INSTITUT FÜR INFORMATIK der Ludwig-Maximilians-Universität München

OVERVIEW OF ARGUMENTATION LOGIC

Joshua Klinger

Bachelorarbeit

SupervisorProf. Dr. François BryAbgabe am30.07.2022

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

München, den 30.07.2022

Joshua Klinger

ii

Abstract

Argumentation is based on distinguishing relevant arguments from insignificant arguments and examining conclusions for a given problem. It can also entail identifying conflicts, causing to seperate pros and cons for certain conclusions. In the 1990s, innovations and shifts in the field of artificial intelligence led to a more formal and computational argumentation theory, which yielded the field known as Argumentation Logic.

Argumentation Logic is likely to be a very influential field of research for the future of artificial intelligence more specifically logic, law, optimization, security administration and even philosophy.

This thesis is an overview of the achievements in Argumentation Logic and of the history of Argumentation Logic research. It also mentions the most significant scientists in the field among others Dung, Pollock, Loui, Prakken. Furthermore the general semantics of the abstract argumentation framework, applications are displayed and finally an extension of the framework to structured arguments is introduced.

iv

Zusammenfassung

Argumentierung basiert auf der Trennung von relevanten Argumenten von unbedeutenden und der Untersuchung von Schlussfolgerungen für ein gegebenes Problem. Es kann auch die Identifizierung von Konflikten beinhalten, was zu einer Separierung von pro und contra führt für eine bestimmte Schlussfolgerung. In den 1990ern haben Innovationen und Veränderungen im Forschungsgebiet der künstlichen Intelligenz zu einer mehr formalen und rechnerischen Argumentationstheorie geführt.

Das Thema Argumentationslogik war und wird ein sehr einflussreiches Forschungsgebiet für die Zukunft der künstlichen Intelligenz sein, im speziellen für die Logik, Recht, Optimierung, Argumentationsprobleme, Sicherheitsüberwachung oder auch philosophische Probleme.

Diese Arbeit schafft einen Überblick über das Thema und gibt einen Einblick in die Forschungsgeschichte und ihre bedeutsamsten Wissenschaftler, wie Dung, Pollock, Loui, Prakken, die generelle Semantik des abstrakten Argumentationssystems, Anwendungen und abschließend mit einer Erweiterung des System für strukturierte Argumente. vi

Acknowledgments

First of all I would like to thank Prof. Bry for the constant help and advice. Throughout the entire work he always had great comments and improvements, which really helped improve not only my work but also myself on how to approach academic work. Furthermore I would like to thank my family and close friends, that always supported me and recognized my dedication towards this work and gave the required room. Thank you for always listening to me and being so understanding.

viii

Contents

1	Arg	umentation Logic: Introduction	1	
	1.1	History	1	
	1.2	Current state of research and motivation	3	
	1.3	Outline	4	
2	Argumentation Logic: A framework for abstract argumentation			
	2.1	Motivation	5	
	2.2	The status of arguments	7	
	2.3	The two assignment approaches	10	
	2.4	Relation between different semantics	14	
	2.5	Argument-based reconstruction of nonmonotonic logics	19	
3	Arg	umentation Logic: Applications	23	
	3.1	General ideas	23	
	3.2	Dialectics for grounded and preferred semantics	25	
	3.3	Simplification of the P-game	32	
4	Arg	umentation Logic: Argumentation with structured arguments	35	
	4.1	Argumentation systems with structured arguments	35	
	4.2	Domain-specific vs general inference rules	43	
	4.3	Caminada's and Amgoud's Rationality postulates	45	
	4.4	Serial and parallel self-defeat	47	
5	Arg	umentation Logic: Conclusion and discussion	49	
	5.1	Relevance for present and outlook on future	49	

CONTENTS

CHAPTER 1

Argumentation Logic: Introduction

1.1 History

The history of argumentation dates back to the days of Platon and his method "Plato's Dialectic" [51]. This argumentative method encapsulates a discourse between two or more people with different opinions about a subject, with the wish to determine the truth of the matter through reasoned arguments. A second example would be "Leibniz's Dream" [41]: "The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right.".

