	0
$\mathsf{alt}_{\mathsf{ere}}$	First extinguish fire near victims, then rescue victims, then extinguish fire fully	15	10	some	some	5	5

[a] Recall from Table 4.1 that x_1 till x_6 are the attributes in which outcomes are expressed.

Argumentation System

To argue about what agent α (the student) should do, we will now describe the argumentation system for perspective-based practical reasoning (PPR), which requires a language for PPR. To make the language for PPR, we will use the PVCS δ for the running example as described in Section 4.4. Furthermore, as the set of agents we will use only agent α, i.e., $\mathsf{Agents} = \{\alpha\}$, and as the set of alternatives $\mathsf{Alt} = \{\mathsf{alt}_{\mathsf{re}}, \mathsf{alt}_{\mathsf{er}}, \mathsf{alt}_{\mathsf{ere}}\}$. Then, the language for perspective-based practical reasoning w.r.t. δ, Agents and Alt is denoted with $\mathcal{L}_{\mathsf{ppr}}$ and defined as in Definition 6.2. The argumentation system for PPR is then $\langle \mathcal{L}_{\mathsf{ppr}}, \mathcal{R}, \mathsf{cf} \rangle$ as in Definition 6.4.

The assignments that result from alternatives $\mathsf{alt}_{\mathsf{re}}$, $\mathsf{alt}_{\mathsf{er}}$ and $\mathsf{alt}_{\mathsf{ere}}$ are s_{re}, s_{er}, and s_{ere} respectively. Consequently, the knowledge base $\mathcal{K}_{\mathsf{op}}$ of ordinary premises will contain the premises $\mathsf{results}(\mathsf{alt}_{\mathsf{re}}, s_{\mathsf{re}})$, $\mathsf{results}(\mathsf{alt}_{\mathsf{er}}, s_{\mathsf{er}})$, and $\mathsf{results}(\mathsf{alt}_{\mathsf{ere}}, s_{\mathsf{ere}})$.

Goals

Recall the monadic evaluations of student α that are described in Table 4.3. For example, α evaluates an x_1-value of 10 (i.e., that the victims are inside for ten minutes) as good and the x_1-value of 20 as bad from criterion perspective c_1. Because the knowledge base is

instantiated with all these monadic evaluations, it contains, for example, $\mathsf{good}(\alpha, t_1, c_1)$ with assignment t_1 denoting $\{(x_1, 10)\}$.

Using an agent's monadic evaluations and his value tree, achievement and avoidance goals can be justified using defeasible rules d_{goal} and d_{avoid}.

$$\frac{\mathsf{cntxt}(\alpha, c_1) \quad c_1{\downarrow}\alpha \quad \forall_{s\in\{t_1\}}[\mathsf{good}(\alpha, s, c_1)]}{\mathsf{goal}(\alpha, c_1, \{t_1\})}\ d_{\mathsf{goal}}$$

In a similar fashion, achievement goals can be justified for every criterion perspective. Let G_i and R_i denote the sets of assignments that α evaluates as good and bad respectively from criterion perspective c_i. Then for $1 \leq i \leq 6$ the formulae $\mathsf{goal}(\alpha, c_i, G_i)$ and $\mathsf{avoid}(\alpha, c_i, R_i)$ can be justified. For example, for the criterion c_3 of the amount of smoke the achievement goal $\mathsf{goal}(\alpha, c_3, \{t_3\})$ can be justified as follows (with t_3 denoting assignment $\{(x_3, \mathsf{none})\}$):

$$\frac{\mathsf{cntxt}(\alpha, c_3) \quad c_3{\downarrow}\alpha \quad \forall_{x\in\{t_3\}}[\mathsf{good}(\alpha, x, c_3)]}{\mathsf{goal}(\alpha, c_3, \{t_3\})}\ d_{\mathsf{goal}}$$

In a similar way, the following avoidance goal can be justified w.r.t. criterion c_3 (with $t' = \{(x_3, \mathsf{lots})\}$):

$$\frac{\mathsf{cntxt}(\alpha, c_3) \quad c_3{\downarrow}\alpha \quad \forall_{x\in\{t'_3\}}[\mathsf{bad}(\alpha, x, c_3)]}{\mathsf{avoid}(\alpha, c_3, \{t'_3\})}\ d_{\mathsf{goal}}$$

Justifying Alternatives

Alternative $\mathsf{alt}_{\mathsf{re}}$ possibly achieves α's goal w.r.t. criterion perspective c_1 of the time that the victims are inside the factory. As can be seen in Table 6.5, the outcome of doing alternative $\mathsf{alt}_{\mathsf{re}}$ is quite uncertain. Namely, $\mathsf{alt}_{\mathsf{re}}$ may result in any x_1-value between 10 and 25. The student α has the achievement goal of an x_1-value of 10 or less (denoted set G_1) and the avoidance goal of x_1-values of 20 or more (denoted R_1) from perspective c_1. It is thus possible that $\mathsf{alt}_{\mathsf{re}}$ achieves α's achievement goal w.r.t. c_1, but it is also possible that it achieves α's avoidance goal w.r.t. c_1. Consequently, the following arguments can be constructed.

$$A^{c_1}_{mAch}(\mathsf{alt}_{\mathsf{re}}) = \frac{\mathsf{goal}(\alpha, c_1, G_1) \quad \mathsf{mayAch}(\mathsf{alt}_{\mathsf{re}}, G_1)}{(\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{re}}))^{mAch(c_1)}}\ d_{mAch}$$

Recall from Section 2.2 that in an accrual argumentation system the conclusions of defeasible rules are labeled with the application of the rule. For simplicity we have labeled them with the most important aspect of the rule, which is in this case $mAch(c_1)$. Because $\mathsf{alt}_{\mathsf{re}}$ may also result in x_1-values of 20 till 25 it is possible that $\mathsf{alt}_{\mathsf{re}}$ achieves an avoidance goal of α. This creates the following argument for α not to do $\mathsf{alt}_{\mathsf{re}}$.

$$A^{c_1}_{mAv}(\mathsf{alt}_{\mathsf{re}}) = \frac{\mathsf{avoid}(\alpha, c_1, R_1) \quad \mathsf{mayAv}(\mathsf{alt}_{\mathsf{re}}, R_1)}{(\neg\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{re}}))^{mAch(c_1)}}\ d_{mAv}$$

The agent has the avoidance goal to achieve a large amount of smoke. Because $\mathsf{alt}_{\mathsf{re}}$ always results in a large amount of smoke, the following argument can be constructed.

$$A^{c_3}_{alwAv}(\mathsf{alt}_{\mathsf{re}}) = \frac{\mathsf{avoid}(\alpha, c_3, R_3) \quad \mathsf{alwAch}(\mathsf{alt}_{\mathsf{re}}, R_3)}{(\neg\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{re}}))^{alwAv(c_3)}}\ d_{alwAv}$$

In a similar fashion, if an alternative x surely / possibly achieves α's achievement goal w.r.t. criterion perspective c_i, then an accrual argument can be constructed that concludes $(\mathsf{do}(\alpha, x))^{alwAch(c_i)}$ and $(\mathsf{do}(\alpha, x))^{mAch(c_i)}$ respectively. If x surely / possibly achieves α's avoidance goal w.r.t. criterion c_i, then an accrual argument can be constructed concluding $(\neg\mathsf{do}(\alpha, x))^{alwAv(c_i)}$ and $(\neg\mathsf{do}(\alpha, x))^{mAv(c_i)}$. Table 6.6 describes what achievement and avoidance goals alternatives $\mathsf{alt_{re}}$, $\mathsf{alt_{er}}$ and $\mathsf{alt_{ere}}$ may achieve. A + and - denote that the alternative may achieve α's achievement and avoidance goal respectively. A 0 denotes that the alternative may not achieve either the achievement or avoidance goal. Finally, $+/-$ denotes that the alternative may achieve the achievement goal, but may also achieve the avoidance goal.

Table 6.6: *Achievement and avoidance goals achieved by the alternatives*

Criterion	$\mathsf{alt_{re}}$	$\mathsf{alt_{er}}$	$\mathsf{alt_{ere}}$
c_1	+/-	-	0
c_2	-	0	-
c_3	-	0	0
c_4	-	+	+
c_5	-	0/-	+
c_6	-	+	0

Because we are dealing with many arguments concerning what alternative to do, we will use the following abbreviations for arguments: (1) $A^{c_i}_{mAch}(a)$ denotes the argument that concludes $(\mathsf{do}(\alpha, a))^{mayA(c_i)}$; (2) $A^{c_i}_{alwAch}(a)$ denotes the argument that concludes $(\mathsf{do}(\alpha, a))^{alwA(c_i)}$; (3) $A^{c_i}_{mAv}(a)$ denotes the argument that concludes $(\neg\mathsf{do}(\alpha, a))^{mAv(c_i)}$; and (4) $A^{c_i}_{alwAv}(a)$ denotes the argument that concludes $(\neg\mathsf{do}(\alpha, a))^{alwAv(c_i)}$. For example, the - in Table 6.6 for c_2 and $\mathsf{alt_{re}}$ means that there is an argument $A^{c_2}_{alwAv}(\mathsf{alt_{re}})$. Also, the $0/-$ for c_5 and $\mathsf{alt_{er}}$ means that there is an argument $A^{c_2}_{mayAv}(\mathsf{alt_{er}})$ and the $+/-$ for c_1 and $\mathsf{alt_{re}}$ means that there is an argument $A^{c_1}_{mayAch}(\mathsf{alt_{re}})$ and an argument $A^{c_1}_{mayAv}(\mathsf{alt_{re}})$.

Accruing

Because $\mathsf{alt_{re}}$ only possibly achieves α's achievement goal w.r.t. c_1, only this argument can be accrued to conclude $\mathsf{do}(\alpha, \mathsf{alt_{re}})$. In contrast, $\mathsf{alt_{er}}$ surely achieves two achievement goals of α w.r.t. the amount of escaping chemicals and the time that the firefighters have to go inside the factory near fire (criteria c_4 and c_6). Consequently, the following two accrual arguments can be constructed.

$$A^{+}_{\mathsf{re}} = \frac{A^{c_1}_{mayAch}(\mathsf{alt_{re}})}{\mathsf{do}(\alpha, \mathsf{alt_{re}})} \qquad A^{+}_{\mathsf{er}} = \frac{A^{c_4}_{alwAch}(\mathsf{alt_{er}}) \quad A^{c_6}_{alwAch}(\mathsf{alt_{er}})}{\mathsf{do}(\alpha, \mathsf{alt_{er}})}$$

Alternative $\mathsf{alt_{ere}}$ has different advantages: $\mathsf{alt_{ere}}$ surely achieves α's achievement goals w.r.t. the amount of escaping chemicals and the time that the victims are inside the factory near fire (criteria c_4 and c_5).

$$A^{+}_{\mathsf{ere}} = \frac{A^{c_4}_{alwAch}(\mathsf{alt_{ere}}) \quad A^{c_5}_{alwAch}(\mathsf{alt_{ere}})}{\mathsf{do}(\alpha, \mathsf{alt_{ere}})}$$

We can also look at the avoidance goals that the alternative achieves. Recall that an avoidance goal is something that the agent wants to avoid, so achieving an avoidance goal is not good. Because alternative $\mathsf{alt}_{\mathsf{re}}$ surely achieves the avoidance goals w.r.t. criteria c_2 up until c_6 and possibly achieves α's avoidance goal w.r.t. criterion c_1, six different arguments can be accrued concluding that α should not do alternative $\mathsf{alt}_{\mathsf{re}}$.

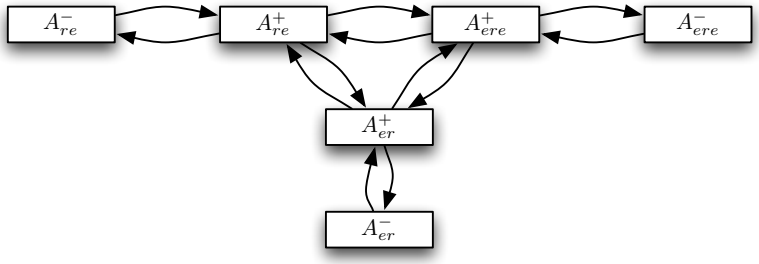
$$A^-_{\mathsf{re}} = \frac{A^{c_1}_{mayAv}(\mathsf{alt}_{\mathsf{re}}) \quad A^{c_2}_{alwAv}(\mathsf{alt}_{\mathsf{re}}) \quad \dots \quad A^{c_6}_{alwAv}(\mathsf{alt}_{\mathsf{re}})}{\neg\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{re}})}$$

In contrast, alternative $\mathsf{alt}_{\mathsf{er}}$ achieves α's avoidance goals w.r.t. criteria perspectives c_1 and c_5. Finally, alternative $\mathsf{alt}_{\mathsf{ere}}$ only achieves α's avoidance goal w.r.t. criterion perspective c_2. This results in the following two accrual arguments concluding that alternatives $\mathsf{alt}_{\mathsf{er}}$ and $\mathsf{alt}_{\mathsf{ere}}$ respectively should not be done.

$$A^-_{\mathsf{er}} = \frac{A^{c_1}_{alwAv}(\mathsf{alt}_{\mathsf{er}}) \quad A^{c_5}_{mayAv}(\mathsf{alt}_{\mathsf{er}})}{\neg\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{er}})} \qquad A^-_{\mathsf{ere}} = \frac{A^{c_2}_{alwAv}(\mathsf{alt}_{\mathsf{ere}})}{\neg\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{ere}})}$$

The attack relations between these accrual arguments are visualized in Figure 6.2. Recall from Definition 6.3 that $\mathsf{do}(\alpha, a)$ conflicts with $\mathsf{do}(\alpha, b)$.

Figure 6.2: *Attacks between accrual arguments about what to do and not to do*



If all these accrual arguments were equally strong, then in the corresponding argumentation framework A_{re}, A_{er} and A_{ere} are all defensible conclusions and there is no justified conclusion concluding what the student should do.

Meta-Argumentation

Since the student cares more about some criteria than other criteria, not all arguments in favour and against the various alternatives are equally strong for the student. In order to compare the strength of object-level arguments, we need to define the relative strengths of accrual arguments. As we will describe in more detail in Subsection 6.5.2, this is a difficult problem which we do not address in this thesis. However, we will give some intuition about how these accrual arguments can be compared.

Alternative $\mathsf{alt}_{\mathsf{re}}$ achieves the student's avoidance goals w.r.t. five criteria perspectives and only achieves one achievement goal. It is thus plausible that A^-_{re} is a stronger argument than A^+_{re}. Because A^-_{re} and A^+_{re} attack each other, but A^-_{re} is stronger, only the attack of A^-_{re} is successful. Consequently, A^+_{re} is not a justified argument.

Because c_5 is more important than c_2, A^-_{er} is a stronger argument than A^-_{ere}. This means that alternative $\mathsf{alt}_{\mathsf{er}}$ has a stronger disadvantage than alternative $\mathsf{alt}_{\mathsf{ere}}$. Both $\mathsf{alt}_{\mathsf{er}}$ and $\mathsf{alt}_{\mathsf{re}}$ achieve the student's achievement goal w.r.t. criterion perspective c_4. However, $\mathsf{alt}_{\mathsf{er}}$ also achieves the achievement goal w.r.t. c_6 while $\mathsf{alt}_{\mathsf{ere}}$ also achieves the achievement goal w.r.t. c_5. The relative importance of c_5 and c_6 is incomparable for the student. Consequently, A^+_{er} and A^+_{ere} are incomparable in strength. Because A^+_{er} and A^+_{ere} are incomparable, but alternative $\mathsf{alt}_{\mathsf{er}}$ has a stronger disadvantage than alternative $\mathsf{alt}_{\mathsf{ere}}$, we could say that A^+_{ere} is a justified argument. In that case, the justified conclusion for the student is to choose alternative $\mathsf{alt}_{\mathsf{ere}}$.

6.5 Related Work

In this section we will discuss the similarities and differences between the approach of this chapter and the approaches by Atkinson et al. (2006) and Amgoud and Prade (2009). First, Subsection 6.5.1 explains how our approach extends the approach by Atkinson et al. (2006) by allowing to specify what an abstract value means to an agent, to argue about what outcome is better from the perspective of an abstract value, and by making the relationship between goals and abstract values explicit. In the approach of this chapter it is possible to construct arguments in favor and against decisions. Since there may be multiple of such arguments it is important to accrue those arguments. Subsection 6.5.2 discusses related work on different kinds of accrual rules for such arguments. Finally, Subsection 6.5.3 explains how we extend Amgoud and Prade's (2009) approach by allowing to justify goals and priorities between goals.

6.5.1 Atkinson et al.

We will now compare our approach to practical reasoning to the approach by Atkinson et al. (2006), who propose argument scheme AS1 that an agent can use to argue about what action he should perform. AS1 justifies an action if that action achieves a goal that promotes the agent's values. Argument scheme AS1 is as follows.

> In the current circumstances R,
> Agent 1 should perform action ac1,
> Which will result in new circumstances S,
> Which will realize goal G,
> Which will promote value V.

Our approach is similar but differs in several ways. Table 6.7 summarises how the approaches differ and Table 6.8 describes how the critical questions associated to AS1 can be interpreted in our approach. But first, we will justify why Argument Schemes 13 and 15 are similar to AS1. Namely, Argument Schemes 13 and 15 justify that an agent does an alternative if that alternative (possibly or always) achieves a goal of the agent. In our formalization, the formula $\mathsf{results}(a, s)$ denotes that alternative a results in assignment s. This could be translated to the terminology of AS1 as 'in the current circumstances, alternative a results in new circumstances s'. Note that in contrast to AS1, our approach does not explicitly represent the current circumstances, but implicitly represents this by stating in what outcome performing alternatives result. The formula $\mathsf{goal}(\alpha, p, G)$ denotes that agent α has the achievement

goal w.r.t. perspective p to achieve an assignment in G. This means that $\mathsf{goal}(\alpha, p, G)$ in combination with $\mathsf{results}(a, s)$ could be translated into 'in the current circumstances, alternative a results in new circumstances s, which achieves α's achievement goal G'. Argument Schemes 13 and 15 are thus similar to AS1 except that they do not concern what values of the agent are promoted. Instead, the values that an alternative promotes are considered in the justification of an agent wanting to achieve a certain goal. Argument Scheme 11 justifies an agent having an achievement goal w.r.t. a certain perspective p if p influences a value that the agent pursues, p is relevant in the decision context and the agent evaluates the goal assignments as 'good' from p.

Table 6.7: *Atkinson et al.'s (2006) approach compared to PVM approach*

AS1	**Our Approach**
One 'big' argument scheme	Multiple smaller argument schemes
Valuation function is input to the system and exactly defines what constitutes a value. Not possible to discuss what a value means.	What constitutes a value is represented in a value tree and is open for debate
Valuation function completely specifies what transitions promote and demote what values	Argumentation about what transitions promote and demote values
Goals are input to the system	Given a value tree and monadic evaluations, goals are justified
Only achievement goals[a]	Distinction between achievement and avoidance goals
Priorities between values, not between goals	Priorities between values and between goals
No uncertainty in whether an alternative achieves a goal	Rough distinction between always achieving and possibly achieving
Explicit representation of the current circumstances	Current circumstances implicitly represented in the attributes of outcomes
Formalization in AATS	Formalization in ASPIC+

[a] Atkinson and Bench-Capon (2007b) discuss avoidance goals and relate it to demoting values.

In AS1, the notions of promoting and demoting values are used to justify decisions. A so-called valuation function maps a value and a state transition to 'promotes', 'demotes' or 'neutral'. This valuation function completely determines when a transition promotes and demotes a value and thus what the value means. The valuation function is input to the system and cannot be changed or discussed. Atkinson and Bench-Capon (2007b) do allow agents having different valuation functions, but their approach does not permit further arguments to resolve this.

As explained in Section 2.5, research in psychology has investigated what values people in all parts of the world find important. Schwartz (1992) proposes ten basic values that are found universally. Recall that perspective structures (see Definition 4.6) contain a set $\mathcal{P}_v$ for perspectives representing general areas of concern, i.e., values. The ten basic values of Schwartz can be modeled by putting a perspective in $\mathcal{P}_v$ for each basic value. If an agent α pursues basic value v, then v positively influences α's perspective. What a basic value means for a particular agent is then decomposed into perspectives that are more specific until

those perspectives represent specific evaluation criteria. If criterion perspective p measures value $v \in \mathcal{P}_v$, i.e., p positively or negatively influences v, and alternative a achieves agent α's achievement goal w.r.t. p, then we say that *alternative a promotes value v*. In contrast, if alternative a achieves agent α's avoidance goal w.r.t. p, then we say that *alternative a demotes value v*.

Several critical questions of argument scheme AS1 concern the values that are promoted and demoted. Critical question CQ3a asks whether the action also promotes another value. In our system this can be translated as whether there is another basic value $v' \in \mathcal{P}_v$ such that the alternative promotes v'. Critical question CQ4a asks whether the action has a side effect which demotes the value v. In our system this translates to whether there is another specific evaluation criterion $q \in \mathcal{P}_c$ such that the alternative achieves α's avoidance goal w.r.t. q. Similarly, critical question CQ4b asks whether the action demotes some other value, which is translated to whether there is another value $v' \in \mathcal{P}_v$ such that the alternative demotes v'.

As explained in Section 2.1, ASPIC+ is an abstract framework for structured argumentation. By using ASPIC+ it is straightforward to extend our argumentation framework. For example, in our approach it is not possible to justify that doing an alternative results in a particular assignment. However, it is straightforward to enrich the logical language with more predicates (e.g., that state properties of the current circumstances) and to add defeasible rules that infer the outcome of an alternative.

6.5.2 Decision Rules

If the object-level argument system is an accrual argumentation system, then its corresponding meta-argumentation system can be used to compare the relative strength of accrual arguments. Some work has been done on how to compare accrual arguments in decision making. For example, Dubois et al. (2008) propose a number of qualitative decision rules that specify what decision to prefer given arguments pro and con the decisions and an ordinal ranking over these arguments by importance.

In Bonnefon and Fargier (2006) and Bonnefon et al. (2008), 62 human subjects were given 33 situations where they had to choose one from two alternatives. Each alternative was described as a list of arguments pro and con. Using this data, they tested how well seven decision rules could predict what decision subjects would make. The $\preceq^{\mathsf{lexi}}$ rule (also called the level-wise tallying heuristic) scans the arguments pro and con of each alternative from the most important to the least important important. If a level of importance is reached such that one decision has more arguments pro or con, then the decision with most arguments pro or the least arguments con is chosen. The $\preceq^{\mathsf{lexi}}$ rule ordered the decisions in the same way as the human subjects in 80 percent of the cases. Bench-Capon et al. (2011) propose a formal two-phase model of democratic policy deliberation and formalize the $\preceq^{\mathsf{lexi}}$ rule to accrue reasons for and against proposals. In future work, it would be useful to formalize the $\preceq^{\mathsf{lexi}}$ rule in the meta-level argumentation system to determine the relative strength of accrual arguments.

Bench-Capon and Prakken (2006) use Prakken (2005b)'s accrual mechanism to accrue arguments pro and con decisions. Arguments w.r.t. decisions are constructed using Atkinson et al. (2006)'s practical reasoning scheme. A function v is assumed that returns the sets of promoted and demoted values given an argument pro or con a decision. For example, if argument A is an argument pro some decision, then $v(A) = \langle \mathsf{prom}, \mathsf{dem} \rangle$ is calculated, where prom is the set of promoted values and dem the set of demoted values. The relative strength

Table 6.8: *Critical Questions of AS1 in ASPPR*

Critical Questions for AS1	ASPPR
CQ1a: Are there alternative ways of realizing the same consequences?	Is there an alternative b such that $\mathsf{results}(b, s)$?
CQ1b: Are there alternative ways of achieving the same goal?	Is there an alternative b such that $\mathsf{alwAch}(b, G)$ or $\mathsf{mayAch}(b, G)$?
CQ1c: Are there alternative ways of promoting the same value?	Is there another alternative that achieves a goal w.r.t. another criterion for value v?
CQ2: Is it possible to do action A?	Is $\mathsf{results}(a, s)$ really true?
CQ3a: Would doing action A promote some other value?	Is there a different value v' that α pursues such that alternative a promotes v'?
CQ3b: Does action A preclude another action which promotes another value?	Is there an alternative b that promotes a different value that α pursues?
CQ4a: Does doing action A have a side effect which demotes the value V?	Does a demote value v on some other criterion perspectives that influences v?
CQ4b: Does doing action A have a side effect which demotes some other value?	Is there another value that α pursues which is demoted by alternative a?
CQ5a: Are the believed circumstances R really possible?	Not applicable, requires extension.
CQ5b: Are the believed circumstances R really true?	Not applicable, requires extension.
CQ5c: Assuming both of these, does action A have the stated consequences S?	Is $\mathsf{results}(a, s)$ really true?
CQ5d: Assuming all of these, does action A really bring about the desired goal G?	Is $\mathsf{alwAch}(a, G)$ really true?
CQ6a: Whether goal G does realize the value intended	Does criterion p really influence value v and are all assignments in G really good?
CQ6b: Whether the value proposed is indeed a legitimate value.	Do criterion p and value v really influence α and are they important in the decision context?
CQ7a: Whether the situation S believed by agent a to result from doing action A is a possible state of affairs.	Is $\mathsf{results}(a, s)$ really true? Namely, if s is not possible, then it can never be the result of an alternative.
CQ7b: Whether the particular aspects of situation S represented by G are possible.	Is $\mathsf{results}(a, s)$ really true? Alternatives cannot result in impossible assignments.

of accrual arguments is then defined using a preference ordering over the sets of promoted and demoted values. The authors refer to Amgoud (2005) for related definitions to order sets of promoted and demoted values.

Another approach to accrue arguments is described in the paper by Amgoud and Prade (2009). This approach will be discussed in more detail in the next subsection.

6.5.3 Amgoud and Prade

Amgoud and Prade (2009) unify a number of papers written by Amgoud and Prade, such as,

Amgoud et al. (2005) and Amgoud and Prade (2006) in order to propose an argumentation-based framework for decision making. The framework distinguishes between so-called epistemic arguments and practical arguments. An epistemic argument supports beliefs, whereas practical arguments justify or refute candidate decisions. As already explained in Subsection 6.1.3, in their framework agents have a set of goals and a set of rejections, which explicate what an agent wants to achieve and to avoid respectively. Practical arguments conclude whether a candidate decision achieves a goal or rejection.

Two steps are then taken to compare candidate decisions. Firstly, both epistemic and practical arguments are evaluated using acceptability semantics. This step results in a number of arguments that are accepted, i.e., arguments of which the conclusions are justified. Secondly, candidate decisions are compared using the accepted practical arguments. Several kinds of so-called *decision principles* are proposed that use the arguments pro and con to compare candidate decisions. *Unipolar principles* either use the goals or use the rejections that an alternative achieves to compare decisions. For example, the more goals that are achieved, the more preferred a decision is. *Bipolar principles* use both goals and rejections to compare decisions. Finally, *non-polar principles* are described as principles that accrue arguments pro and con candidate decisions into a unique meta-argument having a unique strength. Comparing two candidate decisions is then done by comparing their corresponding meta-arguments. Unipolar and bipolar decision principles can be implemented in a non-polar principle. For example, consider the unipolar principle 'the more arguments pro a candidate decision, the more preferred it is'. This principle could be implemented as: the meta-argument that accrues all arguments pro and con candidate decision d_1 is stronger than the meta-argument that accrues all arguments pro and con candidate decision d_2 if and only if there are more arguments pro d_1 accrued than there are arguments pro d_2 accrued.

We will now compare Amgoud and Prade's framework (for easy reference we will call this APF) to the framework proposed in this and the previous chapter. Table 6.9 summarizes the most important differences and some similarities. Both frameworks consider alternatives (called options or candidate decisions in the APF) in a similar fashion. In our framework, arguments that conclude in what assignment performing an alternative results can be seen as epistemic arguments in APF and agents can have achievement and avoidance goals, which correspond to APF's notions of goals and rejections.

APF distinguishes between four kinds of practical arguments w.r.t. a candidate decision d: (1) positive arguments pro d conclude that d achieves a goal; (2) negative arguments pro d conclude that d successfully avoids a rejection; (3) positive arguments con d conclude that d achieves a rejection; and, (4) negative arguments con d conclude that d does not achieve a goal. In our framework we can make a similar distinction. Table 6.10 describes to what these different positive / negative arguments pro / con correspond in our framework.