"You take the words in the sense which is most damaging to the argument" ~ Plato[51]

In the past decade argumentation gained in importance: John Pollock (1940-2009) made valuable contributions to artificial intelligence (AI): first to the study of defeasible reasoning and then to the study of decision-theoretic planning and practical cognition. Many vital topics in the AI field were first or further studied by Pollock, for instance argument structure, argument strength, argument labellings, the nature of defeasible reasons, the interaction between deductive and defeasible reasons, rebutting versus undercutting, defeat, self-defeat and resource-bounded argumentation [56], [57], [52], [53], [54]. While investigating defeasible reasoning, Pollock created one of the first formal systems for argumentation-based inference.

Pollock's work on argumentation was also impacted by Ross's defeasible reasons [68]. "Defeasibility" derives from legal philosophy, in particular Hart [38]. Hart encountered, that "legal concepts are defeasible in that the conditions for when a fact situation classifies as an instance of a legal concept (such as 'contract'), are only ordinarily, or presumptively, sufficient.". If one party successfully proves these conditions in a law suit, it does not affect the outcome that the case is settled.

Alongside Pollock, there were multiple groups of researchers, continuing his work. Through the use of the notions *support* and *attack*, Birnbaum, Flowers and McGuire [17] applied argumentation inspired methods in an AI setting. A major historical influence emerged from AI and Law work on the computational modelling of legal argument. One of the earliest works in that field was the TAXMAN II project [44], [45]. It was supposed to help lawyers and judges in "hard cases", where they have to "construct a theory of the disputed rules that produces the desired legal result, and then to persuade the relevant audience that this theory is preferable to any theories offered by an opponent.". There were a number of scientist stemming from AI and Law that contributed to the overall topic of argumentation, including Trevor Bench-Capon [10], [8], [9], [7], [14], [27], Tom Gordon [31], [32], [34], Giovanni Sartor [63], [65], [11], Bart Verheij [73], [30], [16], [74], [72], and Henry Prakken [60], [66], [61], [33], [67], [62].

Taking a look at the first AI systems for argumentation-based inference, we can see that they were not influenced by the aforementioned philosophical advancement. They were rather introduced as new ways to execute nonmonotonic logic. The topic of nonmonotonic logic became popular around 1980 and many approaches were being pursued. Towards the end of the 1980's, the discipline of nonmonotonic logic had been seen as a big part of artificial intelligence. The field was driven due to the fact that commonsense reasoning often includes incomplete or inconsistent information, where cases logical deduction is not a useful reasoning model. If information is incomplete, then there cannot be a useful deduction. If it is inconsistent, then anything deductively is implied. Nonmonotonic logic accepts 'jumping to conclusions', if there is no information proving the contrary. A very prominent example is 'birds can typically fly, Tweety is a bird, therefore (presumably) Tweety can fly'. This inference holds until there is information available that Tweety is not a typical bird with respect to flying, for example a penguin. Nonmonotonic logic can also simulate the deduction of useful conclusions from inconsistent information, in particular, by concentrating on consistent subsets of the inconsistent information. A few years after the first paper in the now popular special issue on nonmonotonic logic of the Artificial Intelligence journal [18], the theory emerged that nonmonotonic inference can be displayed as the competition between arguments.

The first nonmonotonic reasoning systems with an argumentation spirit includes the work of Touretzky [70], [71] on inheritance systems. That thought was later on continued by several researchers Horty et al. [39]. Inheritance systems show in what way objects inherit properties from the classes to which they belong. They are nonmonotonic because of inheriting properties of classes by subclasses can be blocked by exceptions. For instance, penguins do not inherit the property of being able to fly from birds. Inspite of the fact that the work on inheritance systems did not make use of argumentation terms, those systems still have all the attributes of argumentation systems. Firstly inheritance paths practically are arguments. Especially the conclusion of Tweety the penguin being able to fly can be drawn using the path 'Penguins are birds and birds can fly' while the conclusion that Tweety the penguin in fact cannot fly can be drawn via the inheritance path 'Penguins cannot fly'. Inheritance systems also contain various notions of conflict between inheritance, including definitions of whether a path is 'permitted' considering its conflict relations with other paths. Since there are old-fashioned technical solutions, the work on inheritance paths has clearly influenced the growth of the first AI argumentation systems. Because of that, these publications are a great source of examples.

Another influencial person in the beginnings was Ron Loui. His work [42] was important in promoting the concept of formulating nonmonotonic logic as argumentation. He and Guillermo Simari developed a technically mature version of his ideas [69]. Numerous of his papers supported the idea of computational dialectics and were consequently relevant for dialogue models of argumentation. The entirety of these thoughts is in [43], which spread amongst researchers for various of years until the publication in 1998.

Other notable early work came from Nute [48], which later turned into Defeasible Logic [49]. This approach is similar to the one of argumentation, but despite from conflict and defeat occuring between arguments, in Defeasible Logic they happen between rules. Because of that, the work on Defeasible Logic has deviated from the field of computational

argument. In spite of that, scientists have taken a closer look at the formal relation with argumentation approaches. Specifically, [36] studied to what degree defeasible logics can be reformulated in the sense of Dung's theory of abstract argumentation frameworks. In the following section, the developments after Dung's paper is shown and further extensions of his work with a few examples.

1.2 Current state of research and motivation

Up until 1995, certain publications on structured argumentation had specific and at times offhand definitions of argument evaluation. On the contrary most of the work since 1995 on structured argumentation follows Dung's approach or at the minimum explains the relation with it. Various work that is using Dung's view gives definitions for the structure of arguments and the nature of attack. Because of that, abstract argumentation frameworks are generated in a special way. Arguments can be evaluated based on one of the abstract argumentation semantics and the framework's acceptability status can be used to define nonmonotonic consequence notions for their statements.

Nevertheless, there are various other advances that digresse from Dung's approach. The first one is Argumentation models of plausible reasoning, which can be divided into assumption-based argumentation and classical argumentation. Parallel to the proposition of argumentation as a way of dealing with inconsistency in classical logic, assumption-based argumentation (ABA) arose through attempts to deliver an argumentation-theoretic semantics towards the negation of logic-programming as failure [20], [19]. Similar to the views of classical logic, ABA also presumes a unique 'base logic', which is called 'deductive system' in ABA, composed of a set of inference rules defined over a logical language. If there is a set of 'assumptions' formulated in the logical language, arguments are then deductions of claims, that are using rules and maintained by sets of assumptions.

Even though ABA and Dung's approach clearly have the commonalities, ABA, which was first developed by [19], does not generate abstract argumentation frameworks. Rather, sets of assumptions are its extension, caused by turning attack relations among arguments to attack relations among sets of assumptions. Merely ten years later, Dung gave ABA an explicit Dungean formulation [25]. At present, there is a debate on whether the correspondence holds for all current abstract argumentation semantics or not [29], [22]. At first, ABA was used theoretically as a framework for nonmonotonic logic. Over time though the focus has deviated into developing algorithms and implementations and the application of them to a broad spectrum of reasoning and decision problems.

The second kind of argumentation model of plausible reasoning is 'classical' or 'deductive' argumentation. Elvang-Göransson et al. [26] and [12] were the originators of this work. The 'classical' part refers to examples with classical propositional or first order logic. 'Deductive argumentation' whereas is intended for approaches that draw off from particular base logics, if they are 'deductive'. The term 'deductive' is often used in an informal way, for example in [13]. A deductive inference is being described as 'infallible in the sense that it does not introduce uncertainty'. This in fact though goes hand in hand with Pollock's notion of a deductive reason. Not to long ago Amgoud and Besnard [1], [2] delivered an explicit interpretation, using the assumption that the base logic satisfies the properties of a 'Tarskian abstract logic'. It is based on Tarski's notion of an abstract logic that only assumes some unspecified logical language L, and an outcome or consequence operator over this language, that assigns a subset of L to every subset of L (its logical result or consequence).

A couple of years after Dung's paper, there have also been several graph-based approaches. Not arguments but statements and their relations are the focal point. Here one can also see the effect of Pollock's work [55], considering his system is rather formalised in terms of 'in-

ference graphs' than in terms of arguments, where nodes are connected by inference links or defeat links.

Gordon et al. [35] considered the Carneades framework 'of argument and burden of proof'. Its main form is that of an argument graph, that is similar to Pollock's inference graphs. Here statement nodes are linked to each other through argument nodes, that register the inferences from one or multiple nodes to another. Opposite to Pollock, Carneades does not mark conflicts as a unique type of link between statement nodes. The inferences can rather be pro or con a statement.