The part of our framework that allows arguing about what achievement and avoidance goals an agent should have can be used to extend the APF since this is not possible in the APF. If the APF is extended in this way, then the APF does not require specifying what goals and rejections an agent has or how important they are. Instead, this extension requires that the agent's value tree, monadic evaluations, and how important the agent finds the criteria perspectives in his value tree. This enables agents to argue about what abstract values to hold, what objectives to pursue, how to measure the performance of those objectives and what goals to adopt.

In contrast, the decision principles in the APF can be used to replace the part of our

Table 6.9: *Amgoud and Prade (2009)'s approach compared to our PVM approach*

Amgoud and Prade	Our Approach
Alternatives are abstract entities and only one can be chosen	The same
Requires goals and rejections of decision maker	Achievement and avoidance goals are justified from the decision maker's value tree and monadic evaluations
Requires ordering over goals and rejections by importance	Meta-level argumentation for reasoning about the relative importance of achievement and avoidance goals
Distinction between epistemic and practical arguments	No distinction between epistemic and practical arguments: both kinds of arguments are evaluated in the same way
Preferences must be transitive	Preferences are typically transitive, but exceptions are allowed
Three kinds of decision principles to compare candidate decisions	General argument accrual mechanism that can be used to accrue arguments in favour and against a decision, but no specific ways to determine the relative strength of accrual arguments

Table 6.10: *Kinds of practical arguments and how they correspond to our framework*

APF distinguishes between:	Which corresponds to:
positive arguments pro a^a conclude that a achieves a goal	an argument concluding $\mathsf{goal}(\alpha, p, G) \wedge \mathsf{alwAch}(a, G)$ (or $\mathsf{mayAch}(a, G)$)
negative arguments pro a conclude that a successfully avoids a rejection	an argument concluding $\mathsf{avoid}(\alpha, p, G) \wedge \neg\mathsf{alwAch}(a, G)$ (or $\neg\mathsf{mayAch}(a, G)$)
positive arguments con a conclude that a achieves a rejection	an argument concluding $\mathsf{avoid}(\alpha, p, G) \wedge \mathsf{alwAch}(a, G)$ (or $\mathsf{mayAch}(a, G)$)
negative arguments con a conclude that a does not achieve a goal	an argument concluding $\mathsf{goal}(\alpha, p, G) \wedge \neg\mathsf{alwAch}(a, G)$ (or $\neg\mathsf{mayAch}(a, G)$)

[a] Here, a denotes a candidate decision, which is the same as what we call an alternative.

framework that is used to compare decisions. The arguments described in Table 6.10 can be used as input for either APF's unipolar or bipolar decision principles. By using unipolar or bipolar decision principles to compare what decision the decision maker should prefer it becomes unnecessary to use the accrual mechanism described in Section 2.2 in combination with this adapted version of our framework. Note that it is still necessary to use the meta-level argumentation system in order to determine the relative strength of the decision maker's goals.

6.6 Chapter Summary

In the previous chapters an argumentation-based framework was proposed that allows justifying and refuting what outcome of a decision an agent values more. What an agent values is

decomposed into the abstract values he finds important, which are then further decomposed into specific evaluation criteria. If a decision maker chooses an alternative, i.e., he makes a decision, then it may not be known in advance in what outcome that decision results. For decision making it was thus required to extend the framework of the previous chapters with a mechanism to determine what decision should be preferred by the decision maker. This chapter proposed one way how this can be done.

Ideally, the decision maker should compare all alternatives on all criteria he cares about and choose the alternative that is the best on all criteria. As pointed out in the beginning of this chapter, in realistic scenarios this results in a complex comparison that takes many resources to perform. In the decision scenarios in which we are interested there is a limited amount of time available. Therefore, it was necessary to use a different approach to make a decision. The approach proposed in this chapter was inspired by Simon (1957) and does not aim to find the best decision, but a decision that is satisfactory to the decision maker. Recall that criteria were modeled as perspectives, which are orderings over outcomes. This chapter simplified criteria by splitting them into three (possibly empty) sets: a set containing satisfactory or good outcomes, a set with unsatisfactory or bad outcomes, and a set with outcomes that are neither good nor bad. If a certain assignment is good from a certain perspective to the decision maker, then he is justified to have the achievement goal to achieve that assignment. Similarly, if a certain assignment is bad, then he is justified to have the avoidance goal to avoid that assignment. Note that we went from an ordering (which can be used to express subtle differences in value) to an achievement goal, an avoidance goal and a set of outcomes that are neutral to the decision maker.

Whether an alternative achieves a goal was then used to find reasons in favor and against decisions. Namely, if an alternative achieves an achievement goal of the decision maker, then that is a reason for him to choose that alternative. However, if an alternative achieves an avoidance goal, then that is a reason not to choose that alternative. The object-level argumentation system of the previous chapter was then extended such that these kinds of inferences can be made. Because not all criteria may be equally important to a decision maker, the meta-level argumentation system of the previous chapter was extended such that these arguments can be compared in strength. An accrual mechanism can then be used to accrue all the reasons in favour and against decisions. By evaluating what arguments are justified, the decision maker can determine what decision he can justify making. The extended argumentation system was then illustrated on the running example from the introduction after which we compared it to some related work in the literature.

7 A Dialogue Framework for Supporting Decisions

When faced with the need to choose, human decision makers often use argumentation to resolve conflicts and to justify their choice to themselves and to others (Shafir et al., 1997). In situations where there is little time but decisions have critical consequences, it is necessary to consider all the important aspects and to use the right criteria to determine what decision is better. The previous chapter proposed an argumentation framework for decision-making. In this framework, arguments can be constructed that use those aspects that are important to an agent to justify what decision the agent should prefer. Dialogue is a natural way for people to support each others' decisions by exchanging arguments justifying what decision to take. In this chapter, we propose a dialogue framework that allows an artificial agent and a (human) user to put forward arguments and ask questions in order to reason collectively about what decision is the best for the user. By means of such a dialogue, the user could test his justification behind a decision. Moreover, the agent could support the user by putting forward arguments taking into account perspectives that the user may have forgotten and making the user aware of mistakes in his line of reasoning.

In a training situation such as in the running example, both the student and the decision support agent have knowledge that they use to construct arguments concerning what decision to take. Although the decision support agent may be familiar with the decision scenario, he may not be familiar with what the student values, i.e., the student's motivation. The dialogue may start with the user putting forward the argument that he should first rescue the victims and then extinguish the fire because he has the goal to get the victims out of the building within ten minutes. Although this alternative gets the victims out quickly, it is risky for the firefighters who have to go inside the burning factory. The supporting agent could put forward an argument stating that the user should not send the firefighters in because of their safety. Furthermore, the supporting agent could ask the student why he uses the criterion 'the time that the victims are inside' rather than the criterion 'the time that the victims are near the fire'. The latter criterion invites the user to consider the alternative of first extinguishing the fire near the victims and then get them out. This alternative greatly reduces the risk for the firefighters who have to go inside. Moreover, because the victims may be near fire for a smaller amount of time, it may actually be safer for the victims as well.

The Argumentation System for Perspective-based Practical Reasoning (ASPPR) as proposed in Chapter 6 allows an agent to argue about what he values and to argue about what achievement and avoidance goals his value tree justifies. The alternatives that are available

to an agent may have multiple advantages and disadvantages with respect to the goals of the agent that they achieve. The meta-ASPPR is an argumentation system that is on a meta-level with respect to an ASPPR and allows agents to argue about the strength of arguments pro and con decisions and to argue about the effect of this on what decisions are justified. To support a decision it is important to discuss the relative strength of arguments pro and con decisions, which means that it must be possible to put forward both object and meta-level arguments.

This chapter addresses research question 2a of how we can formally represent a dialogue framework for arguing to motivate decisions. First, Section 7.1 discusses several approaches in the literature that are relevant for supporting a decision. We shall argue that for our purposes these approaches need to be extended. Section 7.2 proposes a dialogue framework that extends the approach by Prakken (2005a) to enable agents to put forward arguments but also preferences over arguments. Subsections 7.2.2 and 7.2.3 propose a dialogue framework and protocol respectively that are tailored for decision support dialogues. To support a user properly in making a decision, the agent must understand what motivates the user. Therefore, it is important that the agent updates what he knows about the user based on the arguments the user gives in the dialogue. Section 7.3 proposes how an agent should do this given the dialogue moves he observes. Finally, in Section 7.4 we will demonstrate how the extended dialogue framework that is proposed in this chapter can be used in the running example from the introduction.

7.1 Related Work

First we will discuss a popular taxonomy for dialogue types to determine what kind of dialogue a decision support dialogue is in Subsection 7.1.1. Next, in Subsection 7.1.2, we will focus on one particular dialogue type, namely deliberation dialogues, because deliberation dialogues and dialogues for decision support have several similarities.

7.1.1 Dialogue Types

Walton and Krabbe (1995b) proposed a typology of human dialogue types. This typology has been influential in the literature. Many dialogue systems in the literature are classified using this typology and therefore we will investigate what dialogues types are important for the research questions in this thesis. Six types of dialogues are distinguished based on what information the participants have at the start of the dialogue, their private goals for the dialogue and their shared goals.

- In an *information-seeking dialogue*, one participant has the goal of getting some question answered by another participant.
- In an *inquiry dialogue*, the participants collaborate to answer one or more questions whose answers are not known to any participants.
- In a *persuasion dialogue*, one participant has the goal to persuade another participant to accept some proposition the other participant does not endorse.
- In a *negotiation dialogue*, the participants bargain over the division of some scarce resource.
- In a *deliberation dialogue*, the participants collaborate to decide what alternative should be chosen in a particular situation.

- In an *eristic dialogue*, participants quarrel verbally.

Dialogues that people typically have involve a mixture of these types of dialogue types (McBurney and Parsons, 2009). For example, a dialogue may start with information-seeking where agent α asks agent β whether some proposition is true and the reasons why. Agent β may then give a number of arguments supporting his position. Assume that agent α disagrees with one of these arguments. Then the dialogue may progress to persuasion, where agent α tries to persuade agent β that a certain argument is wrong. In that persuasion dialogue, agent α may say something that insults agent β and the dialogue may progress to a quarrel.

The main goal of a decision support dialogue is to help the user in determining what decision is the best for the user to make. In doing so, the system and user collaborate on finding the answer to this question, which means that decision support dialogues can be classified as inquiry dialogues. In contrast, because in a decision support dialogue the system and the user collaborate on deciding what alternative the user should choose in a certain situation, decision support dialogues can also be classified as deliberation dialogues.

In a dialogue to support a decision, several other dialogue types can occur. For example, the dialogue may start with the system trying to understand what the user values. In that part of the dialogue, the user expresses his general areas of concern that matter to him, intermediate objectives and criteria that specify in detail what general areas of concern mean to the user. When this is taking place, the dialogue could be classified as information-seeking. If the system thinks that the user is forgetting some area of concern, objective or criterion, then the system could propose that a new element should be considered. If the user disagrees, then the system and the user may argue about whether the element should be considered. In that case, the dialogue could be classified as persuasion because the system is trying to persuade the user of the importance of that element, but it might as well be classified as inquiry because the system and the user are collaborating to find the best answer. The dialogue progresses similarly when determining what goals the agent should pursue and determining what decision best fulfills the goals of the agent. In summary, dialogues for decision support consist of a mixture of dialogue types as defined in Walton and Krabbe (1995b).

7.1.2 Deliberation Dialogues

Walton and Krabbe (1995b) argue that the purpose of deliberation dialogue is that a group of agents collaborates finding an appropriate joint action. This thesis is not concerned with finding the best joint action, but with finding the best decision for an individual agent. We call a dialogue with the goal to find the best decision for a single agent a *decision support dialogue*. Decision support dialogues and deliberation dialogues thus have similarities but also differences.

McBurney et al. (2007) present a formal dialogue game for deliberation dialogue in which the dialogue is split into eight stages: (1) in the *open* stage the question of what to do is raised; (2) in the *inform* stage goals, constraints, criteria and relevant facts are discussed; (3) in the *propose* stage alternatives are suggested; (4) in the *consider* stage participants comment on the proposed alternatives; (5) in the *revise* stage participants can revise goals, constraints, criteria, and alternatives; (6) in the *recommend* stage alternatives can be recommended and the acceptability of the recommendations can be discussed; (7) in the *confirm* stage the acceptance of recommendations is confirmed; and, (8) in the *close* stage the deliberation dialogue is closed.

A deliberation dialogue can only be in one stage simultaneously, but it is not required that the stages occur in the given order. Several rules are proposed that govern what sequences of stages are valid. For example, a deliberation dialogue must always start in the open stage, end in the close stage, and the open and close stages can only occur once. Ten locutions are proposed that agents can use in a deliberation dialogue: open dialogue, enter dialogue, propose, assert, prefer, ask justification, move, reject, retract, and withdraw from dialogue.

Kok et al. (2010) propose an argumentation framework for deliberation dialogues that is based on persuasion dialogues as described in Prakken (2005a). Alternatives can be proposed and for each proposed alternative, a persuasion dialogue is started amongst the participants. During or after proposing actions, the status of proposals can be determined. A proposal can either be justifiable, defensible or invalid. After that, agents can express their preferences over proposals, which is used in a preference aggregation function to determine what joint action the participants should choose.

7.1.3 Relevance in Dialogues

To ensure that the participants of a dialogue only make utterances that relate to the dialogue's topic, several notions of relevance have been proposed in the literature. In Parsons et al. (2007) and Amgoud and de Saint-Cyr (2009), participants of a dialogue can put forward arguments. The dialogue starts with a participant putting forward an initial argument. Next, all participants can put forward arguments. From these utterances in a dialogue, a Dung argumentation framework can be constructed, which can be visualized in an argument graph. The argument graph that corresponds to a dialogue is then used to define different notions of relevance.

Parsons et al. (2007) distinguish between three notions of relevance. A move m is: *R1-relevant* if making m changes the status of the initial argument; *R2-relevant* if m puts forward an argument that is connected to the initial argument in the argument graph; and, *R3-relevant* if m puts forward an argument that is connected to the argument that is last put forward in the dialogue. Note that for these definitions of relevance it does not matter which participant utters an argument.

Amgoud and de Saint-Cyr (2009) propose definitions for when a move is relevant, useful and decisive. These definitions use the argument graph that corresponds to the dialogue. If argument A_1 is the initial argument and a move m puts forward argument A_2, then m is called: *relevant* iff there A_2 and A_1 are connected in the argument graph; *useful* iff there is a directed path from A_2 to A_1; and, *decisive* iff the status of A_1 in the dialogue including m is different from the status of A_1 in the dialogue excluding m. R1-relevance is the same as decisiveness and R2-relevance is the same as Amgoud and de Saint-Cyr (2009)'s notion of relevance. R3-relevance is not similar to Amgoud's notions because Amgoud's notions are defined with respect to the initial argument, whereas R3 is not defined on the initial argument. Note that neither Parsons nor Amgoud considers which agent made the move or uttered the initial argument. A move in which an agent attacks his own argument can thus be relevant, useful or decisive.

In the dialogue framework by Prakken (2005a), two notions of relevance are introduced: strong and weak relevance. Since Prakken's framework has more locutions than only advancing an argument, the definitions of relevance are somewhat more complicated. We will describe Prakken's notions of relevance in Subsection 7.2.3 and we will relate these defini-

tions to the definitions of Amgoud and Parsons.

7.2 Dialogue Framework

Prakken (2005a) proposes a formal dialogue framework that enables agents to argue with each other. It consists of a set of locutions, an explicit reply structure that imposes what responses are legal, and specifies when a dialogue move attacks or surrenders another dialogue move. Although this dialogue framework allows for different underlying logics, it is particularly suitable for argumentation logics like ASPIC+ (which we use in this thesis).

The dialogue framework by Prakken (2005a) has been used for different types of dialogue. For example, Prakken (2006) uses this dialogue framework for persuasion dialogues. Also, Kok et al. (2010) show how Prakken's framework can be used for deliberation dialogues. Because we use ASPIC+ and because we are interested in dialogues that are similar to deliberation dialogues, we will use Prakken's dialogue framework. In Subsection 7.2.1 we will first describe and extend Prakken's dialogue framework for the use of object-level and meta-level arguments. Then in Subsection 7.2.2, this dialogue framework is instantiated for decision support dialogues and in Subsection 7.2.3 a protocol for decision support is proposed.

7.2.1 Communication Language, Dialogue Moves, and Dialogues

In our Perspective-based Value Model (PVM), multiple argumentation systems are used to argue about what decision to make as described in Chapter 5 and 6. The argumentation systems that are proposed in these chapters are organized in a so-called 'tower of argumentation systems', i.e., an 'object-level' argumentation system, with on top of that a number of meta-argumentation systems. Because both object-level and meta-level arguments are important when making a decision, it is important that arguments on all levels of a tower of argumentation systems can be uttered in a dialogue.

We distinguish four kinds of *locutions* (also called utterances) that agents can make in a dialogue. If an agent α utters an argument A that is an argument of the argumentation system of level i, then we say that α *advances argument* A *on level* i. This locution is denoted by $\mathsf{advance}_i(A)$. An agent can also ask why a certain formula ϕ is true where ϕ is a formula of the language of the argumentation system on level i. We denote this by $\mathsf{why}_i(\phi)$. If an agent admits that formula ϕ (on level i) is true, then we say that the agent *concedes* ϕ, which is denoted with $\mathsf{concede}_i(\phi)$. Finally, if an agent wants to withdraw formula ϕ (on level i) as unjustified, then we say that the agent *retracts* ϕ, denoted by $\mathsf{retract}_i(\phi)$.

Definition 7.1 (Locutions) *Let* $\mathcal{T}_{\mathcal{AS}} = \{\mathcal{AS}_1, \ldots, \mathcal{AS}_n\}$ *be a tower of argumentation systems of level* n *with* $\mathcal{AS}_i = \langle \mathcal{L}_i, \mathcal{R}_i, \mathsf{cf} \rangle$ *for each* $\mathcal{AS}_i \in \mathcal{T}_{\mathcal{AS}}$*. The* locutions for tower of argumentation systems $\mathcal{T}_{\mathcal{AS}}$ *is a set* $L = L_1 \cup \ldots \cup L_n$ *such that for* $1 \leq i \leq n$*:*

- *for all* $A \in \mathsf{Args}(\mathcal{AS}_i)$*:* $\mathsf{advance}_i(A) \in L_i$
- *for all* $\phi \in \mathcal{L}_i$*:* $\mathsf{why}_i(\phi), \mathsf{concede}_i(\phi), \mathsf{retract}_i(\phi) \in L_i$

Using the advance locution, every argument on every level of a tower of ASs can be communicated. This means that arguments concerning the preferences between arguments can be

advanced. Note that Prakken (2005a) distinguishes between claiming a formula and justifying a formula with an argument. In contrast, this definition does not distinguish between this. Rather, if a participant wants to claim a formula, then he should advance an atomic argument concluding that formula.

Not every locution can be used in reply to another locution. For example, $\mathsf{why}_1(\phi)$ cannot be uttered in reply to $\mathsf{why}_2(\psi)$. A reply structure specifies what locutions can be uttered in reply to other locutions. Prakken distinguishes two kinds of replies: *attacking replies* and *surrendering replies*. For example, if locution l_1 advances argument A_1 and locution l_2 advances argument A_2, which attacks A_1, then l_2 attacks l_1. In contrast, if locution l_1 advances argument A and locution l_2 concedes A's conclusion, then l_2 surrenders to l_1. A reply structure is a tuple consisting of an attack and a surrendering relation between locutions and is defined formally as follows.

Definition 7.2 (Reply Structure) *Let L be a set of locutions. A* reply structure for L *is a tuple $\langle R_a, R_s\rangle$ with R_a and R_a irreflexive binary relations on L such that for all locutions $k, l, m \in L$:*

- *if $(k, l) \in R_a$, then $(k, m) \notin R_s$, and*
- *if $(k, l) \in R_s$, then $(m, k) \notin R_a$.*

The first constraint states that locutions that can be used as attacking replies can never be used as surrendering replies. The second constraints forbids that a surrendering locution is attacked.

Example 7.1 If locution k of advancing an argument in reply to the locution l of advancing an argument counts as an attack, then it must be that $(k, l) \in R_a$. This also means that k cannot be a surrendering reply to another move: $(k, m) \notin R_s$ for any m. In addition, if locution k of conceding in reply to locution l of advancing an argument counts as a surrender, then it must be so that $(k, l) \in R_s$. However, it is then not allowed that k also counts as an attack: for any m it is true that $(k, m) \notin R_a$.

A communication language is used in a dialogue by its participants to communicate to each other. To reason about what decision to make, the argumentation framework of Chapter 6 uses a tower of argumentation systems. To support a user in making a decision, it is therefore necessary to have a communication language which can be used to communicate the arguments in a tower of argumentation systems. Recall that locutions as specified in Definition 7.1 allow to do this. A communication language for a tower of argumentation systems is defined as follows.

Definition 7.3 (Communication Language) *Let $\mathcal{T}_{\mathcal{AS}}$ be a tower of argumentation systems. A* communication language $\mathcal{L}_{\mathcal{C}}$ for $\mathcal{T}_{\mathcal{AS}}$ *is a tuple $\langle L, R\rangle$ with L a set of locutions for $\mathcal{T}_{\mathcal{AS}}$ and R a reply structure on L.*

We consider the context in which a dialogue is held as consisting of a particular tower of argumentation systems, a communication language for this tower, and a set of agents that participate in the dialogue.

Definition 7.4 (Dialogue Context) *A* dialogue context *is a tuple $\langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \mathcal{P} \rangle$ such that $\mathcal{T}_{\mathcal{AS}}$ is a tower of argumentation systems of level n, $\mathcal{L}_{\mathcal{C}}$ is a communication language for $\mathcal{T}_{\mathcal{AS}}$, and $\mathcal{P} \subseteq \mathsf{Agents}$ the set of participating agents.*

When multiple agents engage in a dialogue, then we assume that they all agree on the dialogue context. This means that each participant can determine whether an argument is valid with respect to the argumentation system and whether arguments attack each other. Within a dialogue context, the participants can make dialogue moves in which they utter locutions from the communication language.

Definition 7.5 (Dialogue Move) *Let $\delta = \langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \mathcal{P} \rangle$ be a dialogue context. A* dialogue move in δ *is a tuple $\langle i, \alpha, l, j \rangle$ with $i, j \in \mathbb{N}$ such that $0 \leq j < i$, $\alpha \in \mathcal{P}$ the speaker, and $l \in \mathcal{L}_{\mathcal{C}}$ the utterance.*

If $m = \langle i, \alpha, l, j \rangle$ is a dialogue move, then (a) $\mathsf{id}(m) = i$ denotes the identifier of move m; (b) $\mathsf{pl}(m) = \alpha$ denotes the agent that made move m; (c) $\mathsf{loc}(m) = l$ denotes the locution of move m; and, (d) $\mathsf{target}(m) = j$ denotes the move's identifier at which m is targeted. We use $\mathsf{Mov}(\delta)$ to denote the set of all dialogue moves in dialogue context δ.

If m and m' are dialogue moves in a dialogue such that $\mathsf{id}(m') = \mathsf{target}(m)$, then we say that *move m replies to move m'*. Note that every move's identifier is at least 1 and that the target of a move m is always lower than m's identifier. If a move m's identifier is 1, then m must target 0. Since there is no move with identifier 0, m does not reply to any dialogue move. Consequently, every first dialogue move will have target 0. A move m is a *valid reply* to m' if and only if the locution of m is a reply to the locution of m' according to the reply structure of the communication language. Suppose that move m replies to move m'. If $\mathsf{loc}(m)$ is an attacking reply to $\mathsf{loc}(m')$, then we say that move m is an *attacking reply* to move m'. If $\mathsf{loc}(m)$ is a surrendering reply to $\mathsf{loc}(m')$, then we say that m is a *surrendering reply* to m'.

Example 7.2 Let $\delta = \langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \{\alpha, \beta\} \rangle$ be a dialogue context. We will now look at several dialogue moves.

- Dialogue move $m_1 = \langle 1, \alpha, \mathsf{advance}_1(\phi), 0 \rangle$ is uttered by α and advances the object-level atomic argument ϕ. The id of m_1 is 1 and it replies to 0, so it does not reply to any move.
- Move $m_2 = \langle 2, \beta, \mathsf{advance}_1(\neg\phi), 1 \rangle$ is uttered by β and puts forward the object-level atomic argument $\neg\phi$. Move m_2 replies to m_1 and attacks m_1 (assuming a typical reply structure).
- Finally, move $m_3 = \langle 3, \alpha, \beta, \mathsf{why}_1(\psi), 2 \rangle$ is uttered by α in reply to m_2 and asks why ψ is true. However, β did not claim ψ in m_2 and therefore m_3 is not a valid reply.

A sequence of dialogue moves is denoted as $[m_1, \ldots, m_n]$, where m_1 is the first move in the sequence and m_n the n-th and last move in the sequence. If $M = [m_1, \ldots, m_n]$ is a sequence and m is a dialogue move, then $M + m$ denotes the sequence $[m_1, \ldots, m_n, m]$. The set of all dialogue move sequences in dialogue context δ is denoted as $\mathsf{Seq}(\delta)$. Given the notions of a dialogue context and dialogue moves within that context, we will now define dialogues.

Definition 7.6 (Dialogue) *A* dialogue *is a tuple* $\langle \delta, M \rangle$ *such that* δ *is a dialogue context and* M *is a non-empty sequence* $[m_1, \ldots]$ *of dialogue moves in* δ *such that for each* $m_i \in M$*:*

- $\mathsf{id}(m_i) = i$*,*
- $\mathsf{target}(m_i) = 0$ *iff* $\mathsf{id}(m_i) = 1$*, and*
- $\mathsf{target}(m_i) \neq 0$ *iff* m_i *is a valid reply to* m_i*'s target.*

We will assume that there is a mechanism that can be used by agents to start a new dialogue with each other. Furthermore, we will assume that each participant of a dialogue can terminate the dialogue whenever he likes. In Definition 7.6, the first constraint $\mathsf{id}(m_i) = i$ ensures that every dialogue move in a dialogue has a unique identifier. The second constraint ensures that only the first dialogue move can have target 0. Finally, the third constraint ensures that every dialogue move that has a target, is a valid reply to that target.

Protocols regulate dialogues by specifying what sequences of dialogue moves are *legal*. A protocol is defined as a set of sequences of dialogue moves. Recall that $\mathsf{Seq}(\delta)$ denotes the set of all possible sequences of dialogue moves in context δ. It is convenient if it can easily be determined what dialogue moves are legal for agents to make given a sequence of moves that have already been uttered. To facilitate this, a so-called 'protocol function' is defined that returns the set of legal moves given a sequence of moves.

Definition 7.7 (Protocol) A protocol for dialogue context δ *is a set* $Pr \subseteq \mathsf{Seq}(\delta)$ *of sequences of dialogue moves in* δ*.* The protocol function for protocol Pr *is the function* $\pi : \mathsf{Seq}(\delta) \rightarrow 2^{\mathsf{Mov}(\delta)}$ *such that*

$$\pi(M) = \begin{cases} \mathsf{undefined} & \text{if } M \notin Pr \\ \{m \in \mathsf{Mov}(\delta) \mid M + m \in Pr\} & \text{otherwise} \end{cases}$$

When a sequence of moves is in Pr, then that sequence is called *legal w.r.t.* Pr. If $d = \langle \delta, M \rangle$ is a dialogue, Pr is a protocol and π the protocol function of Pr, then $\pi(M)$ is the set of all dialogue moves that are legal in d. If $\pi(M) = \emptyset$, then dialogue d is *terminated*. Note that enforcing that participants must take turns can be done with a protocol, i.e., if it is participant α's turn, then only dialogue moves by α are legal.

Often it is useful to regulate which participant can make a dialogue move at what time in the dialogue. To this end, Prakken introduces a turn-taking function that determines the participants that can make a move, i.e., whose turn it is, in a given dialogue. We will now define turn-taking functions similar to Prakken, but adapted to dialogue contexts. Recall that $\mathsf{Seq}(\delta)$ denotes the set of all sequences of dialogue moves.

Definition 7.8 (Turn-Taking Function) *Let* $\delta = \langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \mathcal{P} \rangle$ *be a dialogue context. A* turn-taking function for δ *is a function* $\mathsf{tt} : \mathsf{Seq}(\delta) \rightarrow 2^{\mathcal{P}}$.

An example turn-taking function Prakken gives for a dialogue between agents α and β is that α can make the first move (i.e., $\mathsf{tt}([]) = \{\alpha\}$), β the second (i.e., $\mathsf{tt}([m_1]) = \{\beta\}$), and every later move can be made by either of them.

7.2.2 Decision Support Dialogues

In this section we tailor the dialogue framework as described in Subsection 7.2.1 for decision support dialogues. We will first propose a communication language for decision support, then define dialogue contexts for decision support and finally define decision support dialogues. Then Subsection 7.2.3 proposes a protocol for decision support.

Recall from Definition 7.3 that a communication language is a tuple consisting of a set of locutions and a reply structure on those locutions. Locutions are defined w.r.t. a tower of argumentation systems. For decision support we will use a specific tower of argumentation systems $\{\mathcal{AS}_1, \mathcal{AS}_2, \ldots, \mathcal{AS}_n\}$ of height $n \geq 2$ where $\mathcal{AS}_1$ is an ASPPR and $\mathcal{AS}_2$ is a meta-ASPPR. We will call this tower a *decision support tower* for easier reference. The locutions in a communication language for decision support must be based on a decision support tower.

Table 7.1 contains the reply structure for attacking replies. The first column contains the locution to which is replied. The second column contains the locutions that attack the locution in the first column given that the preconditions in the third column are satisfied. Prakken's replies are extended with an extra attacking reply. Namely, in response of $\mathsf{advance}_i(A_1)$ it is possible to utter $\mathsf{advance}_{i+1}(B)$, where B is a meta-level argument w.r.t. A_1 that concludes $A_1 \prec A_2$ s.t. A_1 and A_2 preference-dependent[1] attack each other. To ensure that the dialogue is coherent, we will encode in the protocol that A_2 must have been uttered before. This is done later in this chapter.

Table 7.1: *Attacking Replies*

Locution	Attack	Precondition[a]
$\mathsf{advance}_i(A_1)$	$\mathsf{advance}_i(A_2)$	A_2 attacks A_1
	$\mathsf{why}_i(\phi)$	$\phi \in \mathsf{premises}(A_1)$
	$\mathsf{advance}_{i+1}(B)$	$\mathsf{conc}(B) = A_1 \prec A_2$ for some argument A_2 that preference-dependent attacks A_1
$\mathsf{why}_i(\phi)$	$\mathsf{advance}_i(A_1)$	$\mathsf{conc}(A_1) = \phi$

[a] Recall from Section 2.1.1 that conc and premisesare functions that return the conclusion and premises of an argument respectively.

Table 7.2 contains the reply structure w.r.t. surrendering replies and is the same as the surrendering replies defined by Prakken.

Table 7.2: *Surrendering Replies*

Locution	Surrender	Precondition
$\mathsf{advance}_i(A)$	$\mathsf{concede}_i(\phi)$	$\mathsf{conc}(A) = \phi$
	$\mathsf{concede}_i(\phi)$	$\phi \in \mathsf{premises}(A)$
$\mathsf{why}_i(\phi)$	$\mathsf{retract}_i(\phi)$	

[1] Recall from Subsection 2.1.3 that a preference-dependent attack refers to rebutting or undermining attacks. These kinds of attacks are called preference-dependent because their successfulness depends on the preferences over (or relative strengths of) arguments.

Given the set of locutions and the reply structure on them, we can now define communication languages for decision support.

Definition 7.9 (Communication Language for Decision Support) *A* communication language for decision support *is a communication language $\langle L, R\rangle$ with L the locutions for a tower of argumentation systems $\mathcal{T}_{\mathcal{AS}}$ and R the reply structure for L as described in Table 7.1 and Table 7.2.*

For decision support, we will restrict ourselves to dialogues between two agents, e.g., agents α and β, in which they argue about α's motivation concerning what decision α should make.

Definition 7.10 (Dialogue Context for Decision Support) *A* dialogue context for decision support *is a dialogue context $\langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \mathcal{P}\rangle$ with $\mathcal{T}_{\mathcal{AS}}$ a tower of argumentation systems for decision support, $\mathcal{L}_{\mathcal{C}}$ a communication language for $\mathcal{T}_{\mathcal{AS}}$ and $\mathcal{P} = \{\alpha, \beta\}$ two agents.*

We call a dialogue $\langle \delta, M\rangle$ a *decision support dialogue* if and only if δ is a dialogue context for decision support.

7.2.3 A Protocol for Decision Support

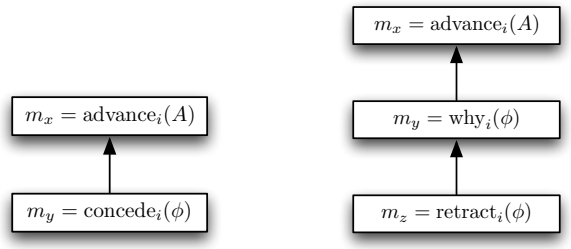
Protocols regulate what moves can be made in a dialogue. Recall from Definition 7.7 that a protocol Pr is a subset of all the possible sequences of dialogue moves and that a protocol function π is associated to a protocol that returns the set of legal dialogue moves given a dialogue. If $m \in \pi(d)$, then it is *legal* to make move m in dialogue m.

In this section we will propose a protocol that is tailored for decision support dialogues. First we will introduce the protocol rules that are proposed in Prakken (2005a). Not all of these protocol rules are suitable for decision support dialogue and we will adapt and add several. Next, different variants of when a dialogue move is *relevant* are defined in order to keep the dialogue coherent. For this, each move is assigned a *dialogical status*, which depends on the dialogical statuses of the moves that reply to that move. Relevancy is then defined based on whether a move changes or may change the status of other moves.

Prakken's Protocol Rules

Prakken (2005a) proposes the following rules that protocols should follow in order to ensure a minimal level of coherence. Let $d = \langle \delta, M\rangle$ be a dialogue with $\delta = \langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \{\alpha, \beta\}\rangle$.

- R_1: if m is legal in d, then it is true that $\mathsf{pl}(m)$ has the turn in d (i.e., $\mathsf{pl}(m) \in \mathsf{tt}(M)$)
- R_2: if m is legal in d and $\mathsf{target}(m) \neq 0$, then $\mathsf{loc}(m)$ is a valid reply according to $\mathcal{L}_{\mathcal{C}}$
- R_3: if m is legal and m replies to m', then $\mathsf{pl}(m) \neq \mathsf{pl}(m')$
- R_4: if m is legal, then there is no $m' \in M$ with $\mathsf{target}(m) = \mathsf{target}(m')$ and $\mathsf{loc}(m) = \mathsf{loc}(m')$.
- R_5: if m is legal, then for every $m' \in M$ that surrenders to the target of m it is true that m is not an attacking counterpart of m'



(a) *Concede* **(b)** *Retract*

Figure 7.1: *Concede and Retract Moves*

Rule R_3 encodes that a participant can never make a move that replies to a move that he made himself. Rule R_4 encodes that a move m is illegal when the same locution has already been uttered in reply to m's target. Note that this only forbids agents making the same locution in reply to a move, but does not forbid making the same locution in reply to a different move. For example, if move m_2 advances argument A in reply to move m_1, then a new move m_3 must not also advance A in reply to m_1. However, if m_3 targets a different move, then it can advance A.

Replying to Own Moves

If rule R_3 is not used and participants can thus reply to their own moves, then special attention has to be given to the concede and retract locutions. Otherwise, a participant can retract claims of others or concede his own claims. We will now introduce two protocol rules for this purpose. Namely, if a participant makes a move in which he concedes ϕ, then ϕ must have been uttered by another participant. Figure 7.1a visualizes that first move m_x is made in which argument A is put forward. Then move m_y is made in reply, which concedes either a premise or the conclusion of A. Because participants should only be able to concede claims of other participants, it must be the case that $\mathsf{pl}(m_x) \neq \mathsf{pl}(m_y)$. The following protocol rule captures this constraint.

- R_{cnc}: if m is legal and m is a replying move and $\mathsf{loc}(m) = \mathsf{concede}_i(\phi)$, then the target of m is not made by the player of m.

Similarly, we must constrain who can make retract moves. Recall from Table 7.1 and Table 7.2 that a retract locution can only be made in reply to a why locution and that a why locution can only be made in reply to an advance locution. Figure 7.1b visualizes that first move m_x is made in which argument A is advanced. Next, move m_y is made in which a premise ϕ of A is questioned and finally move m_z is made, which retracts the premise. Because a participant should only be able to retract claims that he has made and not claims of other participants, it must be the case that $\mathsf{pl}(m_z) = \mathsf{pl}(m_x)$. The following protocol rule captures this.

- R_{rtr}: if m_x is legal and m_x replies to m_y and m_y replies to m_z and if $\mathsf{loc}(m_x) = \mathsf{advance}_i(A)$, $\mathsf{loc}(m_y) = \mathsf{why}_i(\phi)$ and $\mathsf{loc}(m_z) = \mathsf{retract}_i(\phi)$, then $\mathsf{pl}(m_z) = \mathsf{pl}(m_x)$.

Dialogical Status of Moves

To determine the outcome of a dialogue, Prakken (2005a) considers two dialogical statuses of dialogue moves: *warranted* and *unwarranted* (he calls these 'in' and 'out'). The status of a move m is determined using the statuses of the moves that reply to m. For this, we first need to define when a move is surrendered. This definition is taken from Prakken.

Definition 7.11 (Surrendered Move) *Let $d = \langle \delta, M \rangle$ be a dialogue. Move $m \in M$ is* surrendered in d *iff*

- $\mathsf{loc}(m) = \mathsf{advance}_i(A)$ *and there is a move m' in M that replies to m and concedes A's conclusion; or else*
- *m has a surrendering reply in d.*

We can now define the dialogical status of a move in a dialogue.

Definition 7.12 (Dialogical Status) *Let $d = \langle \delta, M \rangle$ be a dialogue. The* dialogical status of $m \in M$ *is* warranted *if and only if*

- *m is surrendered in d, or*
- *all attacking replies to m are not warranted.*

If move m is not warranted, then m is unwarranted.

In a dialogue, a move does not necessarily change the status of the move it replies to. For example, if m attacks m', but m' was already unwarranted, then m does not change the status of m'. For the same reason, a new move could change the status of the initial move. Different notions of when a move is relevant can be defined based on whether the move changes or potentially changes the dialogical status of other moves.

To define various kinds of relevance, we will use the notion of a *winning part* of a move in a dialogue for an agent. Informally, a winning part of a dialogue for a move is a part of the dialogue that makes that move 'win', i.e., be warranted. The notion of a winning part is taken from Prakken, but generalized in two ways. Firstly, we define the winning part for any move in a dialogue instead of only for the initial dialogue move. Secondly, instead of defining the winning part only for moves that are warranted, they are defined for moves that are either warranted or unwarranted.

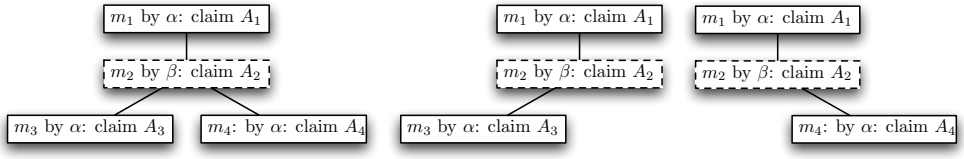
Recall that if a move is warranted, then it either is surrendered or all attackers are unwarranted. Consequently, if a warranted move is in a winning part and there are surrendering replies, then its surrendering replies make it 'win' and should thus be in the winning part. Moreover, if a warranted move is in a winning part and it has no surrendering replies, then all its attacking replies are successfully defeated and should thus be in the winning part. Finally, if a winning part contains an unwarranted move, then the winning part should include at least one warranted attacking reply in order to make the winning part win. Because we want to distinguish between winning parts, exactly one warranted attacking reply will be included in the winning part.

Definition 7.13 (Winning Part) *Let $d = \langle \delta, M \rangle$ be a dialogue where agent α is a participant.* A winning part for α of move m_i in d *is a set W of dialogue moves that is defined recursively:*

- *if* $\mathsf{pl}(m_i) = \alpha$ *and* m_i *is warranted in* d*, then* $m_i \in W$
- *if* $\mathsf{pl}(m_i) \neq \alpha$ *and* m_i *is unwarranted in* d*, then* $m_i \in W$
- *if* $m \in W$*,* m *is warranted and* m *is surrendered in* d*, then all surrendering replies of* m *in* d *are included in* W*,*
- *if* $m \in W$*,* m *is warranted and* m *is not surrendered in* d*, then all attacking replies of* m *in* d *are included in* W*,*
- *if* $m \in W$ *and* m *is unwarranted, then exactly one warranted dialogue move* $m' \in M$ *is included in* W *that attacks* m *in* d

Note that if m_i's status is warranted, but α is not the speaker of m_i, then there is no winning part of m_i for α because α does not win the argument concerning m_i. Similarly, if m_i's status is unwarranted and α is the speaker of m_i, then α has no winning part for m_i. Further note that this generalizes Prakken's definition of a winning part: if $d = \langle \delta, [m_1, \ldots, m_n] \rangle$ is a dialogue, then Prakken's notion of a winning part corresponds to looking at the winning part of m_1 for $\mathsf{pl}(m_1)$ in d.

Example 7.3 (Winning Part) Consider the dialogue $\langle \delta, [m_1, m_2, m_3, m_4] \rangle$ between participants α and β. In m_1, α claims argument A_1, which is attacked by β's move m_2. However, m_2 is attacked by both α's moves m_3 and m_4. The dialogue is visualized in Figure 7.2a. Because m_3 and m_4 are not attacked in d, they are both warranted. Then m_2 is unwarranted



(a) *Full Dialogue* **(b)** *Winning Part* W_1 **(c)** *Winning Part* W_2

Figure 7.2: *Dialogue and the winning parts for agent* α *of move* m_1 *in Example 7.3*

because m_2 has two attackers that are warranted. Finally, m_1 has no attackers that are warranted, so m_1 is warranted. There are two winning parts W_1 and W_2 in d for participant α of m_1. Winning part W_1 is $\{m_1, m_2, m_3\}$ and visualized in Figure 7.2b. Winning part W_2 is $\{m_1, m_2, m_4\}$ and visualized in Figure 7.2c.

Prakken (2005a) proves several properties for winning parts of m_1 for $\mathsf{pl}(m_1)$. Let $d = \langle \delta, [m_1, \ldots, m_n] \rangle$ be a dialogue between participants α and β such that m_1 is made by α. Then: (1) d contains a winning part of m_1 for α iff m_1 is warranted; (2) the leaves of any winning part of m_1 for α are either surrenders by β or attackers by α; and, (3) in any winning part of m_1 for α, all moves of α are warranted in d and all moves of β are unwarranted in d. We will now investigate similar properties of our generalized notion of winning part.

Proposition 7.1 *Let* $d = \langle \delta, [m_1, \ldots, m_n] \rangle$ *be a dialogue. Dialogue* d *contains a winning part of* m_i *for* $\mathsf{pl}(m_i)$ *iff* m_i *is warranted.*

Proof From left to right: if d has a winning part of m_i for $\mathsf{pl}(m_i)$, then by Definition 7.13 it is true that m_i is warranted (since α in this case is $\mathsf{pl}(m_i)$ and m_i is in the winning part, the first bullet states that m_i must be warranted). From right to left: if m_i is warranted, then by definition it is true that m_i is in a winning part of m_i for $\mathsf{pl}(m_i)$. ■

Proposition 7.2 *Let $d = \langle \delta, [m_1, \ldots, m_n] \rangle$ be a dialogue between agents α and β. For every move m_i in d there is a winning part of m_i for either α or β.*

Proof Recall that every move m_i in a dialogue is either warranted or unwarranted by definition. Furthermore, if m_i is warranted, then by definition it is true that m_i is in the winning part of m_i for $\mathsf{pl}(m_i)$. Otherwise m_i is unwarranted and m_i is thus in the winning part of m_i for the player who did not make m_i. ■

Prakken is interested in dialogues where participants cannot reply to their own moves, i.e., dialogues satisfying protocol rule R_3. In such a dialogue, the replies to a warranted move m are either unwarranted attackers or surrenders made by the other participant. Consequently, the following proposition holds.

Proposition 7.3 *In a dialogue satisfying protocol rule R_3, if W is a winning part of m_i for $\mathsf{pl}(m_i)$, then every move in W by $\mathsf{pl}(m_i)$ is warranted and every attacking move of the other player is unwarranted.*

Proof Proof by induction: first we will show that all replies to m_i are made by the other player and unwarranted. For every move m in W that attacks m_i and is made by the other player: if m is warranted, then m_i must be unwarranted. Since m_i is warranted, m cannot be warranted. The induction step distinguishes between moves by $\mathsf{pl}(m_i)$ and the other player. If $m \in W$ is made by the other player and is unwarranted, then there must be a warranted attacking reply to m. By definition, W contains exactly one warranted attacking reply of $\mathsf{pl}(m_i)$. In contrast, if $m \in W$ is warranted, made by $\mathsf{pl}(m_i)$ and $m' \in W$ attacks m, then m' is made by the other player and must be unwarranted. Summarizing, all moves attacking m_i are made by the other player and are unwarranted. Furthermore, if a move of the other player is unwarranted, then there is a reply of $\mathsf{pl}(m_i)$ that is warranted. Finally, every reply to a warranted move of $\mathsf{pl}(m_i)$ is made by the other player and unwarranted. ■

The dialogue trees of dialogues satisfying R_3 have a specific structure because the player of moves interleave. The following property holds for the leaf nodes of winning parts.

Proposition 7.4 *In a dialogue satisfying protocol rule R_3, the leaves of any winning part of m_i for $\mathsf{pl}(m_i)$ are either surrenders by the other player or attackers by $\mathsf{pl}(m_i)$.*

Proof Because Proposition 7.3 holds and leaf nodes are warranted by definition, an attacking leaf node must be made by $\mathsf{pl}(m_i)$. A move of the other player can also be a leaf node when it surrenders to a move of $\mathsf{pl}(m_i)$. A surrendering leaf node m of $\mathsf{pl}(m_i)$ that replies to m' makes the move m' warranted. Because Proposition 7.3 holds and move m' is made by the other player, this leads to a contradiction. ■

For dialogues that do not satisfy R_3 Proposition 7.4 does not hold. Namely, in a dialogue satisfying protocol rules R_{cnc} and R_{rtr}, the leaves of a winning part of m_i for $\mathsf{pl}(m_i)$ can be made by any player. For example, consider dialogue $d = \langle \delta, [m_1, m_2, m_3, m_4] \rangle$ where: (1) α makes m_1 in which argument A_1 is advanced; (2) α then makes m_2 in which A_1 is attacked with argument A_2; (3) β makes m_3 attacking A_2 with A_3; and, finally (4) α concedes A_3's conclusion in m_4. Note that dialogue d satisfies R_{cnc} and R_{rtr}. Because m_3 is surrendered in m_4, move m_3 is warranted. This makes m_2 unwarranted and m_1 warranted. Consequently, the winning part of m_1 for α is $\{m_1, m_2, m_3, m_4\}$. However, note that m_4 is a leaf node and a surrendering move of α. This means that in dialogues satisfying R_{cnc} and R_{rtr}, it is possible that winning parts of a move m have surrendering leave nodes made by $\mathsf{pl}(m)$. Moreover, winning parts for m_i in dialogues satisfying R_{cnc} and R_{rtr} can also contain attacking leaf nodes of the other player. For example, consider dialogue $d = \langle \delta, [m_1, m_2, m_3] \rangle$ where: (1) α makes m_1 advancing argument A_1; (2) α then makes m_2 attacking m_1 by advancing argument A_2; and, finally (3) β makes m_3 attacking m_2 by advancing argument A_3. Because m_3 has no replies, m_3 is warranted making m_2 unwarranted and m_1 warranted. Consequently, the winning part of m_1 for α is $\{m_1, m_2, m_3\}$ in which m_3 is a leaf node that is an attacker and made by the other player.

Relevance

Using the notion of winning part, we distinguish the following levels of relevance. First, strongly and weakly relevant are taken from Prakken and are concerned with the status of the first move in the dialogue. An attacking move is *strongly relevant* if it changes the status of the first move and *weakly relevant* if it potentially changes the status of the first move, i.e., if it creates a new winning or removes a winning part for the first move. For example, if the first move m_1 is played by α and m_1 is already warranted, then attacking some move of β does not have any effect on the status of m_1, so it is not strongly relevant. However, it does create a new winning part for m_1 because it provides a new way to attack β's moves, so it is weakly relevant.

The generalized notion of winning part allows to define two other notions of relevance, which we will call strongly related and weakly related. Rather than focusing on the status of the first move, these relevancy notions are concerned with the winning parts of any moves. *Strongly related* is defined as creating a new winning part for a move of the speaker himself or removing a winning part of the other agent for any move. *Weakly related* is defined as creating or removing a winning part of any move of any player. Formally, these notions are defined as follows.

Definition 7.14 (Relevance) *Let $d = \langle \delta, [m_1, \ldots, m_n] \rangle$ be a dialogue between agents α and β. Move m is (with $d' = \langle \delta, [m_1, \ldots, m_n, m] \rangle$)*

- strongly relevant in d *iff* *m is an attacking move and the status of m_1 in d' is different than in d or m is a surrendering move and m's attacking counterparts are strongly relevant*
- weakly relevant in d *iff*
  - *m is made by α and creates a new winning part for α of m_1, or*
  - *m is made by β and removes a winning part for α of m_1*

- strongly related to d *iff there is a move* m_i *in* d *such that*
  - m *creates a new winning part for* $\mathsf{pl}(m)$ *of* m_i*, or*
  - m *removes a winning part for the other agent of* m_i
- weakly related to d *iff there is a move* m_i *in* d *such that*
  - m *creates a new winning part for any player of* m_i*, or*
  - m *removes a winning part for any player of* m_i*.*

Every strongly relevant move is also weakly relevant, every weakly relevant move is also strongly related and every strongly related move is also weakly related. In Example 7.3, all moves are related and weakly relevant, but only moves m_1, m_2 and m_3 are strongly relevant. Note that move m_4 is not strongly but weakly relevant because m_4 does not change the status of m_1.

Example 7.4 (Relevancy of Moves) First consider the dialogue $d_1 = \langle \delta, [m_1] \rangle$ between agent α and β s.t. m_1 is made by α. In d_1, m_1 is warranted and $\{m_1\}$ is the only winning part of m_1 for α. Note that there is no winning part for β. Move m_2 of β attacks m_1. In $d_2 = \langle \delta, [m_1, m_2] \rangle$, m_2 is warranted and m_1 is unwarranted. Consequently, m_2 changes the status of the initial move and is therefore strongly relevant. In d_2 there is no winning part of m_1 for α, but $\{m_1, m_2\}$ is a winning part of m_1 for β.

Now consider move m_3 by β that also attacks m_1. In $d_3 = \langle \delta, [m_1, m_2, m_3] \rangle$, move m_1 is also unwarranted. Because m_3 does not change m_1's status, move m_3 is not strongly relevant. Moreover, because α did not have any winning parts in d_2, move m_3 did not remove any winning parts and is therefore also not weakly relevant. However, m_3 is strongly related because it creates a new winning part for β for move m_1.

Finally consider move m_4 by α that attacks m_1. In $d_4 = \langle \delta, [m_1, m_2, m_3, m_4] \rangle$, move m_1 is still unwarranted. Note that α is attacking his own move here. For similar reasons, move m_4 is thus not strongly nor weakly relevant. Because m_4 does not create a winning part for α or remove a winning part for β, move m_4 is not strongly related. However, because m_4 creates a new winning part for β, move m_4 is weakly related.

Example 7.5 The following dialogue between agents α and β illustrates the notions of relevance.

- Move m_1 by α advances argument $A_1 = \frac{\phi_1}{\phi}$.
- In reply to m_1, move m_2 by β asks why ϕ_1 is true. Move m_2 is warranted and makes m_1 unwarranted. Therefore, m_2 is strongly relevant. Because m_1 is unwarranted, there is no winning part for α of m_1, but $\{m_1, m_2\}$ is a winning part for β of m_1.
- Move m_3 by α advances argument $A_2 = \frac{\phi_2}{\neg\phi_1}$ in reply to m_1. Agent α's move m_3 attacks his own move m_1. Move m_3 has no attackers and is therefore warranted. Move m_1 is still unwarranted, so there is no winning part for α of m_1. However, there are now two winning parts $\{m_1, m_2\}$ and $\{m_1, m_3\}$ for agent β of move m_1. Also note that $\{m_3\}$ is a winning part for α of move m_3. Move m_3 does not change the status of m_1 and is therefore not strongly relevant. Because m_3 does not create a new winning part for α of m_1, move m_3 is neither weakly relevant nor strongly related. However, m_3 does create a new winning part for β of m_1 and is therefore weakly related.

- Move m_4 by α advances argument $A_3 = \frac{\phi_3}{\neg\phi_2}$ in reply to move m_3. Move m_4 attacks m_3. Because m_4 is warranted, m_3 is unwarranted. Move m_4 does not change m_1's status nor does it create a new winning part for α of m_1 and is therefore neither strongly or weakly relevant. However, m_4 does remove the winning part $\{m_1, m_3\}$ for agent β of move m_1. Consequently, m_4 is strongly related.

Recall from Subsection 7.1.3 that in some dialogue systems in the literature it is only possible to put forward arguments and not ask why-questions or concede or retract statements. Because these locutions are possible in our dialogue framework, the different notions of relevance as proposed by Parsons et al. (2007) and Amgoud and de Saint-Cyr (2009) cannot be directly translated to our dialogue framework. However, we can investigate how they correspond when we only consider moves for putting forward arguments. If a move m puts forward an argument A_1 in Prakken's framework, then either (1) m has target 0 (i.e., it starts a new topic), (2) m's target puts forward argument A_2 and A_1 attacks A_2, or (3) m's target is a move asking why ϕ and A_1 justifies why ϕ is true. To investigate the correspondence, we only need to look at the second case because the initial argument is always relevant and why questions cannot be posed in the other frameworks.

Prakken's notion of relevance was introduced in Prakken (2000) and further distinguished into strong and weak relevance in Prakken (2005a). Strong relevance corresponds to Amgoud's notion of decisiveness and to Parsons' notion of R1-relevance because they all concern changing the status of the initial argument. Because our notion of weakly related does not depend on the speaker of moves and it does not require that the initial argument's status changes, weakly related corresponds to Amgoud's notion of usefulness. Because weakly related is not defined with respect to the initial argument, weakly related also corresponds to both Parsons' notions of R2- and R3-relevance. Amgoud's notion of relevance is similar to usefulness, but differs in the following way: the direction of the path in which an argument is connected to the initial argument does matter for being useful, but does not matter for being relevant. This means that if there is no path from argument A to the initial argument, but the initial argument does attack A, then A is relevant but not useful. Because our reply structure does not allow putting forward argument A, Amgoud's notion of relevance does not correspond to any of our notions of relevance.

Protocol For Decision Support

If it is desirable in a dialogue that an agent can attack his own dialogue moves, then a protocol should be used that allows making weakly related dialogue moves. We think this is desirable for the decision support dialogues that we have in mind. Complex decisions involve many aspects that a decision maker has to consider. It is therefore possible that the decision maker makes a mistake, e.g., by forgetting an aspect or not using the right criteria. If a decision maker realizes that he made a mistake and can attack his own moves, then he can clarify why he changed his mind. Assume that the user put forward argument A justifying why he has a certain achievement goal. After receiving new information the user realizes that the criterion he used for that goal was not the right criterion. To conform this with the decision support system, the user could attack A by advancing another argument A' concluding that it is not the right criterion. By doing so, the user can check with the decision support system whether A' is indeed a justified argument to conclude that A is overruled. Prakken's protocol rule R_3

enforces that participants cannot reply to their own moves. To allow participants to attack their own moves, rule R_3 should thus not be used.

Summarizing, a protocol for decision support dialogues should allow making weakly related dialogue moves as this gives the participants the opportunity to attack their own moves. To enforce that moves must be weakly related, we introduce the following protocol rule that is called R_{wr}:

- R_{wr}: if m is legal and m replies to another move, then m is weakly related in d.

We are now ready to propose a protocol specifically for decision support. Prakken's protocol rule R_3 states that if a move m is legal, then m and m's target are not made by the same participant. If R_3 is used, then an agent can never attack his own moves, which we want to allow. Therefore, R_3 will not be used in the protocol for decision support. Because decision support is a collaborative rather than competitive setting, we will not enforce any rules concerning turn-taking. Therefore, R_1 will not be included.

We will use Prakken's protocol rules R_2, R_4, and R_5 because we do want that all moves are valid replies, that no move can be repeated, and that surrenders cannot be 'revoked' respectively. Furthermore, we will use protocol rule R_{wr} to enforce that moves must be weakly related.

Definition 7.15 (Protocol for Decision Support) *A* protocol for decision support *is a protocol Pr that satisfies R_2, R_4, R_5, R_{wr}, R_{cnc}, and R_{rtr}.*

A protocol for decision support does not guarantee that every dialogue will terminate at some point. Because participants can terminate a dialogue whenever they like (we make this assumption after Definition 7.6), we do not think it is a problem that termination is not guaranteed. Namely, if the dialogue is still considered to be useful by its participants, then they can continue. If the dialogue is considered useless by one of its participants, then he can terminate the dialogue.