Contrary to most of the early work being on epistemic reasoning, there has now been more attention for practical reasoning. In the midst of the early papers was Fox and Parsons [28], which was motivated by medical decision making. Pieces of work like Grasso et al. [37] plan for a nutrition advice system and Bench-Capon's [8] formal work on value-based argumentation frameworks. Each work was affected by Perelman and Olbrecht-Tyteca's [50] idea, that if an argument in a normal discourse is good does not depend on its logical form, but if it is able to convince the actual audience. Value-based argumentation was then continued by Atkinson [6] and Bench-Capon [7]. They characterized value-based argumentation frameworks with an argumentation scheme approach taken from Walton's [79] schemes for practical reasoning.

Finally, a recent trend was combining argumentation-based inference with probability theory. Since argumentation has been used as a model for reasoning under uncertainty, this is no surprise. This combination was needed, because of some work, where probabilistic models are the object of argumentative discourse, e.g. Nielsen and Parsons [47], who showed a way of jointly constructing Bayesian networks in argumentation process. There are two manners, where the uncertainty is in or about the arguments: if it is in them, probabilities are intrinsic to an argument. They can be used for weakening an argument, if there is an uncertainty towards the truth of its conjectures or its reliability of its inferences. For instance *Most Belgians speak French, Mathieu is Belgian, therefore (presumeably) Mathieu speaks French.* So, if all conjectures of an argument are definite and it only makes deductive inferences, the argument should attain maximum probabilistic strength. This use of probability is called the epistemic approach [40].

1.3 Outline

Taking a look at the following chapters, this thesis gives an in-depth overview of the famous abstract argumentation framework, where we will see the two main approaches and their specific semantics. Thereafter a display certain applications follows, in particular the famous P-game and a variation of it. It will also include an extension to the semantics described in chapter 2. In chapter 4, the introduction of structured arguments follows and a demonstration of how to apply this framework on them. Last but not least, a conclusion will be drawn and an open discussion about the future relevance and the possibilities of this topic will be extended.

CHAPTER 2

Argumentation Logic: A framework for abstract argumentation

In this chapter a totally abstract framework for the argumentation semantics is presented. Such a framework allows the inner structure of arguments and the class of defeat relation to remain undefined. As an input, the framework expects a set (of arguments), which is ordered by a binary relation (of defeat). It then defines multiple 'semantics', namely properties, which all subsets of the set of arguments should fulfill: justifiability or defensibility. Contrary to standard first-order logic, argumentation semantics are not based around the notion of truth: Argumentation systems schematize reasoning which is defeasible. They are not focused on the truth of propositions, but rather with justification of accepting a proposition as the truth. One is justified in taking a proposition as the truth, if there exists an argument in favour of the proposition, which one is reasonable to accept. Argumentbased semantics identify the terms for this situation.

Dung [24] introduced the abstract framework in 1995. Theoretically, his publication was built upon various more detailed argumentation systems. To be exact, Pollock [52]-[57] and Vreeswijk [75], the contributions to Argumentation Logic of both of whom will be discussed further on in Chapter 4. Dung's framework was groundbreaking in various ways: Firstly, it entails a broad account of argumentation semantics, which can be applied to all systems that illustrates his framework. Secondly he drew an exact correlation between several systems, through converting them into his abstract format. It also made a broad study of formal properties of systems possible. It was inherited by all systems that embedded his framework. Lastly, all of this applies to argumentation systems but also to several other nonmonotonic logics, given that Dung [24] displayed for various non-monotonic logics how they can be converted into his abstract framework.

2.1 Motivation

As an introduction, this thesis illustrates the concept of argumentation-based inference with a discussion between two people, namely *A* and *B*. They want to clarify, if it is morally acceptable for a newspaper to release information regarding a politician's secluded life. Let us now understand the following information:

- (1) The information *I* refers to the health of person *P*.
- (2) *P* disagrees with the release of *I* which is part of her/his privacy.

(3) Information referring to the health of a person is part of that person's privacy.

A now proclaims the ethical principle that

(4) Information which is part of a person's privacy should not be published if that person does not agree with the release of it.

and concludes "With that, the newspaper may not publish I".