Section 7.4 demonstrates the protocol for decision support on the running example from the introduction, but here we will first give a small example to demonstrate the protocol.

Example 7.6 (Decision Support Dialogues) Let $d = \langle \delta, [m_1] \rangle$ be a dialogue between α and β. In move m_1, agent α claims argument A_1, which concludes that agent α should perform alternative a because a certainly achieves an achievement goal of α w.r.t. perspective p.

$$A_1 = \frac{\mathsf{goal}(\alpha, p, G_p) \qquad \mathsf{alwAch}(a, G_p)}{\mathsf{do}(\alpha, a)}\ d_{alwG}$$

Agent β wants to respond and has the following two arguments that attack A_1. Both arguments conclude that α should not perform alternative a because a certainly fails to avoid agent α's avoidance goals w.r.t. perspectives q and r.

$$A_2 = \frac{\mathsf{avoid}(\alpha, q, G_q)\ \ \mathsf{alwAch}(a, G_q)}{\neg\mathsf{do}(\alpha, a)}\ d_{alwR} \quad A_3 = \frac{\mathsf{avoid}(\alpha, r, G_r)\ \ \mathsf{alwAch}(a, G_r)}{\neg\mathsf{do}(\alpha, a)}\ d_{alwR}$$

Agent β could either make move m_2 claiming argument A_2 or move m_3 claiming A_3. Making move m_2 is strongly relevant because it changes the status of m_1 to unwarranted. If the

protocol only allows making strongly relevant moves, then making move m_3 after making move m_2 is not legal because m_3 does not change the status of move m_1. Similarly, if the protocol only allows weakly relevant moves, then making move m_3 after m_2 also is not legal. However, if the protocol satisfies rule R_{wr}, then making move m_3 after m_2 is legal because it is strongly related. Note that move m_3 is strongly related because it creates a new winning part for β of m_1.

7.3 Revising Beliefs

Communicating is exchanging information, but if agents do not do anything with the information they get, communication is useless. The dialogue moves that agents make are valuable sources of information because they may provide information about the world and about what the speaker believes and wants. For example, if agent α claims that it currently rains in Amsterdam, then you could learn that it currently rains in Amsterdam, but you could also learn that α believes it currently rains and that α wants you to believe so as well.

The process of how the beliefs of an agent should be revised given new information is called *belief revision* and is studied in philosophy, logic, computer science and artificial intelligence. Falappa et al. (2009) give an overview of approaches of belief revision and describe the relationship between belief revision and argumentation. Some approaches focus on argumentation frameworks where arguments are abstract and in other approaches arguments can have structure. For example, Cayrol et al. (2008) investigate the revision of argumentation frameworks by studying the impact of adding a single new argument to an argumentation framework. In contrast, Paglieri and Castelfranchi (2005) propose a two-step process for argumentation-based belief revision called 'data-oriented belief revision'. An agent's beliefs are revised if he has been persuaded by arguments. A distinction is made between data and beliefs. Data are pieces of information that an agent has gathered and stored, whereas beliefs are statements about the world that the agent considers to be truthful (possibly to a certain degree). The set of data that an agent has can be inconsistent, but the beliefs of an agent cannot be. Belief revision is done in two steps: (1) determine the effect of the new data on other data; and, (2) select what to believe. For example, if agent α observes agent β claiming that it rains in Amsterdam, then α adds the datum 'it rains in Amsterdam'. However, α may also have the datum 'it does not rain in Amsterdam', which was claimed by agent γ before. Now α has to select what to believe. Because α trusts β more than γ, agent α selects to believe that it rains in Amsterdam.

In our framework, an agent has a knowledge base consisting of a set of necessary premises, a set of ordinary premises, and a set of assumptions (see Definition 2.6). The set of ordinary premises and the set of assumptions can both be inconsistent. From an agent's knowledge base, arguments are constructed. Meta-level argumentation is then used to reason about the relative strength of object-level arguments. The arguments are then evaluated using the successful attacks between them resulting in a set of justified, defensible and overruled conclusions. When comparing our framework to data-oriented belief revision, the agent's ordinary premises and assumptions can be seen as what is called data, and the justified conclusions as the agent's beliefs. The process of constructing the arguments from a knowledge base and evaluating them can be seen as what Paglieri and Castelfranchi (2005) call belief selection.

We will now define knowledge update functions, which update the knowledge of an agent

given a new observed dialogue move.

Definition 7.16 (Knowledge Update Function) *Let $\mathcal{K}$ be the set of all possible knowledge bases that an agent could have. A* knowledge update function *is a function* update : $\mathcal{K} \times M \rightarrow \mathcal{K}$.

There are many ways of how to update an agent's knowledge after observing a dialogue move of another agent. Some ways update the agent's knowledge if another agent claims that ϕ is true: add ϕ as an ordinary premise, add ϕ as an assumption, or add ϕ as an ordinary premise under certain conditions (e.g., whether the other agent is trusted) and as an assumption otherwise.

To illustrate knowledge update functions, we will now informally introduce the *assumption knowledge update function*, which adds claims as assumptions. Recall from Definitions 2.9 and 2.16 that every undermining attack on an assumption is successful. By assuming that claims of others are true, an agent does learn from the utterances of others but is skeptical to them because every attack on an assumption wins. This update function must make sure that the sets of necessary premises, ordinary premises and assumptions stay disjoint. For example, if agent α observes dialogue move m, which advances argument A, then $\mathcal{K}_{\mathsf{as}}$ should be updated to $\mathcal{K}_{\mathsf{as}} \cup \mathsf{premises}(A) \setminus (\mathcal{K}_{\mathsf{np}} \cup \mathcal{K}_{\mathsf{op}})$. Furthermore, suppose that agent β advances argument A_1 and that premise ϕ of A_1 was not in α's knowledge base. Then α adds ϕ to his assumptions. Next, α asks β why ϕ is true, to which β responds by advancing argument A_2. In that case, α should remove the assumption of ϕ and add possible new assumptions arising from A_2. It becomes more complicated when agents retract claims. Suppose that agent β first claims that ϕ is true and α adds ϕ as an assumption, but later β retracts this claim. Should α remove ϕ as an assumption (taking into account that other agents may also have claimed ϕ) or is there still enough reason for α to assume that ϕ is true? We will now illustrate the behavior of the assumption knowledge update function in an example dialogue.

Example 7.7 (Observing Dialogue Moves) Let $\mathcal{AS}$ be an argumentation system and let agent α's knowledge bases w.r.t. $\mathcal{AS}$ be $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ where $\mathcal{K}_{\mathsf{op}}$ and $\mathcal{K}_{\mathsf{as}}$ are empty, i.e., α knows nothing. Agent α is in a dialogue with agent β and β makes dialogue move m_1 in which he puts forward argument $A_1 \in \mathsf{Args}(\mathcal{AS})$:

$$A_1 = \frac{\phi_1 \qquad \phi_2}{\phi}\ d_1$$

Because α cannot construct any arguments that conclude the premises of A_1, all premises of A_1 will be added as assumptions. Given m_1, the assumption knowledge update function updates α's assumptions $\mathcal{K}_{\mathsf{as}}$ to:

$$\mathcal{K}_{\mathsf{as}} := \{\phi_1, \phi_2\}$$

From the updated knowledge base, agent α can now construct argument A_1. The dialogue with β ends and agent α starts a dialogue with agent γ where γ makes dialogue move m_2 putting forward argument A_2:

$$A_2 = \frac{\neg\phi_2 \qquad \phi_3}{\neg\phi}\ d_2$$

After observing γ's dialogue move, the assumption knowledge update function updates agent α's assumptions as follows:

$$\mathcal{K}_{\mathsf{as}} := \{\phi_1, \phi_2, \neg\phi_2, \phi_3\}$$

Because nothing is removed from α's knowledge bases, argument A_1 can still be constructed. However, because α has new knowledge, he can now also construct argument A_2. Note that arguments A_1 and A_2 have conflicting conclusions and therefore attack each other.

The knowledge update function could also update the meta-level knowledge base of the agent with the information of who made what claim. This information can then be used to reason on a meta-level about the relative strength of arguments. For example, assume that object-level argument A_1 is based on a claim of agent α and object-level argument A_2 on a claim of agent β. If agent α is more trustworthy than agent β, then that is a reason to believe A_1 is a stronger argument than A_2. However, if β is an expert w.r.t. that claim and α is not, then you may conclude that A_2 is stronger. Hunter (2008) uses meta-level argumentation to reason about whether a proponent of an argument is appropriate for that particular argument. The knowledge update function could update the meta-level knowledge base such that Hunter's formalization of appropriateness can be applied. However, to investigate comprehensively what the best way is use information about the proponent of claims and arguments to determine the relative strength of arguments is outside the scope of this thesis and recommended for future work.

7.4 Running Example

In this section we will demonstrate how the dialogue framework of this chapter can be used in the running example from Chapter 1, where a fire commander student is confronted with the situation where a factory is on fire and there a number of victims inside the building. The student has to decide what course of action to take. The decision support agent and the student start a decision support dialogue to argue about what alternative the student should decide to choose. In this dialogue they use the protocol for decision support as specified in Definition 7.15. The dialogue context in this example is $\delta = \langle \mathcal{T}_{\mathcal{AS}}, \mathcal{L}_{\mathcal{C}}, \{\alpha, \beta\}\rangle$, where

- $\mathcal{T}_{\mathcal{AS}}$ is $\{\mathcal{AS}_{\mathsf{ppr}}, \mathcal{AS}'_{\mathsf{pv}}\}$ with $\mathcal{AS}_{\mathsf{ppr}}$ and $\mathcal{AS}'_{\mathsf{pv}}$ as in Section 6.4,
- $\mathcal{L}_{\mathcal{C}}$ is a communication language based on $\mathcal{T}_{\mathcal{AS}}$, and
- α denotes the student and β denotes the decision support agent.

We assume that both α and β know how all relevant assignments compare from all criterion perspectives and in what assignments the alternatives result. However, the knowledge of α and β differs. Table 7.3 describes what knowledge the student and the decision support agents have concerning the student's value tree, outcomes of alternatives, evaluation of assignments and the relative importance between perspectives. Note that the student does not know that criterion perspective c_5 influences his objective o_2 nor that c_5 is a more important criterion for objective o_2 than criterion c_1. In addition, the support agent does not know that the student cares about objective o_2.

We will now look into an example dialogue that the decision support agent and the student could have. We will also show how the knowledge base of the support agent can be updated with the assumption knowledge update function that was described informally in the end of the previous section. Step by step, each move will now be described.

Table 7.3: *How the knowledge of the student and the decision support agent differ*

	Student α	**Decision support agent** β
Value tree	$v_2 \uparrow \alpha$, $o_2 \uparrow v_2$, $c_1 \downarrow o_2$, $\mathsf{cntxt}(\alpha, v_2)$, $\mathsf{cntxt}(\alpha, o_2)$, $\mathsf{cntxt}(\alpha, c_1)$	$c_5 \downarrow o_2, v_3 \uparrow \alpha, o_3 \uparrow v_3, c_6 \downarrow o_3, \mathsf{cntxt}(\alpha, c_5)$
Evaluation	$\mathsf{good}(\alpha, c_1, \{(x_1, 10)\})$	$\mathsf{good}(\alpha, c_5, \{(x_5, 10)\})$
Goals	$\mathsf{goal}(\alpha, c_1, \{(x_1, 10)\})$	
Outcomes	$\mathsf{results}(\mathsf{alt}_{\mathsf{re}}, s_{\mathsf{re}}), \mathsf{results}(\mathsf{alt}_{\mathsf{ere}}, s_{\mathsf{ere}})$	$\mathsf{results}(\mathsf{alt}_{\mathsf{re}}, s_{\mathsf{re}}), \mathsf{results}(\mathsf{alt}_{\mathsf{ere}}, s_{\mathsf{ere}})$
Importance		$c_1 \lhd_{o_2} c_5$

Move 1

The student starts the dialogue with move $m_1 = \langle 1, \alpha, \mathsf{advance}_1(A_1), 0\rangle$. In m_1, the student α advances the argument A_1 concluding that he should send firefighters in right away to rescue the victims (alternative $\mathsf{alt}_{\mathsf{re}}$) because it achieves his goal that the victims are inside for ten minutes. Formally, A_1 is as follows:

$$A_1 = \frac{\mathsf{goal}(\alpha, c_1, \{(x_1, 10)\}) \qquad c_1 \downarrow \alpha \qquad \mathsf{alwAch}(\mathsf{alt}_{\mathsf{re}}, \{(x_1, 10)\})}{\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{re}})}\, d_{alwG}$$

From β's knowledge base it is not possible to construct arguments that conclude A_1's premises $\mathsf{goal}(\alpha, c_1, \{(x_1, 10)\})$ and $c_1 \downarrow \alpha$. Using the assumption knowledge update function, agent β's knowledge base is updated as follows because of move m_1.

$$\mathcal{K}_{\mathsf{as}} := \{\mathsf{goal}(\alpha, c_1, \{(x_1, 10)\}), c_1 \downarrow \alpha\}$$

Note that the updated knowledge base allows β to construct A_1 and to determine the acceptability of A_1.

Move 2

Agent β replies to m_1 with move $m_2 = \langle 2, \beta, \mathsf{why}_1(c_1 \downarrow \alpha), 1\rangle$ in which β asks the student why the criterion perspective c_1 negatively influences his preferences. Because β has a limited understanding of α's value tree, asking why α cares about c_1 will give β valuable information that he can use to better support α in making a decision. Move m_2 does not change the object and meta-level knowledge bases of β.

Move 3

Next, the student α answers β's question with move $m_3 = \langle 3, \alpha, \mathsf{advance}_1(A_2), 2\rangle$. In m_3, α advances the argument explaining that $c_1 \downarrow \alpha$ is true because α has the objective of to maximize the safety of the victims, which promotes his value of security. Recall from Table 5.3 that axiom $\mathtt{ax}_5$ states $p \uparrow q \wedge q \uparrow r \supset p \uparrow r$, i.e., if perspective p positively influences perspective q, and q positively influences perspective r, then p positively influences r. Similarly, $\mathtt{ax}_7$

states $p{\downarrow}q \wedge q{\uparrow}r \supset p{\downarrow}r$. Using $\mathtt{ax}_5$ and $\mathtt{ax}_7$, argument A_2 is as follows:

$$A_2 = \dfrac{\dfrac{c_1{\downarrow}o_2 \quad \dfrac{\dfrac{o_2{\uparrow}v_2 \quad v_2{\uparrow}\alpha}{o_2{\uparrow}v_2 \wedge v_2{\uparrow}\alpha}\wedge I \quad \mathtt{ax}_5}{o_2{\uparrow}\alpha}\mathsf{MP}}{c_1{\downarrow}o_2 \wedge o_2{\uparrow}\alpha}\wedge I \quad \mathtt{ax}_7}{c_1{\downarrow}\alpha}\mathsf{MP}$$

After receiving m_3, β adds A_2's premises as assumptions and removes the assumption $c_1{\downarrow}\alpha$ because it can be explained with the premises of A_2. Consequently, β's object-level assumptions are updated to:

$$\mathcal{K}_{\mathsf{as}} := \{\mathsf{goal}(\alpha, c_1, \{(x_1, 10)\}), v_2{\uparrow}\alpha, o_2{\uparrow}v_2, c_1{\downarrow}o_2\}$$

Argument A_2 can now be constructed from β's knowledge base and will thus be in the corresponding argumentation framework. By evaluating this argumentation framework, β can determine whether A_2 is justified, defensible or overruled.

Move 4

Because of α's last move, β now knows that α cares about objective o_2. This allows β to construct argument A_3 that concludes that the criterion perspective c_5 negatively influences α's perspective because c_5 negatively influences o_2.

$$A_3 = \dfrac{\dfrac{c_5{\downarrow}o_2 \quad o_2{\uparrow}\alpha}{c_5{\downarrow}o_2 \wedge o_2{\uparrow}\alpha}\wedge I \quad \mathtt{ax}_7}{c_5{\downarrow}\alpha}\mathsf{MP}$$

Moreover, because β's knowledge base contains that an x_5 value of 10 is good from the criterion perspective c_5, agent β can construct the argument A_3' that concludes that α should have the achievement goal from c_5 to achieve an x_5 value of 10. Recall from Table 5.1 that $\mathsf{infl}(p, q)$ abbreviates $p{\uparrow}q \vee p{\downarrow}q$.

$$A_3' = \dfrac{\dfrac{A_3}{\mathsf{infl}(c_5, \alpha)}\vee I \quad \mathsf{cntxt}(\alpha, c_5) \quad \mathsf{good}(\alpha, \{(x_5, 10)\}, c_5)}{\mathsf{goal}(\alpha, c_5, \{(x_5, 10)\})}\, d_{\mathsf{goal}}$$

Now β makes move $m_4 = \langle 4, \beta, \mathsf{advance}_1(A_4), 1\rangle$ in reply to the first move of α. In m_4, β advances the argument that α should extinguish the fire partly, then rescue the victims and then extinguish the rest (alternative $\mathsf{alt}_{\mathsf{ere}}$) because c_5 is important to α and $\mathsf{alt}_{\mathsf{ere}}$ achieves α's goal w.r.t. criterion c_5.

$$A_4 = \dfrac{A_3' \qquad A_3 \qquad \mathsf{alwAch}(\mathsf{alt}_{\mathsf{ere}}, \{(x_5, 10)\})}{\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{ere}})}\, d_{alwG}$$

Move 5

The student does not agree that argument A_4 is justified and makes move $m_5 = \langle 5, \alpha, \mathsf{advance}_1(A_1), 4\rangle$ in which m_4 is attacked by advancing argument A_1. Note that α already

advanced A_1 in move m_1, but that the protocol allows to repeat an argument if it does not reply to the same move. Because A_1 was already put forward, β's knowledge base is not updated.

Move 6

Because the decision support agent thinks that criterion perspective c_5 is more important to α than c_1, meta-level argument B_1 can be constructed using meta-level defeasible rule d_{aG} (as specified in Table 6.3). Namely, B_1 concludes that inferring to perform an alternative because it achieves the goal w.r.t. c_5 is stronger than inferring to performing an alternative because it achieves the goal w.r.t. c_1. For ease of notation, let r denote $d_{alwG}(\alpha, \mathsf{alt}_{\mathsf{re}}, c_1, G)$ (i.e., defeasibly inferring that agent α has the achievement goal to achieve an assignment in G) and r' denote $d_{alwG}(\alpha, \mathsf{alt}_{\mathsf{ere}}, c_5, G')$.

$$B_1 = \frac{c_1 \lhd_\alpha c_5}{\{r\} \prec_{\mathcal{R}} \{r'\}}\, d_{aG}$$

Note that $\{r\} = \mathsf{lastDef}(A_1)$ and $\{r'\} = \mathsf{lastDef}(A_4)$. Consequently, the following argument can be constructed.

$$B_2 = \frac{\mathsf{lastDef}(A_1) \neq \mathsf{lastDef}(A_4) \qquad B_1}{\mathsf{lastDef}(A_1) \neq \mathsf{lastDef}(A_4) \ \wedge\ \mathsf{lastDef}(A_1) \prec_{\mathcal{R}} \mathsf{lastDef}(A_4)}\ \wedge I$$

We can now apply the last link principle as represented in axiom $\mathtt{lstLnk}'_1$ (see Table 2.4). This is done in the following argument.

$$B_3 = \frac{B_2 \qquad \mathtt{lstLnk}'_1}{A_1 \prec_{\mathsf{Args}} A_4}\ \mathsf{MP}$$

Because of B_3, the attack of A_1 on A_4 becomes unsuccessful. Therefore, agent β makes move $m_6 = \langle 6, \beta, \mathsf{advance}_2(B_3), 5\rangle$ in which the meta-level argument B_3 is put forward. Note that because move m_6 attacks move m_5, move m_6 removes a winning part for α w.r.t. move m_1. Consequently, m_6 is strictly relevant in the dialogue.

Move 7 and 8

The student agrees with the support agent that c_5 is more important than c_1 and that therefore object-level argument A_4 is stronger than object-level A_1. Consequently, the student concedes to B_3's conclusion by making the move $m_7 = \langle 7, \alpha, \mathsf{concede}_2(A_1 \prec_{\mathsf{Args}} A_4), 6\rangle$. The student does not have any other relevant arguments that he can put forward and is thus persuaded by that argument A_1 is overruled and that he should not choose alternative $\mathsf{alt}_{\mathsf{re}}$.

Moreover, the student neither has any arguments that successfully attack the support agent's argument A_4, which concludes that the student should choose alternative $\mathsf{alt}_{\mathsf{ere}}$. Consequently, the student concedes to the conclusion of A_4 by making the dialogue move $m_8 = \langle 8, \alpha, \mathsf{concede}_1(\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{ere}})), 4\rangle$. Because they now agree on what the student should do, the student terminates the dialogue.

7.5 Chapter Summary

In order to argue to motivate decisions, a decision support agent needs to be able (1) to understand and construct arguments concerning decisions and (2) to communicate these arguments in a natural way with the user. Chapter 5 and 6 proposed how arguments concerning decisions can be constructed. In this chapter we have extended the dialogue framework in Prakken (2005a) such that it is possible to communicate all arguments that are relevant for a decision. More specifically, we have extended the communication language such that both object-level and meta-level arguments can be put forward and we have proposed a protocol for decision support that allows the agents to make what we call weakly related dialogue moves. Furthermore, we have proposed a general knowledge update function that updates the knowledge of an agent based on the dialogue move he observes. Because this is a difficult subject, we have informally described one approach to do this. Namely, observed claims are added as assumptions in the agent's knowledge base, which allows the agent to use the observed claims but remain skeptical towards them. Finally, the running example of the introduction was used to demonstrate the extended dialogue framework in the context of decision support.

McBurney et al. (2007) splits deliberation dialogues into eight stages. In each stage a different topic is discussed. For example, goals and criteria are discussed in the inform stage, whereas in the propose stage alternatives are suggested. In the dialogue framework of this chapter it is possible to discuss all the topics that are relevant to making a decision. However, our dialogue framework does not distinguish between stages in the dialogue. Because participants can make any legal dialogue move, a decision support agent may switch stage in every dialogue move he makes. For example, first an alternative is proposed, then what goals the agents has, then another alternative is proposed, and finally another goal may be put forward. If it is undesirable that the participants often switch in what stage they are, e.g., because it is chaotic for the student, then different stages could be modeled in our dialogue framework as either strategies for agents or by adding content-specific protocol rules.

8 Move Selection with Multiple Criteria

In the previous chapters we proposed a mechanism to support a human user in making a decision by means of a dialogue in which the user and the system can exchange arguments, counterarguments and questions. In order to argue to motivate decisions, we extended existing work in argumentation for decision-making with several concepts concerning perspectives. We then proposed an argumentation framework for decision-making that allows an agent to argue about what decision he should prefer most. To support the user in a dialogue, the previous chapter extended an existing dialogue framework such that all arguments relevant to a decision can be put forward. Finally, in this chapter we will address research question 2b by showing how our decision-making framework can be used by the decision support agent to decide what dialogue move has the most preferred outcome in the dialogue with the user.

During a support dialogue, there are several reasons for why there often are different dialogue moves that a participant could make. Firstly, if the set of alternatives to decide from is large, then for each alternative several arguments could be advanced that conclude that the alternative should be done or should not be done. Secondly, the larger the set of perspectives from which alternatives can be compared, the more comparisons between alternatives from different perspectives can be made. Finally, there may be a number of goals that can be justified from a perspective. A possible solution for move selection is to select a legal move randomly until all available legal moves are given. Although simple and comprehensive, the drawback of this solution is that the duration of the support dialogue will be long. To support a decision, it is typically not necessary to discuss all the possible decisions, just the ones that are the most relevant to the user. What decisions are relevant to a user depends on what the user cares about, i.e., the decisions that achieve goals with respect to perspectives that the user cares about.

A decision support agent may have different kinds of objectives in a support dialogue. For example, on the one hand, the agent may want to give the most persuasive arguments to the user in order to persuade the user as fast possible. On the other hand, he may want to be thorough with respect to discussing all aspects that may matter to the decision. If the only objective is to persuade the user that he should choose a certain alternative, then it is possible that the user is already persuaded without considering all relevant perspectives. In such situations, the decision support agent must make a tradeoff between these conflicting objectives. Section 8.1 describes various criteria found in the literature that can be used for selecting what dialogue move to make.

Because the decision support agent does not know what the user believes nor what the user values, the agent cannot determine with certainty what effect his dialogue moves will have on the user. Selecting what dialogue move to make is thus a multi-criteria decision task with uncertainty. In chapters 5 and 6 we proposed an argumentation framework for decision-making in domains with uncertainty and where multiple values, objectives and criteria matter. In Section 8.2 we will illustrate how this argumentation framework can be used to decide what move to select in a support dialogue. This approach is also taken in our paper van der Weide et al. (2011) and by Amgoud and Hameurlain (2007) who also consider move selection as a decision making task. In Section 8.3 we will conclude this chapter and compare our approach to their approach.

8.1 Criteria for Move Selection

Several criteria have been proposed in the literature that are interesting for dialogue move selection. For example, Oren et al. (2007) propose the criterion to minimise revealing information, which is relevant for domains with private information. Also, Amgoud and de Saint Cyr (2008) propose several measures, such as aggressiveness and coherence, to determine the quality of a persuasion dialogue. These measures could be used as criteria to decide what dialogue move to make. In this section, we will first discuss several criteria that are based on the values of agents, then multiple criteria that concern the acceptability of an argument for an agent, and finally criteria concerning whether the agent is interested in the argument. Note that many criteria require much information about the audience, which is typically not available. In this case, it may be possible to make an educated guess about what is actually true and use these criteria as heuristics instead.

Criteria can utilize different kinds of information like what knowledge the audience has, the preferences of the audience, or what has been said in the dialogue before. Subsection 8.1.1 first discusses two criteria based on the values the audience holds, i.e., the audience's preferences. Next, a number of criteria are discussed in Subsection 8.1.2 that compare moves based on what the audience believes. Finally, Subsection 8.1.3 explores two criteria that use the interests of the audiences to compare moves.

8.1.1 Value-based Criteria

Perelman and Olbrechts-Tyteca (1969) claim that whether an agent will accept an argument significantly depends on whether the values in the argument match the values held by the agent to whom the argument is presented. If an agent wants to maximize the likelihood that another agent agrees, then he could use the criterion to put forward arguments that use or are in harmony with the other agent's values. Grasso et al. (2000) are interested in changing the attitudes of a user to promote healthy nutrition. For this purpose, they formalize the theory by Perelman and Olbrechts-Tyteca (1969) and use the values held by the user to select the most convincing argument.

Black and Atkinson (2011) propose a dialogue system for joint deliberation where each agent has a different value ordering. Dialogue moves consist of putting forward practical arguments using the practical argument scheme AS1 by Atkinson et al. (2006). Agents determine what argument to put forward using the heuristic that the more important a value is for

the recipient agent, the better it is to advance an argument on the basis of that value. In this way, agents use the value ordering of another agent to select what argument to put forward in a persuasion dialogue.

8.1.2 Acceptability of Arguments

In Hunter (2007) and Black and Hunter (2009), agents have beliefs about the degree to which formulae can be used as shared knowledge when interacting with other agents. If a formula is shared knowledge between two agents, then both agents believe that formula is true. Agents can use their beliefs about whether an argument's premises are shared knowledge as a criterion to maximize the likelihood that another agent accepts a certain argument. Hunter (2004a) models the 'believability' of arguments for specific agents. For this purpose, the 'empathy' and 'antipathy' of arguments for an agent is defined. To maximize the believability of an argument, both the empathy for the defending arguments and the antipathy of the attacking arguments should be maximized. Maximizing the believability of arguments could be a useful criterion for move selection in dialogues.

In Riveret et al. (2008) the expected utility of dialogue moves in an adjudication dialogue is determined using the probability that the adjudicator accepts the argument's premises and the probability that the argument is attacked. Given these probabilities, possible criteria for advancing an argument A is to maximize the probability that the receiving agent accepts the argument A's premises and to minimize the probability that the receiving agent will attack A.

It is also possible to consider the relation between the strength of an argument and the agent who utters that argument. Hunter (2008) provides a logic-based framework to evaluate arguments by looking how 'appropriate' the speaker of the argument is for that particular argument. An agent reasons on the meta-level about whether an agent is appropriate to assert some argument. If an agent α has asserted argument A and α is appropriate to assert A, then A should be accepted. Assume that agent α wants to persuade agent β and that β evaluates arguments by looking at the appropriateness of the agent who advances them. If α can make an educated guess of whether β will find α appropriate, then α could use this as a criterion for move selection.

8.1.3 Interest

In Hunter (2004b), what an agent finds important is used to determine the impact of arguments. A 'desideratabase' is defined, which captures how much an agent is interested in certain formulae. A desideratabase is a tuple $\langle \Pi, \lambda \rangle$ with Π a set of formulae that the agent wants to be true and λ a mapping from those formulae to $[0, 1]$. The higher $\lambda(\phi)$ for some $\phi \in \Pi$, the more the agent wants ϕ to be true. The 'resonance' of an argument is then defined as the sum of $\lambda(\phi)$ for all $\phi \in \Pi$ that are affected by that argument. Consequently, the higher the resonance of an argument, the more important an argument is for the agent. Maximizing the resonance of an argument could be used as a criterion to select the argument with the highest impact.

In addition, van der Weide et al. (2009a) investigate how the personality type of a person can be used to determine in what premises and critical questions of arguments schemes that agent is most interested in. The techniques are based on Zeisset (2006), who describes how to use the MBTI personality theory by Jung (1921) to communicate effectively. For example,

the Sensing personality type first wants and gives information that is real, concrete, practical, factual, and specific, whereas the Intuition personality type first wants and gives information that is insightful, opens possibilities, uses the imagination, presents an overview or synthesis, and shows patterns. Sensing types ask what and how questions and speak of what is or what has been and give precise factual descriptions. In contrast, intuition types ask what if, and why questions and speak of what might be, what the main issue is, and what jumped out using 'sort of' and general impression descriptions. Agents could use the criterion to give the information that another agent is most interested in.

8.2 Running Example

In this section we will apply our argumentation framework for decision-making on the decision of selecting what move to make. The student (denoted α) is in the decision situation described in the running example. The decision support agent β is having a support dialogue with α about what decision α should make. After a number of moves, β must choose between multiple decisions. To determine what alternative β should choose, our argumentation framework for decision-making is used. Note that we are dealing with two decision situations, i.e., the student's decision about what to do in the scenario with the fire in the factory and the support agent's decision about what dialogue move to make in the dialogue with the student.

We will first describe the current circumstances for β after which we will informally describe what matters to β for his decision about what move to select. Then this decision situation is formalized in the PVM model (see Chapter 4). This formalization is then used to instantiate the argumentation framework for perspective-based value as described in Chapter 5. Arguments are then constructed to determine what move β should make.

The current dialogue

The student (denoted α) is in the decision situation described in the running example. Let α and decision support agent β be in dialogue[1] $d = \langle \delta, [m_1, m_2] \rangle$ in which they are discussing what decision α should take. In dialogue d, two moves m_1 and m_2 have been made where:

- in move m_1, student α advances argument A_1 concluding that α should choose alternative $\mathsf{alt}_{\mathsf{re}}$ (i.e., send firefighters in immediately to rescue the victims, then extinguish fire) because $\mathsf{alt}_{\mathsf{re}}$ achieves α's goal of that the victims are inside for ten minutes or less (criterion perspective c_1);
- in move m_2, agent β attacks m_1 by advancing argument A_2, which concludes that α should perform alternative $\mathsf{alt}_{\mathsf{ere}}$ (i.e., first extinguish the fire near the victims, then rescue them and finally extinguish the rest of the fire) because it achieves α's goal w.r.t. how much time the victims are near fire (criterion perspective c_5).

Because m_2 is warranted and m_2 attacks m_1, move m_1 is not warranted. However, the support agent β wants α to be aware of all perspectives that β thinks α may find important.

[1] Recall from Definition 7.6 that a dialogue is a tuple consisting of a decision context δ and a possibly infinite sequence $[m_1, m_2, \ldots]$ of dialogue moves in δ. A dialogue context consists of a tower of argumentation systems, a communication language and a set of participants.

There are several dialogue moves that β could choose to make. Because the support agent cares about multiple criteria when deciding what move to make, the PVM argumentation framework is used to determine what dialogue move or sequence of dialogue moves he should make. Recall from Definition 7.15 that in the protocol for decision support, participants are allowed to make multiple moves in the same turn. We will use sequences of dialogue moves as the alternatives from which β can choose. Note that there are many (possibly infinite) legal sequences of dialogue moves from which β can choose. However, this is not a problem in the PVM argumentation framework because it does not enforce the agent to consider all possible alternatives. For the example, we will focus on the following three alternatives that β can choose.

- Alternative alt'_a: first make move m_3 conceding some premise of A_1, then make move m_4 advancing argument A_3 justifying alternative $\mathsf{alt}_{\mathsf{ere}}$ because it achieves α's achievement goal w.r.t. the amount of chemicals escaping into the environment. Note that m_4 attacks m_1.
- Alternative alt'_b: only make move m_4.
- Alternative alt'_c: first make move m_3, then attack move m_1 with move m_5. Move m_5 advances argument A_4, which concludes that alternative $\mathsf{alt}_{\mathsf{re}}$ should not be done because it achieves an avoidance goal w.r.t. the safety of the firefighters. Namely, the firefighters have to go into the building while it is on fire.

Note that alternative 1 only differs from alternative 2 by first conceding a premise claimed by α. Further note that both alternative 1 and 3 first concede a premise of α's argument A_1.

Value tree in support dialogues

We will now describe the value tree of decision support agent β. First, we will describe the general areas of concern that matter to β in the dialogue and then decompose them into more specific perspectives. Note that we do not aim at giving precise definitions of these general areas of concern, but that we want to show how influence between perspectives can be used to decompose an abstract perspective into several more specific perspectives. The PVM model is designed in such a way that abstract perspectives do not have to be specified exactly, but that they can be described in a less strict manner.

Agent β has three general areas of concern: *persuading* the student, while being *efficient* and remaining *friendly*. To make clear what this means for β, we will decompose what β values into a value tree. We will now focus on each area of concern and decompose them into more specific perspectives. This decomposition is captured in a value tree. Given this value tree, the next section will instantiate the PVM for β, which we can then use to argue about what dialogue move β should make.

Efficiency

In a decision support scenario, there is typically a limited amount of time available to discuss what decision to make. Therefore, the support agent should not make all legal moves, but carefully select the most important moves. Similarly, if in a training setting a support agent would make all legal moves, then the student would not like working with the support agent. To prevent the support agent to make all dialogue moves that are legal, the agent should care

about being 'efficient'. What makes a dialogue efficient is a complicated matter, but using the PVM we do not have to specify it exactly. For the example, we will use the simple heuristic that the more moves that are made in a dialogue, the less efficient the dialogue is.

Persuasiveness

We will now describe how we interpret persuasiveness within the context of this example. We see persuading someone as causing a person to do something or to cause someone believe something by means of argument. In our framework, a formula ϕ could denote that the student should choose some alternative, but also that the student should have a certain goal or that performing some alternative results in a certain outcome. Persuading an agent thus means causing that the agent agrees that some formula ϕ is true. We consider the *persuasiveness of a dialogue move m for a formula ϕ* as the likelihood that the student will accept the truth of ϕ caused by m.

Assume that move m advances argument A. Whether m causes the student to accept the truth of some formula ϕ depends on whether the student accepts A's premises and how strong A is compared to other arguments of the student. If A's premises are not acceptable, then the student will consider A as overruled. Because the relative strength of arguments determines whether an attack is successful, A's strength is important for whether A 'survives' all attacks. The stronger the argument, the more likely its conclusion will be accepted. Therefore, the decision-support agent decomposes his general area of concern 'persuasiveness' into the objectives of maximizing acceptability of the premises and maximizing argument strength.

The relative strength of arguments concluding what alternative to choose depends on how important the student finds the goal that they achieve and how certain it is that the alternative achieves that goal. The relative importance of goals for a student depends on how important the student finds their corresponding criterion perspectives. Namely, the more important a criterion, the more important a goal w.r.t. that criterion, and the more important it is to achieve that goal. In the PVM, the strength of an argument depends on the importance of the perspectives that it uses. Therefore, to maximize argument strength and thus persuasiveness, the importance of the perspectives used in arguments should be maximized. Note that the support agent β may have an idea about what perspectives the student finds important, but that this knowledge may be incomplete and incorrect.

Whether the premises of an argument are acceptable to the student depends on what premises are in his knowledge base, the arguments that he has constructed and the acceptability of those arguments. For example, a premise ϕ is acceptable to the student if he has constructed a justified argument that concludes ϕ, but ϕ is not acceptable if the student has no arguments that conclude ϕ or the student has no justified or defensible arguments concluding ϕ. To determine whether the student will accept a premise it is thus necessary to know the content of the student's knowledge base and what arguments he has constructed. This information is typically not available and therefore the best the decision support agent can do is to make an educated guess. A possible heuristic for this is to use premises that the student has claimed, or in other words, to *borrow* premises from the student. The underlying idea is that if the student has claimed that a formula ϕ is true, then typically the student finds ϕ acceptable. In other words, the more premises an argument borrows from the student, the more likely it is that the student will accept the argument.

Friendliness

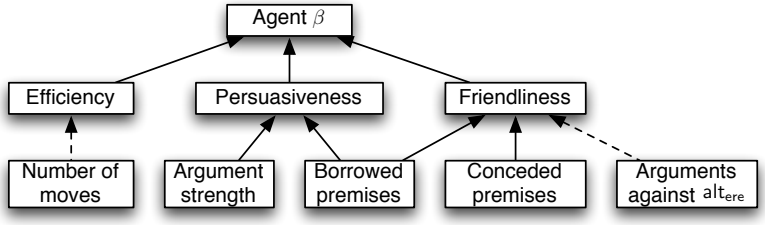
The third general area of concern of the decision-support agent is 'friendliness'. Because friendliness is abstract, we will now decompose it into several more concrete perspectives that have an influence on friendliness. Again note that we do not aim to give a precise definition of friendliness, but that we will use five perspectives that are relevant for friendliness to describe friendliness. One perspective that we say is important for friendliness is reusing or borrowing the claims of the other agent. By borrowing a premise that the student has claimed, the student may feel that his judgment is appreciated. In response, the student may more easily appreciate the judgment of the decision support agent, which contributes to the likelihood that the student will accept an argument put forward by the agent. Therefore, the criterion of how many premises are borrowed has a positive influence on the general area of concern of friendliness. For example, if argument A_1 borrows no premises of the student and argument A_2 borrows three premises, then A_2 is preferred to A_1 from the criterion perspective of how many premises are borrowed, which is a reason to think that A_2 is friendlier than A_1. For similar reasons, the criterion of how many of the student's premises are conceded by the agent also positively influences the friendliness of the agent towards the student.

In the dialogue, the student has put forward argument A_1 concluding that he should send the firefighters in immediately to rescue the victims and then extinguish the fire. The support agent thinks it is a better idea to first extinguish just the fire near the victims, then get them out and then extinguish the rest of the fire. The agent could give the student arguments not to do the student's alternative, but the agent could also give the student arguments to do the alternative of the agent. If the agent first only advances arguments that say the student should not do X, then the student could experience that as aggressive. Namely, the agent could also advance arguments to do Y. The student may then realize that X has disadvantages w.r.t. the perspectives that the agent uses to justify doing Y. In that way, the agent does not have to make explicit why X may not be a good decision.

Value Tree

The decision support agent cares about the general areas of concern of efficiency, persuasiveness and friendliness. We have decomposed these general areas of concern into more concrete perspectives. This decomposition represents the value tree of the decision support agent. Figure 8.1 visualizes the value tree of the decision support agent in the dialogue with the student. In actual applications, the decision support agent may also have to consider more perspectives, but we will focus on these perspectives for the sake of simplicity.

Figure 8.1: *Value tree for move selection*



Although the decision support agent cares about each general area of concern, he does find some more important than others. For the sake of simplicity, we will only assume that agent β finds the perspective of persuasiveness more important than the perspective of friendliness.

Perspective-based Value Model

In order to argue about what dialogue move the support agent can best select, we need an argumentation framework for PV that is instantiated with a perspective-based value model as defined in Chapter 4. Recall that we are dealing with two decision situations: the student's decision about what to do in the scenario with the fire and the factory and the support agent's decision about what dialogue move to make in the dialogue with the student. We will accent perspectives and attributes to denote that they are used for move selection.

Table 8.1 describes the perspectives in the value tree of the support agent β as visualized in Figure 8.1. Note that we do not have any objective perspectives. The perspectives in

Table 8.1: *Perspectives in the value tree of the system*

Perspective	Denotes
β'	The decision support agent
effcn'	Efficiency
prsv'	Persuasiveness of $\mathsf{do}(\alpha, \mathsf{alt}_{\mathsf{ere}})$ to α
frnd'	Friendliness towards α
c'_1	Number of dialogue moves made in the dialogue between α and β
c'_2	Strength of arguments advanced by β
c'_3	Number of α's claims that β has borrowed
c'_4	Number of α's claims that β has conceded
c'_5	Number of arguments against $\mathsf{alt}_{\mathsf{re}}$

Table 8.1 are modeled with the following perspective structure.

$$\mathcal{P}_{\mathsf{move}} = \langle\ \{\beta'\},\ \{\mathsf{effcn}', \mathsf{prsv}', \mathsf{frnd}'\},\ \emptyset,\ \{c'_1, c'_2, c'_3, c'_4, c'_5\}\ \rangle$$

Figure 8.1 visualizes several influence relations between the perspectives in $\mathcal{P}_{\mathsf{move}}$. The perspective influence structure (as in Definition 4.10) $\iota_{\mathsf{move}} = \langle \mathcal{P}_{\mathsf{move}}, I_{\uparrow}, I_{\downarrow} \rangle$ models these influences where

$$\begin{aligned} I_{\uparrow} =& \{(\mathsf{effcn}', \beta'),\ (\mathsf{prsv}', \beta'),\ (\mathsf{frnd}', \beta'),\\ & (c'_2, \mathsf{prsv}'),\ (c'_3, \mathsf{prsv}'),\ (c'_3, \mathsf{frnd}'),\ (c'_4, \mathsf{frnd}'),\\ & (c'_2, \beta'),\ (c'_3, \beta'),\ (c'_4, \beta')\}\\ I_{\downarrow} =& \{(c'_1, \mathsf{effcn}'),\ (c'_5, \mathsf{frnd}'),\ (c'_1, \beta'),\ (c'_5, \beta')\} \end{aligned}$$

For each of the five criterion perspectives in $\mathcal{P}_{\mathsf{move}}$ an attribute is used that measures that perspective. We are only interested in those five attributes, so the set of all attributes is $\mathcal{A} = \{x'_1, x'_2, x'_3, x'_4, x'_5\}$. Table 8.2 describes these five attributes. For each attribute, the table presents what the attribute denotes and its domain of attribute values. We will use $\mathcal{P}_{\mathsf{fire}}$ to the denote the perspectives in the firefighting decision scenario.

Table 8.2: *Attributes for Move Selection*

Attribute	Denotes	Attribute Values
x'_1	Total number of dialogue moves made in the dialogue	$\mathbb{N}$
x_2	Perspectives[a] used by arguments advanced by β	$2^{\mathcal{P}_{\mathsf{fire}}}$
x'_3	Number of α's claims that β borrows	$\mathbb{N}$
x'_4	Number of α's claims that β has conceded	$\mathbb{N}$
x'_5	Number of β's arguments that conclude not to do alternative a	$\mathbb{N}$

[a] Note that $\mathcal{P}_{\mathsf{fire}}$ denotes the student's perspectives in determining what to do with the fire rather than the perspectives of the agent's value tree in deciding what move to make to the student.

The resulting outcomes from choosing either alt'_a, alt'_b and alt'_c are in Table 8.3. Because alt'_a and alt'_c both consist of two dialogue moves and two dialogue moves have already been made in the dialogue, alt'_a's and alt'_c's resulting x'_1-value is 4. The x'_1-value of alt'_b is 3 because it only consists of one move. Alternatives alt'_a and alt'_b advance the same argument and therefore they result in the same x'_2-value of fire-perspectives $\{c_4, o_1, \alpha\}$. Alternative alt'_c advances a different argument that uses different fire-perspectives. No advanced argument borrows any premises from α and therefore all x'_3-values are 0. Alternatives alt'_a and alt'_c both result in x'_4-value of 1 because they concede a premise of α's argument A_1. Finally, only alt'_c advances one argument that concludes that α should not do alternative $\mathsf{alt}_{\mathsf{re}}$. Therefore, alt'_c's x'_5-value is 1 and the other x'_5-values are 0.

Table 8.3: *Outcomes of the alternatives*

Alternative	x'_1	x'_2	x'_3	x'_4	x'_5
alt'_a	4	$\{c_4, o_1, \alpha\}$	0	1	0
alt'_b	3	$\{c_4, o_1, \alpha\}$	0	0	0
alt'_c	4	$\{c_6, o_3, \alpha\}$	0	1	1

We will use s'_a, s'_b and s'_c to denote the assignments resulting from alt'_a, alt'_b and alt'_c respectively. For the example, we will only consider the assignments resulting from the various alternatives, i.e., $\mathcal{S}_{\mathsf{move}} = \{s'_a, s'_b, s'_c\}$.

The value comparisons from the criteria perspectives must correspond to the attribute values. This means that if y_1 and y_2 are two attribute values of attribute x'_i and $y_1 \leq y_2$, then the PVCS should be instantiated with this information. For criterion perspective c'_2, we will assume that β believes that the set $\{c_6, o_3, \alpha\}$ is more important for α than the set $\{c_4, o_1, \alpha\}$. We can now define the PVCS (see Definition 4.7) corresponding to this example as

$$\delta_{\mathsf{move}} = \langle \mathcal{A}_{\mathsf{move}}, \mathcal{S}_{\mathsf{move}}, \mathcal{P}_{\mathsf{move}}, \preceq, \mathsf{msr} \rangle$$

with:

- for $i \in \{1, 3, 4, 5\}$: $y_1 \leq y_2$ iff $\{(x'_i, y_1)\} \preceq_{c'_i} \{(x'_i, y_2)\}$
- for $1 \leq i \leq 5$: $\mathsf{msr}(c'_i) = \{x'_i\}$, and
- $s \prec_{c'_2} t$ with $s = \{(x'_2, \{c_4, o_1, \alpha\})\}$ and $t = \{(x'_2, \{c_6, o_3, \alpha\})\}$.

Note that all the ingredients have been introduced that are necessary to instantiate our argumentation framework for perspective-based value.

Perspective-based Argumentation

We will now use the argumentation system for perspective-based value (ASPV) as in Definition 5.3 to argue about what alternative β should choose. Let $\mathcal{AS}_{\mathsf{pv}}$ be the ASPV w.r.t. PVCS δ_{move}. Furthermore, let $\mathcal{K} = \langle \mathcal{K}_{\mathsf{np}}, \mathcal{K}_{\mathsf{op}}, \mathcal{K}_{\mathsf{as}} \rangle$ be agent β's knowledge such that $\mathcal{K}$ corresponds to δ and ι_{move} (see Definition 5.4 and right below there).

Because after performing alt'_a four moves have been made in total whereas after performing alt'_b only three moves have been made, the ordinary premise $s'_b <_{c'_1} s'_a$ is in the agent's knowledge base (recall that s'_a, s'_b and s'_c are the assignments resulting from performing alternatives alt'_a, alt'_b and alt'_c). Because c'_1 negatively influences the perspective effcn' and because effcn' positively influences β's perspective, the following two arguments can be constructed. These arguments use the defeasible rules $d_\uparrow$ and $d_\downarrow$, which are described in Table 5.2.

$$B_1 = \frac{s'_b <_{c'_1} s'_a \qquad c'_1 \downarrow \mathsf{effcn}'}{s'_a <_{\mathsf{effcn}'} s'_b}\, d_\downarrow \qquad B_2 = \frac{B_1 \qquad \mathsf{effcn}' \uparrow \beta'}{s'_a <_{\beta'} s'_b}\, d_\uparrow$$

Similar arguments can be constructed concluding that $s'_a <_{\mathsf{effcn}'} s'_c$ and $s'_a <_{\beta'} s'_c$. These two arguments will be denoted with B'_1 and B'_2.

The argument that is put forward in alternative alt'_c uses the fire-related set of perspectives $\{c_6, o_3, \alpha\}$, whereas the argument in alt'_a uses fire-related perspectives $\{c_4, o_1, \alpha\}$. Because the agent believes that α finds the perspectives in alt'_c more important, $s'_a <_{c'_2} s'_c$ is an ordinary premise in the knowledge base. Using this knowledge, the following two arguments can be constructed.

$$B_3 = \frac{s'_a <_{c'_2} s'_c \qquad c'_2 \uparrow \mathsf{prsv}'}{s'_a <_{\mathsf{prsv}'} s'_c}\, d_\uparrow \qquad B_4 = \frac{B_3 \qquad \mathsf{prsv}' \uparrow \beta'}{s'_a <_{\beta'} s'_c}\, d_\uparrow$$

Namely, because β thinks that the fire-related perspectives in s'_c are more important to α than the fire-related perspectives in s'_a, argument B_3 concludes that s'_c is preferred to s'_a from the perspective of persuasiveness. Argument B_4 uses B_3 to conclude that thus β should prefer s'_c. Because β also has the premise $s'_b <_{c'_2} s'_c$ in its knowledge base, two similar arguments B'_3 and B'_4 can be constructed concluding $s'_b <_{\mathsf{prsv}'} s'_c$ and $s'_b <_{\beta'} s'_c$ respectively, i.e., that s'_b is less persuasive than s'_c and thus that β should prefer s'_b.

Note that no alternative borrows any premises from α. Consequently, s'_a, s'_b and s'_c are equally preferred from the criterion perspective c'_3. How many of α's premises are conceded (criterion perspective c'_4) positively influences friendliness towards α and how many advanced arguments not to do fire-alternative $\mathsf{alt}_{\mathsf{re}}$ (criterion perspective c'_5) negatively influences friendliness towards α. In alt'_b nothing is conceded and $\mathsf{alt}_{\mathsf{re}}$ is not attacked, but in alt'_c a premise of α is conceded, but $\mathsf{alt}_{\mathsf{re}}$ is attacked. This results in the following two arguments.

$$B_5 = \frac{s'_b <_{c'_4} s'_c \qquad c'_4 \uparrow \mathsf{frnd}'}{s'_b <_{\mathsf{frnd}'} s'_c}\, d_\uparrow \qquad B_6 = \frac{s'_b <_{c'_5} s'_c \qquad c'_5 \downarrow \mathsf{frnd}'}{s'_c <_{\mathsf{frnd}'} s'_b}\, d_\downarrow$$

Note that arguments B_5 and B_6 have conflicting conclusions w.r.t. whether s'_b or s'_c is preferred from the perspective of friendliness.

In contrast, s'_a is preferred to s'_c from the perspective of friendliness because both s'_a and s'_c concede one premise, but s'_a does not advance an argument not to do $\mathsf{alt}_{\mathsf{re}}$.

$$B_7 = \frac{s'_a <_{c'_5} s'_c \qquad c'_5 \downarrow \mathsf{frnd}'}{s'_c <_{\mathsf{frnd}'} s'_a}\, d_\downarrow \qquad B_8 = \frac{B_7 \qquad \mathsf{frnd}' \uparrow \beta'}{s'_c <_{\beta'} s'_a}\, d_\uparrow$$

Note that B_8 and B_4 rebut each other. Namely, s'_c is better because it is more persuasive, but s'_a is better because it is more friendly. Because B_8 concludes $s'_c <_{\beta'} s'_a$ and B_2 concludes $s'_a <_{\beta'} s'_b$, the defeasible rule d_{strct} as in Table 5.2 can be used to infer defeasibly that $s'_c <_{\beta'} s'_b$ is true. This is done in argument B_9.

$$B_9 = \frac{B_8 \qquad B_2}{s'_c <_{\beta'} s'_b}\, d_{strct}$$

Note that because B_4 rebuts B_8 and B_8 is a subargument of B_9, argument B_4 also rebuts B_9.

Meta-Level Argumentation

Object-level arguments B_5 and B_6 attack each other and B_4 and B_8 attack each other. However, support agent β does not find all perspectives in his value tree as important. Namely, β finds persuasiveness more important than friendliness.

Recall Definition 5.8 that defines meta-argumentation systems for PV w.r.t. an argumentation system for PV. Let $\mathcal{AS}'_{\mathsf{pv}}$ be the meta-argumentation system for PV w.r.t. $\mathcal{AS}_{\mathsf{pv}}$ (with $\mathcal{AS}_{\mathsf{pv}}$ the ASPV as defined in the previous subsection). Furthermore, let $\mathcal{K}' = \langle \mathcal{K}'_{\mathsf{np}}, \mathcal{K}'_{\mathsf{op}}, \mathcal{K}'_{\mathsf{as}} \rangle$ be the meta-level knowledge base such that $(\mathsf{frnd}' \lhd_{\beta'} \mathsf{prsv}') \in \mathcal{K}'_{\mathsf{op}}$.

The following meta-level arguments can be constructed from $\mathcal{K}'$ using defeasible rule $d'_{\uparrow\uparrow}$ from Table 5.4. We will use r_4 to denote B_4's last applied defeasible rule $d_{\uparrow}(\mathsf{frnd}', \beta', s'_c, s'_a)$ and r_8 to denote B_8's last applied defeasible rule $d_{\uparrow}(\mathsf{prsv}', \beta', s'_a, s'_c)$.

$$C_1 = \frac{\mathsf{frnd}' \lhd_{\beta'} \mathsf{prsv}'}{\{r_4\} \prec_{\mathcal{R}} \{r_8\}}\, d'_{\uparrow\uparrow}$$

Because $\mathsf{lastDef}(B_4) = \{r_4\}$ and $\mathsf{lastDef}(B_8) = \{r_8\}$, it is the case that argument C_1 concludes $\mathsf{lastDef}(B_4) \prec_{\mathcal{R}} \mathsf{lastDef}(B_8)$. Furthermore, because $\{r_4\}$ is not the same as $\{r_8\}$, it is true that $\mathsf{lastDef}(B_4) \neq \mathsf{lastDef}(B_8)$. The strict rule $\wedge I$ introduces a logical 'and'.

$$C_2 = \frac{\mathsf{lastDef}(B_4) \neq \mathsf{lastDef}(B_8) \qquad C_1}{\mathsf{lastDef}(B_4) \neq \mathsf{lastDef}(B_8) \wedge \mathsf{lastDef}(B_4) \prec_{\mathcal{R}} \mathsf{lastDef}(B_8)}\ \wedge I$$

The following argument uses the last-link principle axiom $\mathtt{lstLnk}'_1$ to conclude that B_8 is a stronger argument than B_4 because B_8's last applied defeasible rules are stronger.

$$C_3 = \frac{C_2 \qquad \mathtt{lstLnk}'_1}{B_4 \prec B_8}\ \mathsf{MP}$$

Agent β does not find criterion perspective c'_4 or c'_5 more important for the perspective of friendliness. Consequently, we cannot infer whether B_5 or B_6 is stronger. Because C_1, C_2 and C_3 do not have any attackers, they are all in the grounded extension. This means that C_3 is a justified argument and that $B_4 \prec B_8$ is a justified conclusion. Consequently, in the resulting grounded argument ordering over object-level arguments it is true that $B_4 < B_8$.

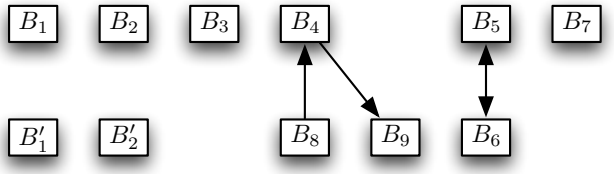
Acceptability of Object-Level Arguments

Now that we have determined the grounded argument ordering $\leq$ resulting from the meta-argumentation, we can construct the object-level argumentation theory $\mathcal{AT} = \langle \mathcal{AS}_{\mathsf{pv}}, \mathcal{K}, \leq \rangle$.

Because in the argument ordering it is true that $B_4 < B_8$, the attack of B_4 on B_8 is unsuccessful (see the definition of defeat in Definition 2.16).

Argumentation theory $\mathcal{AT}$ can now be used to instantiate the argumentation framework $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Attack} \rangle$ with $\mathsf{Args} = \{B_1, B'_1, \ldots, B_8\}$ the set of arguments that can be constructed from $\mathcal{K}$ and Attack the defeat relation between Args. Argumentation framework $\mathcal{AF}$ is visualized in Figure 8.2.

Figure 8.2: *Object-level arguments concerning what move to select*



Every argument is justified except for B_4, B_5 and B_6. Because B_8 defeats B_4 and B_8 is justified, argument B_4 is overruled. Because B_4 attacks B_9 but B_4 is overruled, argument B_9 is reinstated and is a justified argument. Arguments B_5 and B_6 are both defensible.