(3)		(1)		
I refe	rs to the health			
of P v	which is part		(2)	
of his	/her privacy.			
	P disagrees with			
	release of I which			(4)
	is part of his/her	privacy.		
	I she	ould not be		
	relea	ased.		

B as a matter of fact approves of (4) and is thereby now committed to (1)-(4). However *B* does not agree with A's conclusion. The following 3 propositions are B's reply:

- (5) *P* is a cabinet minister.
- (6) *I* concerns a disease that could affect *P*'s political work.
- (7) Information that could affect a cabinet minister's political work is of publical significance.

B also claims that there is a legal norm that

(8) Information that is of public significance may be published by the newspaper.

B closes by claiming that consequently the newspapers can talk about *P*'s disease.

(5)	(6)		
I concer could af work of	ns a disease that fect the political the politician <i>P</i> .	(7)	
<i>I</i> is of public significance.			(8)
	The newspapers may publish <i>I</i> .		

A now on the other hand approves of (5)-(7) and also accepts (8) as a legal norm, but does not agree with the conclusion drawn by *B*. As an alternative, *A* tries to defend his argument by claiming that he has a more persuasive argument, through the following points:

- (9) The possibility of the in I mentioned disease affecting P's work is small.
- (10) If the possibility of the disease affecting *P*'s work is small, then rule (4) has supremacy over rule (8).

6

2.2. THE STATUS OF ARGUMENTS

Therefore it can be concluded, that the legal norm used in A's first argument has priority over the legal norm in B. Through that, A's first argument is stronger than B's and with that all the newspaper should not report on P's disease.

(9) (10) Rule (4) is supreme over rule (8)

Going into more detail, one can disect this dispute into different stages. Naturally, it seems that through the statement of (4) from A and the acceptance from B, viz. (1,2,3,4), guarantees the conclusion that I may not be released. After the counterargument of B (5)-(8) and the approval of A, it does not seem as clear. Now the common ground of this discussion entails (1)-(8), which brings up a conflict between two arguments. In fact (1)-(8) does not have a preference between A and B, which means A's conclusion isn't safe. A's second argument, which proclaims a supremacy between the two legal norms, makes its conclusion more preferable. After the extension from (1)-(10), A's claim must be accepted as warranted.

Formalizing this kind of reasoning in logical frameworks or systems is called 'argumentation logic'. Through this example, one can see that these frameworks miss the monotonic property of 'standard', deductive logic (for instance, first-order predicate logic, FOL). Based on FOL, *A*'s argument is stated through (1)-(4), it is also stated by (1)-(8).

Nevertheless, argumentation logics have a wider range than just reasoning with such empirical abstractions. It can be applied to any sort of reasoning with opposing information. It is also important, that various argumentation frameworks allow the construction and attack of certain arguments. They are called 'ampliative', for instance inductive, analogical and abductive arguments.

One domain, which is a popular application of argumentation systems, is legal reasoning. This is hardly surprising, since legal rules and norms often have objections or have a conflict amongst themselves. Furthermore legal reasoning happens in an hostile environment. This means, that notions like argument, counterargument, rebuttal and defeat are very common.

2.2 The status of arguments

I will now introduce the abstract argument-based semantics. Their task is defining the terms under which an acceptance of an argument is justified. These terms require an 'input' set of arguments, which a binary relation of 'defeat' has structured. In multiple publication, for instance [24], instead of 'defeat' the expression is 'attack'. The framework is upmost abstract, which leaves the structure of arguments and the evidence for defeat unspecified. This input is defined as 'argumentation theory', in [24] argumentation framework:

Definition 2.2.1 (Abstract argumentation theories)

- (1) An *abstract argumentation theory* (*AAT*) is a 2-tuple (*Args,defeat*). *Args* is a set of arguments and *defeat* a binary defeat relation.
- (2) A set *S* of arguments defeats an argument *A*, if one argument in *S* defeats *A*. *S* defeats a set *S'* of arguments, if it defeats a member of *S'*.