The justified conclusions w.r.t. (1) efficiency are $s'_a <_{\mathsf{effcn}'} s'_b$ and $s'_a <_{\mathsf{effcn}'} s'_c$; (2) persuasiveness are $s'_a <_{\mathsf{prsv}'} s'_c$ and $s'_b <_{\mathsf{prsv}'} s'_c$; and, most importantly, (3) β's perspective $s'_c <_{\beta'} s'_a$, $s'_a <_{\beta'} s'_b$ and $s'_c <_{\beta'} s'_b$. The defensible conclusions w.r.t. friendliness are $s'_c <_{\mathsf{frnd}'} s'_b$ (argument B_5's conclusion) and $s'_c <_{\mathsf{frnd}'} s'_b$ (argument B_6's conclusion). This means that it is not clear for agent β whether s'_b or s'_c is more friendly.

Summarizing, given what β finds important in a dialogue, how that is decomposed into measurable criteria and how the various alternatives compare on these criteria, agent β used the PVM to construct arguments concerning what dialogue moves he should make. By evaluating these arguments, agent β is justified to prefer assignment s'_b the most. Because alternative alt'_b results in s'_b, agent β should decide to perform alternative alt'_b, which consists of only making dialogue move m_4 as described in Section 8.2.

8.3 Chapter Summary

In a decision support dialogue, the support agent can typically choose from multiple dialogue moves. Furthermore, the support agent typically has multiple objectives in the dialogue. Because these objectives may be different by nature, they may introduce tradeoffs. For example, a support agent may have the objective to be efficient, but also have the objective to be comprehensive. The support agent has to make a tradeoff when the decision maker is already persuaded why some alternative is better, but not all the important advantages and disadvantages have been discussed. In that case, terminating the dialogue is efficient, but discussing it further is comprehensive.

This chapter demonstrated how move selection can be treated as a decision making problem and demonstrated how the mechanism proposed in the previous chapters can be used for different kinds of decision problems. We have first discussed a variety of criteria in the literature that can be used by the support agent to determine what dialogue moves to make. Then

we used the running example from the introduction to show how the argumentation mechanism of Chapter 5 can be used by the support agent to reason about what dialogue moves to make. Namely, the support agent is in a support dialogue (as defined in Chapter 7) with the student and has to decide what dialogue moves to make. The conceptual framework of Chapter 3 was used to decompose what the support agent values in a dialogue into a value tree consisting of general areas of concern, objectives and measurable criteria. Next, the support agent's value tree was formalized using the formalism proposed in Chapter 4. This formalization was then used to instantiate the argumentation logic from Chapter 5. Finally, this argumentation logic was used to argue about what dialogue moves result in the outcome that the support agent values most.

Amgoud and Hameurlain (2007) are mostly interested in negotiation dialogues and propose to treat move selection as a two-step decision making task: first select the kind of locution and then select the content of the locution. In other words, to select a move, an agent has to make two decisions. For each of these two decisions, agents are assumed to have different kinds of goals (and beliefs). For selecting what kind of locution to make, agents have so-called strategic goals. For example, 'minimizing the dialogue time' or 'selling at the end of the dialogue'. For selecting the content of the locution, agents have so-called function goals, which directly relate to the topic of the dialogue.

In contrast to Amgoud and Hameurlain (2007), we did not separate the locution and the content, but we treated sequences of dialogue moves as alternatives. Recall from Section 2.4 that this is in line with approaches in decision theory. Furthermore, we argue that in most kinds of dialogues it is not useful to separate the locution from the content, because the combination of content and locution determines the effect on other participants. Although we did not distinguish between functional and strategic goals, it is possible to classify the criteria perspectives in an agent's value tree as either strategic or functional. For example, in Section 8.2, the criterion perspective of 'conceded premises' could be classified as strategic, whereas the criterion perspective 'arguments against $\mathsf{alt}_{\mathsf{re}}$' as functional. We therefore argue that our approach for move selection is a more general approach that can be used in more types of dialogues.

9 Conclusion

The main research question that was addressed concerns how argumentation can be used to support decision making in complex scenarios. In situations where it has to be decided between, for example, a higher and a lower profit, making a decision is simple, but when multiple different aspects matter this problem is difficult and a decision maker may benefit from being supported. In a wide variety of situations decision makers care about multiple aspects. Some examples of such situations are the following two. When deciding what camera to buy, aspects matter such as price, image quality, size, weight, whether a camera gets good reviews, and so on. A fire commander cares about aspects like the safety of the public, safety of personnel, the environment, obedience to the rules, and costs when deciding what to do if there is a fire and people are in danger. When determining what decision to prefer, a decision maker may forget to take into account some important aspect or use the wrong evaluation criteria. Therefore, the decision maker may benefit from a system that can support him in making a decision.

In the introduction we have identified two requirements for arguing to motivate decisions. Firstly, in order for a system to support a decision, the system must be able to reason about what decision is the best for a particular user. Secondly, the system must be able to communicate effectively to the user why a certain decision is the best and why another decision is not. After exploring related work, we have formulated five research questions to address several issues in using existing work for our purposes. Research questions 1a, 1b and 1c concern how the system can reason about what decision is the best and research questions 2a and 2b concern how a user can be supported effectively. In this chapter we will conclude the thesis by summarizing how this thesis has answered these five research questions.

In the literature, a wide variety of concepts and techniques have been proposed for decision making. Each approach has its strengths, but it was not clear how these concepts related to each other. The main contribution of this thesis is a rich argumentation framework to argue to motivate decisions in which several concepts and techniques from a variety of disciplines are integrated. Namely, techniques from decision analysis were combined with techniques from argumentation theory to propose a qualitative mechanism to argue about value. Furthermore, we have shown how our framework relates to the concept of value in the psychology literature and to existing approaches in the argumentation literature.

9.1 Identifying The Best Decision

What a person values is the motivation to spend any effort in making a decision (Keeney, 1992). To support a user in making a decision it is thus necessary to understand what it means to value something. Research questions 1a, 1b, and 1c concern how to identify the best decision and in this section we discuss how this thesis has addressed these three questions.

A popular approach to represent what a decision maker values is to use utilities. The subjective utility of an outcome is a numerical value that represents how desirable that outcome is to the decision maker. A utility function maps outcomes onto utility values. To support a decision it is necessary to know the decision maker's utility function. However, utility functions are hard to elicit and in complex domains, decision makers do not know what utility to give to outcomes. In the field of decision analysis, researchers have investigated how to find the decision maker's subjective utility function. In popular approaches like the ones of von Winterfeldt and Edwards (1986), Keeney (1992), and Saaty (2008), a decision analyst is in a dialogue with the decision maker questioning him about what he finds important. When the decision maker states that something is important, then the decision analyst asks why this is important and how it can be made more specific. In this way, what the decision maker values is decomposed until it can be described by a set of specific criteria. von Winterfeldt and Edwards (1986) call this decomposition a *value tree*, which consists of a number of general areas of concern, intermediate objectives and specific evaluation criteria. Given the decomposition of what a decision maker values, a utility function is constructed.

In everyday communication, people use argumentation to reason about and to discuss what decision is the best (Shafir et al., 1997). It would thus be the most natural for a (human) decision maker if he could be supported by arguing with a decision support system about what he should do. Therefore, the main research question was to investigate how argumentation can be used to support decision making in complex scenarios. Moreover, very precise choices have to be made when determining the utility function of a decision maker. For example, in firefighting the importance of the safety of victims, the safety of the personnel, the environment, and material damage all have to be quantified. It is argued, e.g., by Boutilier (1994) and Brafman and Domshlak (2009), that it is easier for people to express what they value with qualitative statements. In this thesis, we have proposed an argumentation-based framework for this purpose.

Required Concepts

Because value is a key concept in decision making, it is necessary to understand what it means to value something. Research question 1a asked what concepts are required to reason about what a decision maker values. The required concepts were used to answer the other research questions. Chapter 3 addressed this research question by proposing a conceptual framework for value in which several concepts concerning value and decisions were described and related. Furthermore, several argument schemes were proposed to reason about value.

To determine what notions are required for decision making, we have looked at the book on decision making written by Keeney and Raiffa (1976). In decision making, a *decision maker* can choose from a number of *alternatives*. An alternative is an abstract entity, which means that an alternative can be a single action or a sequence of actions. However, the decision maker can only choose one alternative, not multiple. The *outcome* of choosing an

alternative is described using a number of *attributes*. Each attribute has a number of *attribute values* that it can take. The concepts of decision maker, alternative, attribute and attribute value are thus required to reason about what a decision maker values.

Hansson (2001) notes that all value statements are done according to a criterion. In our conceptual framework, value is always seen from a *perspective*. Therefore, the conceptual framework proposed in Chapter 3 is called the Perspective-based Value Model. Following Hansson, we distinguished between *relative value* (also called dyadic value) and *monadic value*. For decision making, the outcome with the most value from the perspective of the agent is required. In complex decision scenarios decision makers often do not know what outcome to prefer, but can often identify what perspectives they care about. Inspired by Keeney (1992); Keeney and Raiffa (1976); von Winterfeldt and Edwards (1986), we proposed to decompose the decision maker's perspective into multiple more specific perspectives. These perspectives can then be further decomposed until they are specific enough to compare outcomes. In this way, the perspectives that a decision maker cares about can be structured into general areas of concern, objectives and specific evaluation criteria. To capture the influence between perspectives, we introduced a positive and a negative influence relation between perspectives. Furthermore, because the perspectives that influence a perspective may differ in importance, we introduced a relative importance notion of perspectives with respect to a perspective. A number of argument schemes were also proposed that use either the structure of outcomes or the influence relation between perspectives to reason about the relative value of outcomes from a perspective.

In the argumentation literature, researchers have proposed how the values that a decision maker holds, can be used in decision making. For example, Atkinson et al. (2006) proposed an argument scheme that justifies making a decision if that decision promotes a value the decision maker holds. To investigate how the values a person pursues relate to the PVM, we have explored the psychology literature. In the psychology literature, researchers have investigated what values are and which values are pursued by people all over the world. Schwartz (1992) found ten basic value types that people find important. Individuals differ in the degree to which they find values important. Moreover, Schwartz found correlations between how important people find particular values. The ten basic value types of Schwartz can be used in the PVM to initialize the value tree of individuals and to predict what values they find important.

The main contribution of Chapter 3 is the conceptual framework that combines techniques from decision analysis with the concept of value as discussed by Hansson. In contrast to other approaches, the perspective from which value is seen is modeled explicitly, which allowed decomposing what a decision maker values and to reason about preferences. Furthermore, the way of decomposing value from perspectives into more specific perspectives also corresponds to how value is seen in psychology literature such as Schwartz (1992). In later chapters, this conceptual framework provided to be sufficient for decision making in complex scenarios. This was demonstrated on the running example, but also on the different problem of deciding what dialogue move to make in a dialogue.

Arguing about Value

Research question 1b asked 'how can argumentation be used to reason about, justify, and refute what a decision maker values'. To answer 1b it was necessary to formalize the concep-

tual framework proposed in Chapter 3, which is done in Chapter 4. First, Section 4.1 defines the notions of attribute, attribute value and assignment, which are used to represent the outcomes of decisions. Several relations between assignments are proposed. Next, Section 4.2 formalizes the notion of perspective, influence between perspectives and relative importance of perspectives to a perspective. Finally, Section 4.3 formalizes monadic evaluations. Some example monadic evaluations are 'assignment s is good from the perspective of safety of personnel' and 'assignment t is bad from the perspective of the environment'.

Research question 1b is concerned with how argumentation can be used. To address this, Chapter 5 proposed an argumentation logic that can be used to reason about, justify and refute what a decision maker values. This argumentation framework is based on Chapter 4's formalization of the conceptual framework of Chapter 3 and consists of an object-level and meta-level argumentation system. Attributes, assignments (i.e., the outcomes of decisions) and perspectives are represented as constants in the object-level logical language. Relative value between assignments from a perspective is modeled as a ternary predicate. Two binary predicates are used to model positive and negative influence between perspectives. Using these influence predicates, the value tree of the decision maker can be represented. Next, a number of defeasible inference rules are proposed that formalize the argument schemes proposed in Chapter 3. Using the influence between perspectives, these defeasible rules can be used to construct arguments concluding what the decision maker should value more. The relative importance between perspectives is modeled as a ternary relation between perspectives in the meta-level logical language. Several meta-level defeasible rules are then proposed that use the relative importance between perspectives to reason about the relative strength of object-level arguments. The effect of the meta-level argumentation framework on the object-level arguments is determined using the mechanism proposed in Section 2.3.

In complex decision scenarios it is typically not clear how to value the outcomes of the available decisions. In other words, in complex scenarios the preferences of a decision maker are often not known. By decomposing what a decision maker values into a value tree, an abstract perspective is decomposed into a number of specific evaluation criteria. It is typically possible to compare outcomes on these criteria. The argumentation framework proposed in Chapter 5 can be used to argue about what outcome should be valued more by the decision maker. It is likely that one outcome is better than another outcome on several criteria but worse on other criteria. The accrual mechanism proposed in Section 2.2 can then be used to combine the arguments why one outcome is better than another. This means that the decision maker is faced with a tradeoff. However, because not all criteria may be as important to the decision maker, meta-level argumentation can be used to resolve such conflict.

Since related work in the literature does not allow reasoning about value, the main contribution of Chapter 5 is that it provides a natural way to reason about value by means of argumentation. This is an essential requirement for supporting decisions in complex decision scenarios.

Practical Reasoning

Practical reasoning is reasoning about what to do. Research question 1c asked how argumentation can be used for practical reasoning taking into account resource-boundedness. Chapter 5 proposed an argumentation framework to reason about the value of outcomes of possible decisions. In Chapter 6, we have addressed research question 1c by extending Chapter 5's

argumentation framework with two different notions of goals, argument schemes to justify having a goal, and argument schemes to justify decisions based on the goals they achieve.

Many approaches in the argumentation literature (e.g., Hulstijn and van der Torre (2004), Atkinson et al. (2006) and Amgoud and Prade (2009)) use *goals* to represent aspects of outcomes that are desirable to an agent in some way. Following Simon (1955), we have considered a goal w.r.t. a perspective as a simplification of that perspective by grouping together all assignments that have a certain level of value. Instead of comparing all alternatives to find the one with the most value, a decision maker can stop when an alternative is found with a satisfactory level of value. Following Amgoud and Prade (2009), we distinguished between two kinds of goals: *achievement goals* and *avoidance goals*. Achievement and avoidance goals are always seen from a perspective. It is possible that an assignment is part of an achievement goal from one perspective but part of an avoidance goal from another perspective. Achievement and avoidance goals are modeled as the ternary predicates goal and avoid respectively. Furthermore, several argument schemes were proposed and formalized to justify having a goal using the monadic evaluations of the decision maker and to justify choosing or not choosing an alternative based on the goals it achieves. Several meta-level argumentation schemes were then proposed and formalized that use the relative importance of perspectives to reason about the strength of arguments that conclude what the decision maker should do.

The main contribution of Chapter 6 is that the notion of goal is related with the more general notion of value. In this approach it is possible to justify having a certain goal using the more fundamental concept of value. This is an important contribution, which is not possible in approaches like Atkinson et al. (2006) and Amgoud and Prade (2009). Furthermore, Section 6.5.3 has explained how our approach can be used in combination with the approach of Amgoud and Prade (2009).

9.2 Supporting a Decision

By using argumentation and dialogues to support decisions it is possible to justify why the user should prefer one decision to another. On the one hand, it allows the user to respond to the arguments given by the decision support system by asking questions why something is true, but also by giving counterarguments if the user does not agree. On the other hand, it allows the decision support system to respond in the same way to what the user says. Because the participants can decide what topics they discuss, the dialogue can focus on the important parts and ignore the less important topics. This can be particularly useful in situations where there is little time available. Moreover, it is natural for people to use dialogues to discuss what they should do.

The last two research questions, 2a and 2b, focused on what is required to argue to motivate decisions in a dialogue. First, question 2a is concerned with a dialogue framework that enables communicating what is important in discussing a decision. To make a support dialogue as useful as possible, the decision support system carefully determines what dialogue move to make. Therefore, question 2b asks how a decision support system can select the best dialogue move.

Dialogues for Supporting Decisions

In order for a decision support system to participate in a dialogue it is necessary to describe precisely what can be done in the dialogue. Research question 2a asks how to formally represent a dialogue framework for arguing to motivate decisions. For arguing to motivate decisions it is required that participants of a dialogue can put forward both object-level and meta-level arguments. To accommodate for these requirements, Chapter 7 extended the dialogue framework of Prakken (2005a). More specifically, the communication language was extended to allow putting forward object-level and meta-level arguments, which enables discussing reasons for and against decisions and also discussing their relative strength. Because the communication language was extended, it was necessary to extend the reply structure, which specifies what utterance can be made in reply to another utterance. Then, a protocol was proposed that is tailored for decision support dialogues. In this protocol, participants are allowed to reply to their own moves. Furthermore, to ensure that the dialogue is coherent, different notions of relevance have been proposed that are used to regulate when participants can make what dialogue move.

In a decision support dialogue, it is important for the decision support system to learn from the utterances the user makes because it may not be known in advance what the user values. However, in a support dialogue, the user will reveal parts of what he finds important. The support system should learn from what the user expresses but should also not take every statement for granted. Section 7.3 proposed a general knowledge update function to update the system's knowledge when a dialogue move is observed. Informally, we have then described a specific update function that puts claims of the user in the assumptions of the system. In this way, the system learns from the utterances but retains a level of skepticism. Again, we have illustrated the formalism on the running example of the introduction.

The contribution of Chapter 7 is that the dialogue framework of Prakken (2005a) is extended to satisfy the requirements for support dialogues. To this end, the following changes were made. It is now possible that both object-level and meta-level arguments are exchanged, that participants can attack their own moves, and to distinguish weaker notions of relevance that are required for our purposes. Furthermore, we have indicated how the knowledge of the decision support system can be updated given new utterances of the decision maker.

Move Selection with Multiple Criteria

In complex decision scenarios a decision maker may care about a large number of different perspectives and criteria. This means that two decisions can be compared from many different perspectives. On the other hand, there may be many alternatives from which the decision maker can choose. In a support dialogue it is thus very likely that the decision support agent must choose from a possibly large number of dialogue moves that he can make.

Because the time may be limited to support the decision maker, it is important that the support agent selects the most effective dialogue move. However, to make a good decision, the decision maker needs to consider all perspectives that matter in the decision. This means that the support agent should aim to be comprehensive in the sense that if an important perspective is not discussed, then he should discuss this. Efficiency and comprehensiveness are just two perspectives that may influence what moves the support agent should select, but it is not hard to think of other perspectives that should be taken into account. Because the deci-

sion support agent cares about different perspectives in selecting what move to make, move selection can be seen as multi-criteria decision making.

The final research question 2b is concerned with how to reason about what dialogue move to make in a dialogue. Chapter 5 proposed an argumentation framework that is designed for decision making where multiple criteria matter. Given the observation that move selection is a decision making problem where multiple criteria matter, Chapter 8 demonstrated that the argumentation framework of Chapter 5 can be used to reason about what dialogue move to make in a dialogue, which answers research question 2b. The contribution of Chapter 8 is therefore twofold. Firstly, it demonstrated that the practical reasoning approach in this thesis is general enough to capture a different kind of decision making problem. Secondly, it demonstrated that dialogue move selection can be seen as a normal decision making problem and does not have to be treated in a special way as done in Amgoud and Hameurlain (2007).

9.3 Contributions

Although this chapter already discussed the contributions of this thesis, we would like to highlight the five contributions that we find most important. Firstly, in our approach a decision support agent can take the role of a decision analyst by supporting the decision maker in determining his value tree. Namely, the support agent can question why something is important, give counterarguments why something should be or should not be important, propose new aspects to consider and justify why these aspects should be considered.

Secondly, in our argumentation framework it is possible to state that perspectives are important without specifying exactly how important. This is especially important when perspectives matter that are different by nature. For example, in a firefighting scenario, a decision maker can express that both the safety of the public and minimizing material damage are important and that safety is more important than damage, but he may not be able to express how much more safety is. Given such an imprecise understanding of the decision maker's value tree, the support agent can still reason about and put forward arguments that justify why a certain decision should be taken.

Thirdly, the support agent does not need a complete understanding of what matters to the decision maker in order to support a decision. Namely, the support agent can ask the decision maker why something is important if necessary and also propose new perspectives to take into account. If the decision maker does not agree, then he can attack the support agent's arguments. By doing so, the decision maker reveals more about his value tree at the time that this is necessary. In this way, we hope that decision support is also useful in time-critical decision scenarios.

The fourth contribution concerns decomposing abstract perspectives. Like Atkinson et al. (2006), this thesis recognizes the necessity to consider the decision maker's abstract values in practical reasoning. However, a more comprehensive account of abstract values is given and grounded within research in psychology. An important advantage of this more comprehensive account is that it allows discussing what an abstract value means to a particular agent. When heterogeneous agents argue to motivate decisions, it cannot be assumed that all agents agree on what a value means. For example, what constitutes values like fun or fairness is subjective and although two agents may agree on what safety is, there may be disagreement about what is the best way to measure safety. Moreover, agents may forget to consider impor-

tant aspects of a value. Our approach of decomposing abstract perspectives into more specific perspectives using the influence relation is inspired by techniques in decision analysis (e.g., Keeney (1992)) and allows agents to discuss what constitutes a value for them and to discuss what outcome is the best from the perspective of a value. These are essential requirements for arguing to motivate decisions, especially in complex decision scenarios.

Lastly, the conceptual framework proposed in Chapter 3 uses qualitative statements to express value. As noted in different papers (e.g., Boutilier (1994) and Brafman and Domshlak (2009)), it is cognitively easier for users to reflect upon and express value in a qualitative manner rather than a quantitative manner. Moreover, qualitative value statements occur frequently in daily conversations. From the viewpoint of decision theory, we think it is therefore a contribution that we have proposed a qualitative version of several techniques in decision analysis.

9.4 Recommendations for Future Work

There are many ways how the proposed formalism of this thesis can be extended. In this section, we will suggest a number of possible extensions that we think are the most valuable. First of all, to discover whether our approach is useful for supporting a decision, it is necessary to test whether human subjects can express themselves adequately using the conceptual framework that was proposed. To test this, our approach needs to be implemented and evaluated empirically on different decision scenarios. A major concern is the translation between our symbolic representation of arguments and a representation that human subjects can understand such as natural language. If an *ad hoc* approach for this translation is used, then it will not be clear whether bad performance is caused by the conceptual framework or by the translation. Nevertheless, if such an experiment could be conducted successfully, then much can be learned about the strengths and shortcomings of our conceptual framework. Another valuable way to evaluate our approach is to do a formal evaluation of what properties hold. Such investigation is particularly interesting on the later chapters of this thesis.

Perspective-based Value

We think that the following extensions of the perspective-based value model would be the most useful. Instead of only using orderings to represent perspectives, the model would benefit from extending it with also representing perspectives in a quantitative manner. This would not only allow to express that one assignment is better than another from some perspective, but also *how much* better. Influence between two quantitatively represented perspectives could then be defined more precisely. Namely, if perspective p influences perspective q and assignment s is an amount of x better than assignment t from perspective p, then s is $f(x)$ better than t from perspective q. We would like to stress that it is not desirable to represent all perspectives quantitatively. For example, representing the perspective of beauty quantitatively requires that agents give precise judgments of beauty, which they cannot. We recommend extending the perspective-based value model with numerical strengths of influence relations.

Often it is possible to use different criteria to measure an objective. In this case, it is useful to discuss what criterion perspective is the most suitable to measure the value from

an objective or general area of concern. This thesis proposed several argument schemes to justify influence between perspectives, but these cannot be used to justify that one criterion perspective is more suitable than another. Therefore, we recommend investigating argument schemes concerning the suitability of criteria.

When a perspective p is influenced by multiple perspectives, then it is possible that some arguments can be constructed concluding that $s <_p t$ and other arguments conclude that $t <_p s$. Using the accrual mechanism of Section 2.2, it is possible to accrue these arguments. This results in two accrual arguments: one concluding $s <_p t$ and another concluding $t <_p s$. To determine what accrual argument is justified, we need to know which accrual argument is stronger. We recommend extending our formalism with rules to determine the relative strength of accrual arguments.

Practical Reasoning

Often there is uncertainty w.r.t. in what outcome performing an alternative results. This thesis only distinguishes between certainly and possibly resulting in an outcome. It is thus not possible to distinguish between a very likely outcome and a very unlikely outcome. This means that an alternative achieving a goal very likely is an equally strong reason as an alternative achieving a goal only possibly. Our formalism would therefore benefit from extending it with a more comprehensive account of uncertainty. One possible approach that seems suitable for this purpose is the use of Qualitative Probabilistic Networks as proposed by Wellman (1990). Several extensions have been proposed that use a finer grained qualitative statements to express probabilities (e.g., see Renooij (2001) and Renooij and van der Gaag (2008)).

Mental State Abduction

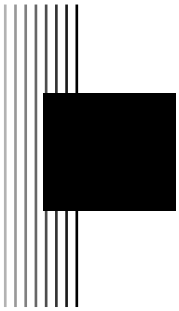
Because the support agent may not know anything about the decision maker when a decision support dialogue is started, it is important that the agent learns as much as possible about what the decision maker believes and values. Mental state abduction is abducing an agent's mental state from observations that are made. The dialogue moves the decision maker makes can reveal much about the decision maker's mental state. It would therefore be interesting to investigate further how the support agent could do accurate mental state abduction.

The claims that the user makes do not necessarily say anything about his mental state, i.e., that the user claims ϕ is true does not mean that the user believes that ϕ is true. However, in general users do believe what they claim. Mental state abduction approaches such as Dragoni et al. (2002); Sindlar et al. (2011) could be used to form hypotheses about what other agents believe and what goals they have. For example, Dragoni et al. (2002) use a plan-based vision on speech acts, where each speech act has a number of preconditions and postconditions expressed in terms of beliefs. From the observed speech act an agent abduces what beliefs the speaker had at the time of uttering the speech act. Suppose that speech act s claims that ϕ is true and that the precondition of s is that ϕ is a belief of the agent and that its postcondition is that the hearer believes ϕ. If you observe agent α uttering s to agent β, then you can abduce that α believes that ϕ is true, that α had the intention that β believes ϕ is true, and that β now believes ϕ. From this you could reason further about the mental state of the agents. For example, if α believes ϕ, then he must also believe ψ and if α had the intention to make β believe ϕ, then α must have had a certain goal.

Although claims do not necessarily say anything about the mental state of an agent, in general agents do believe to what they claim. The following argument scheme describes this intuition informally: *agent α claims formula ϕ is true, therefore, presumably, α believes that ϕ is true*. If the binary predicates $\mathsf{bel}(\alpha, \phi)$ and $\mathsf{claim}(\alpha, \phi)$ are added to the meta-language denoting that agent α believes / claims that ϕ is true respectively, then this argument scheme can be formalized as the defeasible rule $d_{\mathsf{cl2bel}}(\alpha, \phi) : \mathsf{claim}(\alpha, \phi) \Rightarrow \mathsf{bel}(\alpha, \phi)$. Critical questions for this argument scheme could question whether the agent has lied and whether the agent only has claimed this for the sake of the argument. By representing on the meta-level what agents believe, it is possible to extend our framework with epistemic approaches like the ones described in Meyer and Van Der Hoek (2004) to further reason about what agents may believe. The better a support agent is at mental state abduction, the more he learns from the utterances of other agents and the more easily he can detect wrong or missing information. Therefore we think that mental state abduction is important for applications like decision support.

Implementation

To answer our research questions, we have developed an argumentation and dialogue framework, which combine a number of techniques from different disciplines. A next step could be to implement this framework, which allows evaluating our approach with users. If the implementation is used in a practical setting instead of an experimental setting, then special attention should be given to situations in which there is a limited amount of time to make a decision. In such situations, the usability of the decision support system depends on how fast useful counterarguments and questions can be given. To respond within a limited time, it may not be possible to construct and evaluate all possible arguments to make a decision. This means that techniques are required that select a subset of all arguments to construct and evaluate in a given amount of time. For example, if no alternatives result in assignments s and t, then it is still possible to construct an argument comparing the value of s and t from some perspective. In this case, an argument comparing s and t is not important for the decision and does not need to be constructed and evaluated. The approach in Chapter 6 already provides a way to focus on certain kinds of arguments, but more techniques are necessary. Finally, we hope that this thesis will inspire new research on this topic and that it will be used in an actual decision support system so that one day decision makers will benefit from these techniques.



References

Allport, G. and Willard, G. (1961). *Pattern and growth in personality*. Holt, Rinehart and Winston New York.

Altemeyer, B. and Smith, M. (1988). *Enemies of freedom: Understanding right-wing authoritarianism*. Jossey-Bass San Francisco.

Amgoud, L. (2005). A unified setting for inference and decision: An argumentation-based approach. *Proceedings of the IJCAI-2005 Workshop on Computational Models of Natural Argument*, (pp. 40–43).

Amgoud, L., Bodenstaff, L., Caminada, M., McBurney, P., Parsons, S., Prakken, H., van Veenen, J., and Vreeswijk, G. (2006). Final review and report on formal argumentation system. deliverable d2. 6. Tech. Rep. ASPIC IST-FP6-002307.

Amgoud, L., Bonnefon, J., and Prade, H. (2005). An argumentation-based approach to multiple criteria decision. *8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU*.

Amgoud, L. and Hameurlain, N. (2007). An argumentation-based approach for dialog move selection. *Argumentation in multi-agent systems*, (pp. 128–141).

Amgoud, L. and Prade, H. (2006). Explaining qualitative decision under uncertainty by argumentation. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21, (pp. 219–224).

Amgoud, L. and Prade, H. (2009). Using arguments for making and explaining decisions. *Artificial Intelligence*, *173*(3-4), 413 – 436.

Amgoud, L. and de Saint Cyr, F. (2008). Measures for persuasion dialogs: A preliminary investigation. In *Proceeding of the 2008 conference on Computational Models of Argument*, (pp. 13–24).

Amgoud, L. and de Saint-Cyr, F. (2009). Extracting the core of a persuasion dialog to evaluate its quality. In *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, (pp. 59–70). Springer.

Atkinson, K. and Bench-Capon, T. (2007a). Action-based alternating transition systems for arguments about action. In *Proceedings of the Twenty Second Conference on Artificial Intelligence (AAAI 2007), Vancouver, Canada*, Vol. 22, (pp. 24–29). AAAI Press, Menlo Park, CA, USA.

Atkinson, K. and Bench-Capon, T. (2007b). Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, *171*(10-15), 855–874.

Atkinson, K., Bench-Capon, T., and McBurney, P. (2006). Computational representation of practical argument. *Synthese*, *152*(2), 157–206.

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

Bench-Capon, T. and Atkinson, K. (2009). Action-state semantics for practical reasoning. In A. T. R. SS-09-06 (Ed.), *The Uses of Computational Argumentation: Papers from the AAAI Fall Symposium (FS-09-06)*, (pp. 8–13). AAAI Press.

Bench-Capon, T., Atkinson, K., and McBurney, P. (2009). Altruism and agents: an argumentation based approach to designing agent decision mechanisms. In S. Decker, Sichman and Castelfranchi (Eds.), *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Vol. 2, (pp. 1073–1080).

Bench-Capon, T. and Prakken, H. (2006). Justifying actions by accruing arguments. In P. Dunne and T. Bench-Capon (Eds.), *Computational Models of Argument. Proceedings of COMMA*, (pp. 247–258).

Bench-Capon, T., Prakken, H., and Visser, W. (2011). Argument schemes for two-phase democratic deliberation. In *Proceedings of the 13th International Conference on Artificial Intelligence and Law*. New York: ACM Press 2011.

Bharosa, N., Lee, Y., and Janssen, M. (2010). Challenges and obstacles in information sharing and coordination during multi-agency disaster response: Propositions from field exercises. *Information Systems Frontiers*, *12*(1), 49–65.

Black, E. and Atkinson, K. (2011). Choosing persuasive arguments for action. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS11, Taipei, Taiwan)*.

Black, E. and Hunter, A. (2009). A Relevance-theoretic Framework for Constructing and Deconstructing Enthymemes. *Journal of Logic and Computation*.

Bondarenko, A., Dung, P., Kowalski, R., and Toni, F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial intelligence*, *93*(1-2), 63–101.

Bonnefon, J., Dubois, D., Fargier, H., and Leblois, S. (2008). Qualitative heuristics for balancing the pros and cons. *Theory and Decision*, *65*(1), 71–95.

Bonnefon, J. and Fargier, H. (2006). Comparing sets of positive and negative arguments: Empirical assessment of seven qualitative rules. In *ECAI 2006: 17th European Conference on Artificial Intelligence*, (p. 16). Ios Pr Inc.

Boutilier, C. (1994). Toward a logic for qualitative decision theory. *Proceedings of the KR*, *94*, 75–86.

Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., and Poole, D. (2004). Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, *21*, 135–191.

Brafman, R. and Domshlak, C. (2009). Preference handling - an introductory tutorial. *AI Magazine*, *30*(1), 58–86.

Bratman, M., Israel, D., and Pollack, M. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence*, *4*(4), 349–355.

Cacioppo, J., Gardner, W., and Berntson, G. (1997). Beyond bipolar conceptualizations and measures: The case of attitudes and evaluative space. *Personality and Social Psychology Review*, *1*(1), 3.

Caminada, M. and Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artificial Intelligence*, *171*(5-6), 286–310.

Cayrol, C., de Saint-Cyr, F., and Lagasquie-Schiex, M. (2008). Revision of an argumentation system. In *11th International Conference on Principles of Knowledge Representation and Reasoning (KR08)*, (pp. 124–134).

Doyle, J. and Thomason, R. (1999). Background to qualitative decision theory. *AI MAG*, *20*(2), 55–68.

Dragoni, A., Giorgini, P., and Serafini, L. (2002). Mental states recognition from communication. *Journal of Logic and Computation*, *12*(1), 119.

Dubois, D., Fargier, H., and Bonnefon, J. (2008). On the qualitative comparison of decisions having positive and negative features. *Journal of Artificial Intelligence Research*, *32*, 385–417.

Dung, P. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*(2), 321–358.

Edwards, W. (1977). How to use multiattribute utility measurement for social decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*(5), 326–340.

Falappa, M., Kern-Isberner, G., and Simari, G. (2009). Belief revision and argumentation theory. *Argumentation in artificial intelligence*, (pp. 341–360).

Fishburn, P. (1970). *Utility theory for decision making*. Storming Media.

Genesereth, M. and Nilsson, N. (1987). *Logical foundations of artificial intelligence*. Kaufmann.

Grasso, F., Cawsey, A., and Jones, R. (2000). Dialectical argumentation to solve conflicts in advice giving: A case study in the promotion of healthy nutrition. *International Journal of Human-Computers Studies*, *53*(6), 1077–1115.

Hansson, S. (2001). *The structure of values and norms*. Cambridge University Press.

Hulstijn, J. and van der Torre, L. (2004). Combining goal generation and planning in an argumentation framework. *Proc. Workshop on Argument, Dialogue and Decision, at NMR, Whistler, Canada, June*.

Hunter, A. (2004a). Making argumentation more believable. *Proc. of AAAI04*, (pp. 269–274).

Hunter, A. (2004b). Towards higher impact argumentation. *Proc. of the 19th American National Conf. on Artificial Intelligence (AAAI 2004), MIT Press*, (pp. 275–280).

Hunter, A. (2007). Real arguments are approximate arguments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'07)*, Vol. 22, (pp. 66–71). MIT Press.

Hunter, A. (2008). Reasoning about the appropriateness of proponents for arguments. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*, Vol. 8, (pp. 89–94). MIT Press.

Johnson, W., Vilhjalmsson, H., and Marsella, S. (2005). Serious games for language learning: How much game, how much AI? In *Proceeding of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, (pp. 306–313). IOS Press.

Jung, C. (1921). *Psychological Types*.

Kahneman, D. and Snell, J. (1990). Predicting utility. *Insights in decision making*, (pp. 295–310).

Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–292.

Keeney, R. (1992). *Value-Focused Thinking: A Path to Creative Decisionmaking*. Harvard University Press.

Keeney, R. and Raiffa, H. (1976). *Decisions with Multiple Objectives*. Wiley, New York.

Kok, E., Meyer, J.-J. C., and Prakken, G. A. W., H. Vreeswijk (2010). A formal argumentation framework for deliberation dialogues. In *Seventh International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*.

Lewin, K. (1951). Constructs in field theory. In D. Cartwright (Ed.), *Field theory in social science: selected theoretical papers*. Harpers.

Luce, R. (1956). Semiorders and a theory of utility discrimination. *Econometrica, Journal of the Econometric Society*, *24*(2), 178–191.

March, J. G. (1978). Bounded rationality, ambiguity, and the engineering of choice. *The Bell Journal of Economics*, *9*(2), 587–608.

McBurney, P., Hitchcock, D., and Parsons, S. (2007). The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, *22*(1), 95–132.

McBurney, P. and Parsons, S. (2002). Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, *11*(3), 315–334.

McBurney, P. and Parsons, S. (2009). *Argumentation in Artificial Intelligence*, chap. Dialogue games for agent argumentation, (pp. 261–280). Springer Science.

Medellin-Gasque, R., Atkinson, K., McBurney, P., and Bench-Capon, T. (2011). Arguments over co-operative plans. In *Proceedings of the First International Workshop on Theory and Applications of Formal Argumentation (TAFA 2011), Barcelona, Spain*.

Meyer, J.-J. C. and Van Der Hoek, W. (2004). *Epistemic logic for AI and computer science*. Cambridge Univ Pr.

Michael, D. and Chen, S. (2005). *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade.

Modgil, S. (2007). An abstract theory of argumentation that accommodates defeasible reasoning about preferences. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, (pp. 648–659).

Modgil, S. (2009). Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, *173*(9-10), 901 – 934.

Modgil, S. and Prakken, H. (2011). Revisiting preferences and argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*.

von Neumann, J., Morgenstern, O., Kuhn, H., and Rubinstein, A. (1947). *Theory of games and economic behavior*. Princeton University Press.

Oren, N., Norman, T., and Preece, A. (2007). Information based argumentation heuristics. *Argumentation in Multi-Agent Systems*, (pp. 161–174).

Paglieri, F. and Castelfranchi, C. (2005). Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in mas. *Proceedings of the 1 stworkshop on Argumentation in MAS (ArgMAS 2004)*.

Parsons, S., McBurney, P., Sklar, E., and Wooldridge, M. (2007). On the relevance of utterances in formal inter-agent dialogues. In *Proceedings of the 4th international conference on Argumentation in multi-agent systems*, (pp. 47–62). Springer-Verlag.

Payne, J., Bettman, J., and Johnson, E. (1992). Behavioral decision research: A constructive processing perspective. *Annual Review of Psychology*, *43*(1), 87–131.

Perelman, C. and Olbrechts-Tyteca, L. (1969). *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press.

Perlis, D. (1985). Languages with self-reference I: Foundations. *Artificial Intelligence*, *25*(3), 301–322.

Pollock, J. (1987). Defeasible reasoning. *Cognitive Science*, *11*, 481518.

Pollock, J. (1998). The logical foundations of goal-regression planning in autonomous agents. *Artificial Intelligence*, *106*(2), 267–334.

Prakken, H. (2000). On dialogue systems with speech acts, arguments, and counterarguments. *Proc. of the 7th European Workshop on Logic for Artificial Intelligence (JELIA2000)*, (pp. 224–238).

Prakken, H. (2005a). Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, *15*(6), 1009.

Prakken, H. (2005b). A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th International Conference on A.I. and Law*, (pp. 85–94). ACM NY, USA.

Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, *21*(02), 163–188.

Prakken, H. (2010). An abstract framework for argumentation with structured arguments. *Argument and Computation*, *1*(2), 93–124.

Raz, J. (1978). *Practical reasoning*. Oxford University Press.

Renooij, S. (2001). *Qualitative approaches to quantifying probabilistic networks*. Ph.D. thesis, Universiteit Utrecht.

Renooij, S. and van der Gaag, L. C. (2008). Enhanced qualitative probabilistic networks for resolving trade-offs. *Artificial Intelligence*, *172*(12-13), 1470 – 1494.

Riveret, R., Prakken, H., Rotolo, A., and Sartor, G. (2008). Heuristics in argumentation: A game-theoretical investigation. In P. Besnard, S. Doutre, and A. Hunter (Eds.), *Computational Models of Argument. Proceedings of COMMA 2008*, (pp. 324–335). IOS Press.

Rohan, M. (2000). A rose by any name? the values construct. *Personality and Social Psychology Review*, *4*(3), 255–277.

Rokeach, M. (1973). *The nature of human values*. Free Press, New York.

Saaty, T. (1986). Axiomatic foundation of the analytic hierarchy process. *Management Science*, *32*(7), 841–855.

Saaty, T. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83–98.

Savage, L. (1954). *The foundations of statistics*. Wiley, New York.

Schurr, N., Marecki, J., Lewis, J., Tambe, M., and Scerri, P. (2005). The defacto system: Training tool for incident commanders. In M. Veloso and S. Kambhampati (Eds.), *AAAI*, (pp. 1555–1562). AAAI Press / The MIT Press.

Schwartz, S. (1992). Universals in the content and structure of values: theoretical advances and empirical tests in 20 countries. *Advances in experimental social psychology*, *25*, 1–65.

Schwartz, S. and Bilsky, W. (1987). Toward a universal psychological structure of human values. *Journal of Personality and Social Psychology*, *53*(3), 550–562.

Searle, J. (2001). *Rationality in Action*. Bradford Books.

Shafir, E., Simonson, I., and Tversky, A. (1997). Reason-based choice. *Research on judgment and decision making: currents, connections, and controversies*, (p. 69).

Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics*, (pp. 99–118).

Simon, H. A. (1957). *Models of man*. John Wiley.

Sindlar, M., Dastani, M., and Meyer, J.-J. (2011). Programming mental state abduction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Steedman, I. and Krause, U. (1986). Goethes faust, arrows possibility theorem and the individual decision-taker. *The Multiple Self*, (pp. 197–231).

Tonolio, A., Norman, T., and Sycara, K. (2011). Argumentation schemes for policy-driven planning. In *Proceedings of the First International Workshop on Theory and Applications of Formal Argumentation (TAFA 2011), Barcelona, Spain*.

Turner, R. (1990). Truth and modality for knowledge representation. *MIT Press Series Of Artificial Intelligence Series*.

Verheij, B. (2003). Dialectical argumentation with argumentation schemes: An approach to legal logic. *Artificial intelligence and Law*, *11*(2), 167–195.

Vreeswijk, G. (1997). Abstract argumentation systems. *Artificial Intelligence*, *90*(1-2), 225–279.

Walton, D. (1996). *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates.

Walton, D. and Krabbe, E. (1995a). *Commitment in dialogue: Basic concepts of interpersonal reasoning*. State University of New York Press.

Walton, D. and Krabbe, E. (1995b). *Commitment in dialogue: Basic concepts of interpersonal reasoning*. State University of New York Press.

van der Weide, T., Dignum, F., Meyer, J.-J. C., Prakken, H., and Vreeswijk, G. (2009a). Personality-based practical reasoning. In I. Rahwan and P. Moraitis (Eds.), *Argumentation in Multi-Agent Systems: Fifth International Workshop, ArgMAS 2008, Estoril, Portugal, May 2008 Revised Selected and Invited Papers*, Vol. 5384/2009 of *Lecture Notes in Computer Science*, (pp. 3–18). Springer Berlin / Heidelberg.

van der Weide, T., Dignum, F., Meyer, J.-J. C., Prakken, H., and Vreeswijk, G. A. W. (2010). Arguing about preferences and decisions. In *Proc. of the 7th Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*.

van der Weide, T. L. and Dignum, F. (2011). Reasoning about and discussing preferences between arguments. In P. McBurney, S. Parsons, and I. Rahwan (Eds.), *Proceedings of the Eight International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2011), Taipei, Taiwan*.

van der Weide, T. L., Dignum, F., Meyer, J.-J. C., and Prakken, G. A. W., H. Vreeswijk (2009b). Argumentation about motivation. In E. Rondeel (Ed.), *Human Factors Event 2009*.

van der Weide, T. L., Dignum, F., Meyer, J.-J. C., and Prakken, G. A. W., H. Vreeswijk (2009c). Practical reasoning using values. In P. McBurney, I. Rahwan, S. Parsons, and P. Moraitis (Eds.), *Proceedings of the Sixth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009), Budapest, Hungary*, (pp. 225–240).

van der Weide, T. L., Dignum, F., Meyer, J.-J. C., Prakken, H., and Vreeswijk, G. (2011). Multi-criteria argument selection in persuasion dialogues. In S. Yolum, Turner and Sonenberg (Eds.), *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*.

Wellman, M. and Doyle, J. (1991). Preferential semantics for goals. *Proceedings of the National Conference on Artificial Intelligence*, (pp. 698–703).

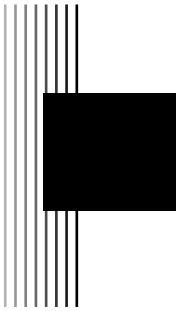
Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, *44*(3), 257 – 303.

von Winterfeldt, D. and Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge University Press.

Wooldridge, M. (2000). *Reasoning about rational agents*. The MIT Press.

Wooldridge, M., McBurney, P., and Parsons, S. (2005). On the meta-logic of arguments. In *Argumentation in Multi-Agent Systems 2005*, Vol. 4049/2006 of *LNCS*, (pp. 42–56). Springer Berlin / Heidelberg.

Zeisset, C. (2006). *The art of dialogue*. Center for Applications of Psychological Type, Ince.



Samenvatting

Beslissen is lastig wanneer veel verschillende aspecten belangrijk zijn. Uitgaande van het ene aspect is de ene beslissing wellicht beter, terwijl diezelfde beslissing in het licht van een ander aspect juist slechter is. Zulke beslissingsproblemen treden op in allerlei situaties: van een consument die een nieuwe camera wil kopen tot een bevelhebber bij de brandweer die moet bepalen of hij zijn personeel met gevaar voor eigen leven slachtoffers gaat laten redden, of dat hij hen de brand eerst volledig laat blussen alvorens het gebouw binnen te gaan. Argumentatie speelt een belangrijke rol bij het motiveren van beslissingen. Wanneer een persoon nadenkt over een beslissing, beschouwt hij argumenten voor en tegen de verschillende opties. Argumenten worden ook gebruikt wanneer een persoon zijn beslissing bespreekt met iemand anders en om beslissingen achteraf te onderbouwen.

De onderzoeksvraag waarop ik mij in mijn proefschrift heb gericht, is hoe een computer argumentatie kan gebruiken om een menselijke gebruiker te ondersteunen bij het maken van complexe beslissingen. Om een computer dit te kunnen laten doen, moeten minstens de volgende twee deelvragen beantwoord worden: (1) hoe kan een computer redeneren over waarom een gebruiker de ene beslissing beter zou moeten vinden dan de andere, en (2) hoe kan de computer op een natuurlijke manier met de gebruiker discussiëren. Mijn proefschrift bestaat uit twee delen, waarin ik oplossingen voorstel om deze twee vragen te beantwoorden.

Het belangrijkste deel van mijn proefschrift gaat over de eerste deelvraag. In gerelateerd onderzoek worden argumentatiemodellen beschreven waarin beslissingen worden onderbouwd met behulp van de waarden die mensen belangrijk vinden en de doelen die ze willen bereiken. Aangezien waarden vaak abstract zijn, is het belangrijk dat een persoon uitlegt wat een waarde voor hem betekent. Daarbij vergeet hij mogelijk belangrijke aspecten. Door middel van een discussie kan de computer niet alleen nieuwe aspecten voorstellen, maar ook tegenargumenten geven als de gebruiker een redeneerfout lijkt te maken. De waardemodellen in de literatuur gaan uit van een statische motivatie, waardoor het niet mogelijk is te discussiëren over waar de gebruiker waarde aan hecht. Om hier wel in te slagen, heb ik onderzocht hoe bestaande modellen moeten worden aangepast.

De voornaamste bijdrage van mijn proefschrift is mijn waardemodel. Dit model is gebaseerd op modellen uit de psychologie, besliskunde en argumentatie. Met het waardemodel is het mogelijk om te argumenteren over wat een waarde voor iemand betekent en wat de beste criteria zijn om dat te meten. Tevens wordt de relatie tussen waarden en doelen expliciet gemaakt, waardoor een persoon vanuit zijn waarden een bepaald doel kan onderbouwen. Op

deze manier kunnen argumenten voor en tegen beslissingen worden geconstrueerd. Niet elk argument is echter even belangrijk. Daarom is het in mijn waardemodel ook mogelijk om op een metaniveau te argumenteren over de sterktes van argumenten.

Om een persoon te ondersteunen bij het maken van een beslissing moet de computer op een natuurlijke manier argumenten kunnen uitwisselen. Een dialoog is hiervoor uitermate geschikt. In de literatuur zijn verschillende dialoogsystemen bedacht waarin argumenten op een eerlijke en effectieve manier kunnen worden uitgewisseld. In dit proefschrift heb ik een bestaand dialoogsysteem verfijnd om het te kunnen gebruiken voor het ondersteunen van beslissingen. Een belangrijke uitbreiding is dat nu ook metaniveau-argumenten kunnen worden uitgewisseld.

In een dialoog zal de computer vaak moet kiezen uit een (groot) aantal mogelijke uitspraken. Om de dialoog zo goed mogelijk te laten verlopen, moet de computer de beste uitspraak uitkiezen. Meestal zijn er meerdere aspecten die bepalen wat de beste uitspraak is. Enerzijds moet de computer bijvoorbeeld geen oninteressante argumenten aandragen, maar van de andere kant moet hij wel zorgvuldig zijn. Het kiezen van een uitspraak is dus een beslissingsprobleem. In het laatste hoofdstuk laat ik zien dat mijn waardemodel toegepast kan worden op dit beslissingsprobleem. Hiermee draag ik enerzijds een oplossing aan voor dit probleem en test ik anderzijds de toepasbaarheid van mijn model.

Dankwoord

In de ruim vier jaar die ik bezig ben geweest met mijn promotieonderzoek hebben veel mensen een belangrijke rol gespeeld. Ze verdienen allemaal een uitgebreid dankwoord, maar daarvoor is hier geen plek. Toch wil ik een aantal mensen hier in de schijnwerpers zetten.

Allereerst wil ik mijn begeleiders Frank, Henry, Gerard en John-Jules bedanken voor de kans die ze mij hebben gegeven. Elke begeleider heeft zijn eigen kwaliteiten en ik heb van iedereen veel geleerd. Met Frank heb ik altijd met veel plezier gebrainstormd. We zaten vaak op één lijn en hadden erg interessante ideeën. Hierbij was ik altijd onder de indruk van zijn inzichten. Ook heb ik veel geleerd van en gehad aan zijn organisatie-instincten. Bij Henry heb ik altijd veel bewondering gehad voor zijn zorgvuldigheid, voorbereiding en meedenken. Ik heb bijzonder veel profijt gehad van zijn commentaar en daar ben ik hem zeer dankbaar voor. Tevens ben ik erg te spreken over Gerard als begeleider. Zijn kunde in de wiskunde heeft mijn werk sterk verbeterd en hij heeft me regelmatig met allerlei dingen geholpen. Tevens heb ik altijd het erg naar m'n zin gehad tijdens de gesprekken met hem in de pauzes. Verder ben ik John-Jules erg dankbaar voor al zijn inzichten, supervisie en de altijd gezellige gesprekken.

Ik wil mijn ouders bedanken voor het altijd klaar staan wanneer ik iets wilde bespreken. In het bijzonder wil ik mijn vader Theo bedanken omdat ik enorm veel heb gehad aan zijn adviezen zowel voor als tijdens mijn promotie. Ook wil Roel prominent bedanken aangezien de gesprekken met hem mijn verlangen om te promoveren hebben aangewakkerd.

Mijn collegae in Utrecht verdienen ook een speciale plek aangezien ik dankzij hen elke dag met veel plezier naar m'n werk ben gegaan. Allereerst wil ik de 'oudere' mede-aio's bedanken: Susan, Nieske, Joost W., Maaike, Paolo, Michal, Hado, Nick, Bob, Chris en Bas. Tevens wil ik de 'nieuwere' mede-aio's bedanken: Eric, Joost van O., Liz, Marieke, Loïs, Max, Gennaro, Sofia, Amanda en Christian. De andere collegae waarmee ik veel gezelligheid heb gedeeld: Mehdi, Virginia, Jan, Davide, Jurriaan en Huib. Sim wil bedanken voor de altijd leuke gesprekken en de vele kennis die ik van hem heb opgedaan. Verder wil ik Linda van der Gaag hartelijk bedanken voor haar gastvrijheid.

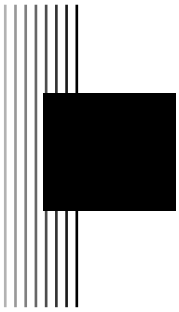
In mijn derde jaar ben ik m'n vriendin Janneke tegengekomen en sindsdien heb ik enorm veel leuke dingen met haar beleefd. Ze heeft mijn leven veel kleur gegeven. Ook heeft ze me altijd geholpen zoals op het eind met de fotoshoot van de beruchte wortel en met de Nederlandse teksten. Lieverd, ontzettend bedankt!

Uiteraard ga ik ook mijn overige vrienden bedanken. Mijn vrienden uit Amsterdam wil

ik graag bedanken voor alle verrijking en leuke herinneringen: dank je Tara, Gioia, Lynda, Moniek en Dorien! Uit Nijmegen wil ik Fabian, Justus en Bart bedanken voor de altijd gezellige momenten. En uit Utrecht bedank ik Hana, Nanneke, Eric en Engracia hartelijk voor de dagelijkse vriendschap.

Ik wil graag afsluiten met het bedanken van mijn leescommissie voor de inspanning die ze hebben geleverd. Bedankt Peter McBurney, Anthony Hunter, Katie Atkinson, Jaap van den Herik en Yao-hua Tan.

Utrecht, *Tom van der Weide*
5 september, 2011



SIKS Dissertation Series

1998

1998-1 | **Johan van den Akker** (CWI), DEGAS - An Active, Temporal Database of Autonomous Objects.

1998-2 | **Floris Wiesman** (UM), Information Retrieval by Graphically Browsing Meta-Information.

1998-3 | **Ans Steuten** (TUD), A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective.

1998-4 | **Dennis Breuker** (UM), Memory versus Search in Games.

1998-5 | **E.W. Oskamp** (RUL), Computerondersteuning bij Straftoemeting.

1999

1999-1 | **Mark Sloof** (VU), Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products.

1999-2 | **Rob Potharst** (EUR), Classification using decision trees and neural nets.

1999-3 | **Don Beal** (UM), The Nature of Minimax Search.

1999-4 | **Jacques Penders** (UM), The practical Art of Moving Physical Objects.

1999-5 | **Aldo de Moor** (KUB), Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems.

1999-6 | **Niek J.E. Wijngaards** (VU), Re-design of compositional systems.

1999-7 | **David Spelt** (UT), Verification support for object database design.

1999-8 | **Jacques H.J. Lenting** (UM), Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

2000

2000-1 | **Frank Niessink** (VU), Perspectives on Improving Software Maintenance.

2000-2 | **Koen Holtman** (TUE), Prototyping of CMS Storage Management.

2000-3 | **Carolien M.T. Metselaar** (UVA), Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 | **Geert de Haan** (VU), ETAG, A Formal Model of Competence Knowledge for User Interface Design.

2000-5 | **Ruud van der Pol** (UM), Knowledge-based Query Formulation in Information Retrieval.

2000-6 | **Rogier van Eijk** (UU), Programming Languages for Agent Communication.

2000-7 | **Niels Peek** (UU), Decision-theoretic Planning of Clinical Patient Management.

2000-8 | **Veerle Coup** (EUR), Sensitivity Analyis of Decision-Theoretic Networks.

2000-9 | **Florian Waas** (CWI), Principles of Probabilistic Query Optimization.

2000-10 | **Niels Nes** (CWI), Image Database Management System Design Considerations, Algorithms and Architecture.

2000-11 | **Jonas Karlsson** (CWI), Scalable Distributed Data Structures for Database Management.

2001

2001-1 | **Silja Renooij** (UU), Qualitative Approaches to Quantifying Probabilistic Networks.

2001-2 | **Koen Hindriks** (UU), Agent Programming Languages: Programming with Mental Models.

2001-3 | **Maarten van Someren** (UvA), Learning as problem solving.

2001-4 | **Evgueni Smirnov** (UM), Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary

Sets.

2001-5 | **Jacco van Ossenbruggen** (VU), Processing Structured Hypermedia: A Matter of Style.

2001-6 | **Martijn van Welie** (VU), Task-based User Interface Design.

2001-7 | **Bastiaan Schonhage** (VU), Diva: Architectural Perspectives on Information Visualization.

2001-8 | **Pascal van Eck** (VU), A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 | **Pieter Jan 't Hoen** (RUL), Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes.