In a more applied view, Args can be seen as all arguments that can be built in a particular logic with a provided set of premises. Later on, we will see an exception regarding 'partial computation' where this is not the case. If not otherwise specified, a random but absolute argumentation theory is implied from here on. If we remind ourselves of the definition '*A* defeats *B*', it can be that *A* defeats *B* and *B* defeats *A*. *A* may here be conflicting with *B* but *A* is stronger than *B*. When *A* defeats *B* and *B* does not defeat *A*, it can be said that *A* strictly defeats *B*, or else *A* weakly defeats *B*.

We will now focus on defining the notion of a justified argument:

Definition 2.2.2 (Justified or not justified arguments)

- (1) An argument is justified, if it is defeated (or not) by not justified arguments.
- (2) An argument is not justified, if it is defeated by a justified argument.

This outlining is good for simple examples, where no argument is clearly winning:

Example 2.2.3

A,B,C are arguments; B defeats A, C defeats B

 $A \longleftarrow B \longleftarrow C$

An example for this image would be:

- (1) A = 'Tweety flies because it is a bird'
- (2) B = 'Tweety cannot fly because it is a penguin'
- (3) C = 'The remark that Tweety is a penguin is uncertain'

C is a justified argument, because it is not defeated by another argument. With that, *B* is not justified and *A* is justified: Even though *A* is defeated by *B*, it is restored by *C*, since *C* makes *B* not justified.

There are instances, where definition 2.2.2 is either circular or ambiguous. Namely, when arguments have equal strength and interfere with eachother, it is uncertain which argument should remain.

Example 2.2.4 (Even cycle)

A,*B* are arguments. *A* defeats *B*, *B* defeats *A*.



An example for this figure would be:

- (1) A = 'Nixon was a pacifist since he was a qaker'
- (2) B = 'Nixon was not a pacifist since he was a republican'

2.2. THE STATUS OF ARGUMENTS

Can *A* be justified ? Yes, if *B* is not justified. Can *B* be not justified ? Yes if *A* is justified. With that being said, definition 2.2.2 can be fully satisfied if *A* is and *B* is not justified.On the other hand, we could say *A* is not and *B* is justified, which would result in two possible 'status assignments': one where *A* is justified with the sacrifice of *B* and the other way around. In the end it is not reasonable to choose one over the other.

There are two ways of avoiding this kind of problem: one is changing definition 2.2.2, where there is only one precise way to designate a status to an argument. If there are 'undecided conflicts' both arguments have the status 'not justified'. The other method accepts multiple status assignments as a characteristic: it defines an argument as 'genuinely justified', if it has this status in all available assignments.

There is however a problem with definition 2.2.2, considering self-defeating arguments.

Example 2.2.5 (Self-defeat)

There is an L, such that L defeats L. L is not justified, so all arguments defeating L are not justified. Because of section 1 of definition 2.2.2, L is justified. This leads to a contradiction. Lets us assume, L is justified. L is now defeated by a justified argument and with section 2 of definition 2.2.2, L is not justified. This also leads to a contradiction.



Hence, definiton 2.2.2 indicates that self-defeating arguments do not exist. On the contrary they can be found in regular disputes:

Example 2.2.6 (The Liar)

Self defeating arguments can be constructed similar to the Liar paradox.

The German population can be separated into two types: people telling the truth and people lying. Simon is a German priest. German priests are telling the truth. Hence we can say that Simon is a truth-teller. If Simon now *claims* he is a liar, what is he then: a truth-teller or a liar ?

The Liar paradox has either way no conclusion and always ends in a contradiction.

- (1) Simon is telling the truth. Therefore what he says must be true. So, Simon is a liar. This is a contradiction.
- (2) Simon is telling a lie. What he is saying is therefore false. So, Simon is not a liar. Due to the German population constantly lying or telling the truth, Simon always tells the truth. This is a contradiction.

Because of this paradox, a self-defeating argument *L* can be madde out of (1):

	German priests are reliable truth-tellers	Simon is a German priest
Simon says:	Simor	n is a
'I am lying'	reliable tr	uthteller
	Simon is lying	
_	Simon is not a reliable truth-teller	

If the argument 'Simon is not a reliable truth-teller' is equally as strong as its subargument 'Simon is a reliable truth-teller', then *L* defeats his own subargument. This leads to it being a self-defeating argument.