2001-10 | **Maarten Sierhuis** (UvA), Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design.

2001-11 | **Tom M. van Engers** (VUA), Knowledge Management: The Role of Mental Models in Business Systems Design.

2002

2002-01 | **Nico Lassing** (VU), Architecture-Level Modifiability Analysis.

2002-02 | **Roelof van Zwol** (UT), Modelling and searching web-based document collections.

2002-03 | **Henk Ernst Blok** (UT), Database Optimization Aspects for Information Retrieval.

2002-04 | **Juan Roberto Castelo Valdueza** (UU), The Discrete Acyclic Digraph Markov Model in Data Mining.

2002-05 | **Radu Serban** (VU), The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents.

2002-06 | **Laurens Mommers** (UL), Applied legal epistemology; Building a knowledge-based ontology of the legal domain.

2002-07 | **Peter Boncz** (CWI), Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications.

2002-08 | **Jaap Gordijn** (VU), Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas.

2002-09 | **Willem-Jan van den Heuvel** (KUB), Integrating Modern Business Applications with Objectified Legacy Systems.

2002-10 | **Brian Sheppard** (UM), Towards Perfect Play of Scrabble.

2002-11 | **Wouter C.A. Wijngaards** (VU), Agent Based Modelling of Dynamics: Biological and Organisational Applications.

2002-12 | **Albrecht Schmidt** (UVA), Processing XML in Database Systems.

2002-13 | **Hongjing Wu** (TUE), A Reference Architecture for Adaptive Hypermedia Applications.

2002-14 | **Wieke de Vries** (UU), Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems.

2002-15 | **Rik Eshuis** (UT), Semantics and Verification of UML Activity Diagrams for Workflow Modelling.

2002-16 | **Pieter van Langen** (VU), The Anatomy of Design: Foundations, Models and Applications.

2002-17 | **Stefan Manegold** (UVA), Understanding, Modeling, and Improving Main-Memory Database Performance.

2003

2003-01 | **Heiner Stuckenschmidt** (VU), Onotology-Based Information Sharing In Weakly Structured Environments.

2003-02 | **Jan Broersen** (VU), Modal Action Logics for Reasoning About Reactive Systems.

2003-03 | **Martijn Schuemie** (TUD), Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy.

2003-04 | **Milan Petkovic** (UT), Content-Based Video Retrieval Supported by Database Technology.

2003-05 | **Jos Lehmann** (UVA), Causation in Artificial Intelligence and Law - A modelling approach.

2003-06 | **Boris van Schooten** (UT), Development and specification of virtual environments.

2003-07 | **Machiel Jansen** (UvA), Formal Explorations of Knowledge Intensive Tasks.

2003-08 | **Yongping Ran** (UM), Repair Based Scheduling.

2003-09 | **Rens Kortmann** (UM), The resolution of visually guided behaviour.

2003-10 | **Andreas Lincke** (UvT), Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture.

2003-11 | **Simon Keizer** (UT), Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks.

2003-12 | **Roeland Ordelman** (UT), Dutch speech recognition in multimedia information retrieval.

2003-13 | **Jeroen Donkers** (UM), Nosce Hostem - Searching with Opponent Models.

2003-14 | **Stijn Hoppenbrouwers** (KUN), Freezing Language: Conceptualisation Processes across ICT-Supported Organisations.

2003-15 | **Mathijs de Weerdt** (TUD), Plan Merging in Multi-Agent Systems.

2003-16 | **Menzo Windhouwer** (CWI), Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses.

2003-17 | **David Jansen** (UT), Extensions of Statecharts with Probability, Time, and Stochastic Timing.

2003-18 | **Levente Kocsis** (UM), Learning Search Decisions.

2004

2004-01 | **Virginia Dignum** (UU), A Model for Organizational Interaction: Based on Agents, Founded in Logic.

2004-02 | **Lai Xu** (UvT), Monitoring Multi-party Contracts for E-business.

2004-03 | **Perry Groot** (VU), A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving.

2004-04 | **Chris van Aart** (UVA), Organizational Principles for Multi-Agent Architectures.

2004-05 | **Viara Popova** (EUR), Knowledge discovery and monotonicity.

2004-06 | **Bart-Jan Hommes** (TUD), The Evaluation of Business Process Modeling Techniques.

2004-07 | **Elise Boltjes** (UM), Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes.

2004-08 | **Joop Verbeek** (UM), Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise.

2004-09 | **Martin Caminada** (VU), For the Sake of the Argument; explorations into argument-based reasoning.

2004-10 | **Suzanne Kabel** (UVA), Knowledge-rich indexing of learning-objects.

2004-11 | **Michel Klein** (VU), Change Management for Distributed Ontologies.

2004-12 | **The Duy Bui** (UT), Creating emotions and facial expressions for embodied agents.

2004-13 | **Wojciech Jamroga** (UT), Using Multiple Models of Reality: On Agents who Know how to Play.

2004-14 | **Paul Harrenstein** (UU), Logic in Conflict. Logical Explorations in Strategic Equilibrium.

2004-15 | **Arno Knobbe** (UU), Multi-Relational Data Mining.

2004-16 | **Federico Divina** (VU), Hybrid Genetic Relational Search for Inductive Learning.

2004-17 | **Mark Winands** (UM), Informed Search in Complex Games.

2004-18 | **Vania Bessa Machado** (UvA), Supporting the Construction of Qualitative Knowledge Models.

2004-19 | **Thijs Westerveld** (UT), Using generative probabilistic models for multimedia retrieval.

2004-20 | **Madelon Evers** (Nyenrode), Learning from Design: facilitating multidisciplinary design teams.

2005

2005-01 | **Floor Verdenius** (UVA), Methodological Aspects of Designing Induction-Based Applications.

2005-02 | **Erik van der Werf** (UM), AI techniques for the game of Go.

2005-03 | **Franc Grootjen** (RUN), A Pragmatic Approach to the Conceptualisation of Language.

2005-04 | **Nirvana Meratnia** (UT), Towards Database Support for Moving Object data.

2005-05 | **Gabriel Infante-Lopez** (UVA), Two-Level Probabilistic Grammars for Natural Language Parsing.

2005-06 | **Pieter Spronck** (UM), Adaptive Game AI.

2005-07 | **Flavius Frasincar** (TUE), Hypermedia Presentation Generation for Semantic Web Information Systems.

2005-08 | **Richard Vdovjak** (TUE), A Model-driven Approach for Building Distributed Ontology-based Web Applications.

2005-09 | **Jeen Broekstra** (VU), Storage, Querying and Inferencing for Semantic Web Languages.

2005-10 | **Anders Bouwer** (UVA), Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments.

2005-11 | **Elth Ogston** (VU), Agent Based Matchmaking and Clustering - A Decentralized Approach to Search.

2005-12 | **Csaba Boer** (EUR), Distributed Simulation in Industry.

2005-13 | **Fred Hamburg** (UL), Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen.

2005-14 | **Borys Omelayenko** (VU), Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics.

2005-15 | **Tibor Bosse** (VU), Analysis of the Dynamics of Cognitive Processes.

2005-16 | **Joris Graaumans** (UU), Usability of XML Query Languages.

2005-17 | **Boris Shishkov** (TUD), Software Specification Based on Re-usable Business Components.

2005-18 | **Danielle Sent** (UU), Test-selection strategies for probabilistic networks.

2005-19 | **Michel van Dartel** (UM), Situated Representation.

2005-20 | **Cristina Coteanu** (UL), Cyber Consumer Law, State of the Art and Perspectives.

2005-21 | **Wijnand Derks** (UT), Improving Concur-

rency and Recovery in Database Systems by Exploiting Application Semantics.

2006

2006-01 | **Samuil Angelov** (TUE), Foundations of B2B Electronic Contracting.

2006-02 | **Cristina Chisalita** (VU), Contextual issues in the design and use of information technology in organizations.

2006-03 | **Noor Christoph** (UVA), The role of metacognitive skills in learning to solve problems.

2006-04 | **Marta Sabou** (VU), Building Web Service Ontologies.

2006-05 | **Cees Pierik** (UU), Validation Techniques for Object-Oriented Proof Outlines.

2006-06 | **Ziv Baida** (VU), Software-aided Service Bundling – Intelligent Methods & Tools for Graphical Service Modeling.

2006-07 | **Marko Smiljanic** (UT), XML schema matching – balancing efficiency and effectiveness by means of clustering.

2006-08 | **Eelco Herder** (UT), Forward, Back and Home Again – Analyzing User Behavior on the Web.

2006-09 | **Mohamed Wahdan** (UM), Automatic Formulation of the Auditor's Opinion.

2006-10 | **Ronny Siebes** (VU), Semantic Routing in Peer-to-Peer Systems.

2006-11 | **Joeri van Ruth** (UT), Flattening Queries over Nested Data Types.

2006-12 | **Bert Bongers** (VU), Interactivation – Towards an e-cology of people, our technological environment, and the arts.

2006-13 | **Henk-Jan Lebbink** (UU), Dialogue and Decision Games for Information Exchanging Agents.

2006-14 | **Johan Hoorn** (VU), Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change.

2006-15 | **Rainer Malik** (UU), CONAN: Text Mining in the Biomedical Domain.

2006-16 | **Carsten Riggelsen** (UU), Approximation Methods for Efficient Learning of Bayesian Networks.

2006-17 | **Stacey Nagata** (UU), User Assistance for Multitasking with Interruptions on a Mobile Device.

2006-18 | **Valentin Zhizhkun** (UVA), Graph transformation for Natural Language Processing.

2006-19 | **Birna van Riemsdijk** (UU), Cognitive Agent Programming: A Semantic Approach.

2006-20 | **Marina Velikova** (UvT), Monotone models for prediction in data mining.

2006-21 | **Bas van Gils** (RUN), Aptness on the Web.

2006-22 | **Paul de Vrieze** (RUN), Fundaments of Adaptive Personalisation.

2006-23 | **Ion Juvina** (UU), Development of Cognitive Model for Navigating on the Web.

2006-24 | **Laura Hollink** (VU), Semantic Annotation for Retrieval of Visual Resources.

2006-25 | **Madalina Drugan** (UU), Conditional log-likelihood MDL and Evolutionary MCMC.

2006-26 | **Vojkan Mihajlovic** (UT), Score Region Algebra: A Flexible Framework for Structured Information Retrieval.

2006-27 | **Stefano Bocconi** (CWI), Vox Populi: generating video documentaries from semantically annotated media repositories.

2006-28 | **Borkur Sigurbjornsson** (UVA), Focused Information Access using XML Element Retrieval.

2007

2007-01 | **Kees Leune** (UvT), Access Control and Service-Oriented Architectures.

2007-02 | **Wouter Teepe** (RUG), Reconciling Information Exchange and Confidentiality: A Formal Approach.

2007-03 | **Peter Mika** (VU), Social Networks and the Semantic Web.

2007-04 | **Jurriaan van Diggelen** (UU), Achieving Semantic Interoperability in Multi-agent Systems: A Dialogue-based Approach.

2007-05 | **Bart Schermer** (UL), Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance.

2007-06 | **Gilad Mishne** (UVA), Applied Text Analytics for Blogs.

2007-07 | **Natasa Jovanovic** (UT), To Who It May Concern - Addressee Identification in Face-to-Face Meetings.

2007-08 | **Mark Hoogendoorn** (VU), Modeling of Change in Multi-Agent Organizations.

2007-09 | **David Mobach** (VU), Agent-Based Mediated Service Negotiation.

2007-10 | **Huib Aldewereld** (UU), Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols.

2007-11 | **Natalia Stash** (TUE), Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System.

2007-12 | **Marcel van Gerven** (RUN), Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty.

2007-13 | **Rutger Rienks** (UT), Meetings in Smart En-

vironments; Implications of Progressing Technology.

2007-14 | **Niek Bergboer** (UM), Context-Based Image Analysis.

2007-15 | **Joyca Lacroix** (UM), NIM: a Situated Computational Memory Model.

2007-16 | **Davide Grossi** (UU), Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems.

2007-17 | **Theodore Charitos** (UU), Reasoning with Dynamic Networks in Practice.

2007-18 | **Bart Orriens** (UvT), On the development an management of adaptive business collaborations.

2007-19 | **David Levy** (UM), Intimate relationships with artificial partners.

2007-20 | **Slinger Jansen** (UU), Customer Configuration Updating in a Software Supply Network.

2007-21 | **Karianne Vermaas** (UU), Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005.

2007-22 | **Zlatko Zlatev** (UT), Goal-oriented design of value and process models from patterns.

2007-23 | **Peter Barna** (TUE), Specification of Application Logic in Web Information Systems.

2007-24 | **Georgina Ramrez Camps** (CWI), Structural Features in XML Retrieval.

2007-25 | **Joost Schalken** (VU), Empirical Investigations in Software Process Improvement.

2008

2008-01 | **Katalin Boer-Sorbn** (EUR), Agent-Based Simulation of Financial Markets: A modular,continuous-time approach.

2008-02 | **Alexei Sharpanskykh** (VU), On Computer-Aided Methods for Modeling and Analysis of Organizations.

2008-03 | **Vera Hollink** (UVA), Optimizing hierarchical menus: a usage-based approach.

2008-04 | **Ander de Keijzer** (UT), Management of Uncertain Data - towards unattended integration.

2008-05 | **Bela Mutschler** (UT), Modeling and simulating causal dependencies on process-aware information systems from a cost perspective.

2008-06 | **Arjen Hommersom** (RUN), On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective.

2008-07 | **Peter van Rosmalen** (OU), Supporting the tutor in the design and support of adaptive e-learning.

2008-08 | **Janneke Bolt** (UU), Bayesian Networks: Aspects of Approximate Inference.

2008-09 | **Christof van Nimwegen** (UU), The paradox of the guided user: assistance can be counter-effective.

2008-10 | **Wauter Bosma** (UT), Discourse oriented summarization.

2008-11 | **Vera Kartseva** (VU), Designing Controls for Network Organizations: A Value-Based Approach.

2008-12 | **Jozsef Farkas** (RUN), A Semiotically Oriented Cognitive Model of Knowledge Representation.

2008-13 | **Caterina Carraciolo** (UVA), Topic Driven Access to Scientific Handbooks.

2008-14 | **Arthur van Bunningen** (UT), Context-Aware Querying; Better Answers with Less Effort.

2008-15 | **Martijn van Otterlo** (UT), The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.

2008-16 | **Henriette van Vugt** (VU), Embodied agents from a user's perspective.

2008-17 | **Martin Op 't Land** (TUD), Applying Architecture and Ontology to the Splitting and Allying of Enterprises.

2008-18 | **Guido de Croon** (UM), Adaptive Active Vision.

2008-19 | **Henning Rode** (UT), From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search.

2008-20 | **Rex Arendsen** (UVA), Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.

2008-21 | **Krisztian Balog** (UVA), People Search in the Enterprise.

2008-22 | **Henk Koning** (UU), Communication of IT-Architecture.

2008-23 | **Stefan Visscher** (UU), Bayesian network models for the management of ventilator-associated pneumonia.

2008-24 | **Zharko Aleksovski** (VU), Using background knowledge in ontology matching.

2008-25 | **Geert Jonker** (UU), Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency.

2008-26 | **Marijn Huijbregts** (UT), Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled.

2008-27 | **Hubert Vogten** (OU), Design and Implementation Strategies for IMS Learning Design.

2008-28 | **Ildiko Flesch** (RUN), On the Use of Independence Relations in Bayesian Networks.

2008-29 | **Dennis Reidsma** (UT), Annotations and Sub-

jective Machines - Of Annotators, Embodied Agents, Users, and Other Humans.

2008-30 | **Wouter van Atteveldt** (VU), Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content.

2008-31 | **Loes Braun** (UM), Pro-Active Medical Information Retrieval.

2008-32 | **Trung H. Bui** (UT), Toward Affective Dialogue Management using Partially Observable Markov Decision Processes.

2008-33 | **Frank Terpstra** (UVA), Scientific Workflow Design; theoretical and practical issues.

2008-34 | **Jeroen de Knijf** (UU), Studies in Frequent Tree Mining.

2008-35 | **Ben Torben Nielsen** (UvT), Dendritic morphologies: function shapes structure.

2009

2009-01 | **Rasa Jurgelenaite** (RUN), Symmetric Causal Independence Models.

2009-02 | **Willem Robert van Hage** (VU), Evaluating Ontology-Alignment Techniques.

2009-03 | **Hans Stol** (UvT), A Framework for Evidence-based Policy Making Using IT.

2009-04 | **Josephine Nabukenya** (RUN), Improving the Quality of Organisational Policy Making using Collaboration Engineering.

2009-05 | **Sietse Overbeek** (RUN), Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality.

2009-06 | **Muhammad Subianto** (UU), Understanding Classification.

2009-07 | **Ronald Poppe** (UT), Discriminative Vision-Based Recovery and Recognition of Human Motion.

2009-08 | **Volker Nannen** (VU), Evolutionary Agent-Based Policy Analysis in Dynamic Environments.

2009-09 | **Benjamin Kanagwa** (RUN), Design, Discovery and Construction of Service-oriented Systems.

2009-10 | **Jan Wielemaker** (UVA), Logic programming for knowledge-intensive interactive applications.

2009-11 | **Alexander Boer** (UVA), Legal Theory, Sources of Law & the Semantic Web.

2009-12 | **Peter Massuthe** (TUE, Humboldt-Universitaet zu Berlin), Perating Guidelines for Services.

2009-13 | **Steven de Jong** (UM), Fairness in Multi-Agent Systems.

2009-14 | **Maksym Korotkiy** (VU), From ontology-enabled services to service-enabled ontologies. making ontologies work in e-science with ONTO-SOA

2009-15 | **Rinke Hoekstra** (UVA), Ontology Representation - Design Patterns and Ontologies that Make Sense.

2009-16 | **Fritz Reul** (UvT), New Architectures in Computer Chess.

2009-17 | **Laurens van der Maaten** (UvT), Feature Extraction from Visual Data.

2009-18 | **Fabian Groffen** (CWI), Armada, An Evolving Database System.

2009-19 | **Valentin Robu** (CWI), Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets.

2009-20 | **Bob van der Vecht** (UU), Adjustable Autonomy: Controling Influences on Decision Making.

2009-21 | **Stijn Vanderlooy** (UM), Ranking and Reliable Classification.

2009-22 | **Pavel Serdyukov** (UT), Search For Expertise: Going beyond direct evidence.

2009-23 | **Peter Hofgesang** (VU), Modelling Web Usage in a Changing Environment.

2009-24 | **Annerieke Heuvelink** (VUA), Cognitive Models for Training Simulations.

2009-25 | **Alex van Ballegooij** (CWI), "RAM: Array Database Management through Relational Mapping".

2009-26 | **Fernando Koch** (UU), An Agent-Based Model for the Development of Intelligent Mobile Services.

2009-27 | **Christian Glahn** (OU), Contextual Support of Social Engagement and Reflection on the Web.

2009-28 | **Sander Evers** (UT), Sensor Data Management with Probabilistic Models.

2009-29 | **Stanislav Pokraev** (UT), Model-Driven Semantic Integration of Service-Oriented Applications.

2009-30 | **Marcin Zukowski** (CWI), Balancing vectorized query execution with bandwidth-optimized storage.

2009-31 | **Sofiya Katrenko** (UVA), A Closer Look at Learning Relations from Text.

2009-32 | **Rik Farenhorst and Remco de Boer** (VU), Architectural Knowledge Management: Supporting Architects and Auditors.

2009-33 | **Khiet Truong** (UT), How Does Real Affect Affect Affect Recognition In Speech?.

2009-34 | **Inge van de Weerd** (UU), Advancing in Software Product Management: An Incremental Method Engineering Approach.

2009-35 | **Wouter Koelewijn** (UL), Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling.

2009-36 | **Marco Kalz** (OUN), Placement Support for

Learners in Learning Networks.

2009-37 | **Hendrik Drachsler** (OUN), Navigation Support for Learners in Informal Learning Networks.

2009-38 | **Riina Vuorikari** (OU), Tags and self-organisation: a metadata ecology for learning resources in a multilingual context.

2009-39 | **Christian Stahl** (TUE, Humboldt-Universitaet zu Berlin), Service Substitution – A Behavioral Approach Based on Petri Nets.

2009-40 | **Stephan Raaijmakers** (UvT), Multinomial Language Learning: Investigations into the Geometry of Language.

2009-41 | **Igor Berezhnyy** (UvT), Digital Analysis of Paintings.

2009-42 | **Toine Bogers** (UvT), Recommender Systems for Social Bookmarking.

2009-43 | **Virginia Nunes Leal Franqueira** (UT), Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients.

2009-44 | **Roberto Santana Tapia** (UT), Assessing Business-IT Alignment in Networked Organizations.

2009-45 | **Jilles Vreeken** (UU), Making Pattern Mining Useful.

2009-46 | **Loredana Afanasiev** (UvA), Querying XML: Benchmarks and Recursion.

2010

2010-01 | **Matthijs van Leeuwen** (UU), Patterns that Matter.

2010-02 | **Ingo Wassink** (UT), Work flows in Life Science.

2010-03 | **Joost Geurts** (CWI), A Document Engineering Model and Processing Framework for Multimedia documents.

2010-04 | **Olga Kulyk** (UT), Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments.

2010-05 | **Claudia Hauff** (UT), Predicting the Effectiveness of Queries and Retrieval Systems.

2010-06 | **Sander Bakkes** (UvT), Rapid Adaptation of Video Game AI.

2010-07 | **Wim Fikkert** (UT), A Gesture interaction at a Distance.

2010-08 | **Krzysztof Siewicz** (UL), Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments.

2010-09 | **Hugo Kielman** (UL), Politiële gegevensverwerking en Privacy, Naar een effectieve waarborging.

2010-10 | **Rebecca Ong** (UL), Mobile Communication and Protection of Children.

2010-11 | **Adriaan Ter Mors** (TUD), The world according to MARP: Multi-Agent Route Planning.

2010-12 | **Susan van den Braak** (UU), Sensemaking software for crime analysis.

2010-13 | **Gianluigi Folino** (RUN), High Performance Data Mining using Bio-inspired techniques.

2010-14 | **Sander van Splunter** (VU), Automated Web Service Reconfiguration.

2010-15 | **Lianne Bodenstaff** (UT), Managing Dependency Relations in Inter-Organizational Models.

2010-16 | **Sicco Verwer** (TUD), Efficient Identification of Timed Automata, theory and practice.

2010-17 | **Spyros Kotoulas** (VU), Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications.

2010-18 | **Charlotte Gerritsen** (VU), Caught in the Act: Investigating Crime by Agent-Based Simulation.

2010-19 | **Henriette Cramer** (UvA), People's Responses to Autonomous and Adaptive Systems.

2010-20 | **Ivo Swartjes** (UT), Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative.

2010-21 | **Harold van Heerde** (UT), Privacy-aware data management by means of data degradation.

2010-22 | **Michiel Hildebrand** (CWI), End-user Support for Access to Heterogeneous Linked Data.

2010-23 | **Bas Steunebrink** (UU), The Logical Structure of Emotions.

2010-24 | **Dmytro Tykhonov** (), Designing Generic and Efficient Negotiation Strategies.

2010-25 | **Zulfiqar Ali Memon** (VU), Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective.

2010-26 | **Ying Zhang** (CWI), XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines.

2010-27 | **Marten Voulon** (UL), Automatisch contracteren.

2010-28 | **Arne Koopman** (UU), Characteristic Relational Patterns.

2010-29 | **Stratos Idreos** (CWI), Database Cracking: Towards Auto-tuning Database Kernels.

2010-30 | **Marieke van Erp** (UvT), Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval.

2010-31 | **Victor de Boer** (UVA), Ontology Enrichment from Heterogeneous Sources on the Web.

2010-32 | **Marcel Hiel** (UvT), An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems.

2010-33 | **Robin Aly** (UT), Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval.

2010-34 | **Teduh Dirgahayu** (UT), Interaction Design in Service Compositions.

2010-35 | **Dolf Trieschnigg** (UT), Proof of Concept: Concept-based Biomedical Information Retrieval.

2010-36 | **Jose Janssen** (OU), Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification.

2010-37 | **Niels Lohmann** (TUE), Correctness of services and their composition.

2010-38 | **Dirk Fahland** (TUE), From Scenarios to components.

2010-39 | **Ghazanfar Farooq Siddiqui** (VU), Integrative modeling of emotions in virtual agents.

2010-40 | **Mark van Assem** (VU), Converting and Integrating Vocabularies for the Semantic Web.

2010-41 | **Guillaume Chaslot** (UM), Monte-Carlo Tree Search.

2010-42 | **Sybren de Kinderen** (VU), Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach.

2010-43 | **Peter van Kranenburg** (UU), A Computational Approach to Content-Based Retrieval of Folk Song Melodies.

2010-44 | **Pieter Bellekens** (TUE), An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain.

2010-45 | **Vasilios Andrikopoulos** (UvT), A theory and model for the evolution of software services.

2010-46 | **Vincent Pijpers** (VU), e3alignment: Exploring Inter-Organizational Business-ICT Alignment.

2010-47 | **Chen Li** (UT), Mining Process Model Variants: Challenges, Techniques, Examples.

2010-48 | **Milan Lovric** (EUR), Behavioral Finance and Agent-Based Artificial Markets.

2010-49 | **Jahn-Takeshi Saito** (UM), Solving difficult game positions.

2010-50 | **Bouke Huurnink** (UVA), Search in Audiovisual Broadcast Archives.

2010-51 | **Alia Khairia Amin** (CWI), Understanding and supporting information seeking tasks in multiple sources.

2010-52 | **Peter-Paul van Maanen** (VU), Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention.

2010-53 | **Edgar Meij** (UVA), Combining Concepts and Language Models for Information Access.

2011

2011-01 | **Botond Cseke** (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models.

2011-02 | **Nick Tinnemeier** (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language.

2011-03 | **Jan Martijn van der Werf** (TUE), Compositional Design and Verification of Component-Based Information Systems.

2011-04 | **Hado van Hasselt** (UU), Insights in Reinforcement Learning - Formal analysis and empirical evaluation of temporal-difference learning algorithms.

2011-05 | **Base van der Raadt** (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 | **Yiwen Wang** (TUE), Semantically-Enhanced Recommendations in Cultural Heritage.

2011-07 | **Yujia Cao** (UT), Multimodal Information Presentation for High Load Human Computer Interaction.

2011-08 | **Nieske Vergunst** (UU), BDI-based Generation of Robust Task-Oriented Dialogues.

2011-09 | **Tim de Jong** (OU), Contextualised Mobile Media for Learning.

2011-10 | **Bart Bogaert** (UvT), Cloud Content Contention.

2011-11 | **Dhaval Vyas** (UT), Designing for Awareness: An Experience-focused HCI Perspective.

2011-12 | **Carmen Bratosin** (TUE), Grid Architecture for Distributed Process Mining.

2011-13 | **Xiaoyu Mao** (UvT), Airport under Control; Multiagent Scheduling for Airport Ground Handling.

2011-14 | **Milan Lovric** (EUR), Behavioral Finance and Agent-Based Artificial Markets.

2011-15 | **Marijn Koolen** (UVA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval.

2011-16 | **Maarten Schadd** (UM), Selective Search in Games of Different Complexity.

2011-17 | **Jiyin He** (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness.

2011-18 | **Mark Ponsen** (UM), Strategic Decision-Making in complex games.

2011-19 | **Ellen Rusman** (OU), The Mind's Eye on Personal Profiles.

2011-20 | **Qing Gu** (VU), Guiding service-oriented software engineering - A view-based approach.

2011-21 | **Linda Terlouw** (TUD), Modularization and Specification of Service-Oriented Systems.

2011-22 | **Junte Zhang** (UVA), System Evaluation of Archival Description and Access.

2011-23 | **Wouter Weerkamp** (UVA), Finding People and their Utterances in Social Media.

2011-24 | **Herwin van Welbergen** (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior.

2011-25 | **Syed Waqar ul Qounain Jaffry** (VU), Analysis and Validation of Models for Trust Dynamics.

2011-26 | **Matthijs Aart Pontier** (VU), Virtual Agents for Human Communication.

2011-27 | **Aniel Bhulai** (VU), Dynamic website optimization through autonomous management of design patterns.

2011-28 | **Rianne Kaptein** (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure.

2011-29 | **Faisal Kamiran** (TUE), Discrimination-aware Classification.

2011-30 | **Egon van den Broek** (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions.

2011-31 | **Ludo Waltman** (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality.

2011-32 | **Nees Jan van Eck** (EUR), Methodological Advances in Bibliometric Mapping of Science.

2011-33 | **Tom van der Weide** (UU), Arguing to Motivate Decisions.