Finally, definition 2.2.2 seems to need another correction, so that there is a margin for the self-defeating arguments. In the following there will be a discussion about how each semantics handles self-defeat.

2.3 The two assignment approaches

We will now introduce a way of assigning exactly one available status to an argument. This method of 'unique-status-assignment' can be shown through its validation on 'reinstatement'. Let there be three arguments A,B,C, where B defeats A but B is defeated by C. Here C 'reinstates' A.



An argument defeated by a different argument can be justified, if that other argument is defeated by another justified argument. This can now be formalised by this 'unique-status-assignment' with the notion of *acceptability* with a fixed-point operator.

Definition 2.3.1 (Acceptability)

An argument *A* is *acceptable* towards a set *S* of arguments, if every argument defeating *A* is defeated by *S*. When *A* is acceptable towards *S*, one can say 'S *defends A*'.

Those arguments from *S* could be seen as arguments being able to reinstate *A*, if *A* should be defeated.

This notion is not quite sufficient. In Example 2.2.4 there is a set S = A. It is trivial, that A is acceptable regarding S, given that all arguments defeating A are being defeated by another argument in S. We obviously do not want an argument reinstating itself. We shall now examine the following operator, which returns for every set of arguments the set of all arguments, that are all acceptable to it.

Definition 2.3.2 (Grounded semantics)

Let there be an abstract argumentation theory *AAT* and $S \subseteq Args_{AAT}$. Then the operator F^{AAT} is defined as:

 $F^{AAT}(S) = \{ A \in Args_{AAT} \mid A \text{ is acceptable regarding } S \}$

The grounded extension of AAT is the least fixed point of F^{AAT} .

It is possible to show that *F* has a least fixed, that the notion of a grounded extension is well-defined. The general idea is if an argument is acceptable regarding *S*, it is acceptable regarding any superset of *S*, in order for *F* being monotonic. Self-reinstating can be prevented by specifying the set of justified arguments as the least fixed point. In Example 2.2.4 we would have $F(\emptyset) = \emptyset$, because the sets *A* and *B* are fixed points of *F* but not its least

10

2.3. THE TWO ASSIGNMENT APPROACHES

fixed. Through these remarks we can now define a justified argument in grounded semantics:

Definition 2.3.3 (Justified arguments in grounded semantics)

An argument is defined as *justified* regarding grounded semantics, if it is part of the grounded extension.

When using these definitions, it is beneficial knowing that the least fixed point of *F* can be estimated:

Proposition 2.3.4

(Dung (1995) [24]) Let us examine the following points:

•
$$F^0 = \emptyset$$

• $F^{i+1} = \{A \in Args \mid A \text{ is acceptable regarding } F^i\}.$

Let $F^{\omega} = \bigcup_{i=0}^{\infty} (F^i)$. Then the following applies:

- (1) Every argument in F^{ω} is justified.
- (2) If every argument is being defeated by maximally a limited amount of arguments, then the argument is justified if it is in F^{ω} .

Proof. (1) can be derived from the fact that F^{ω} is included in the least fixed point of F and, if an argument is acceptable regarding S, it also is acceptable regarding any superset of S. With (2), let every argument have a limited amount of defeaters. If $S_0 \subseteq ... \subseteq S_n \subseteq ...$ is a rising sequence of sets of arguments, and $S = S_0 \cup ... S_n \cup ...$. Let $A \in F(S)$. Given that there is a limited amount of arguments which defeat A, there is a number m so that $A \in F^m(S)$. Thus, $F(S) = F(S_0) \cup ... F(S_n) \cup ...$

It should be mentioned, that if (2) does not hold, $F^{\omega} \subset F(F^{\omega})$.

Example 2.3.5 Let $A_1,...,A_n,...$ be an infinite chain of arguments and $1 \le i \le \infty$. Let A_i be defeated by A_{i+1} .

 $A_1 \longleftarrow A_2 \longleftarrow A_3 \longleftarrow \cdots$

Because there is no undefeated argument, there is not least fixed point. Thus $F(\emptyset) = \emptyset$. As a side note it can be said, that here are two other fixed points fulfilling definition 2.2.2: the set A_i in which *i* is odd and A_i in which *i* is even.

Defensible arguments

The definition of justified arguments in the grounded semantics enables a distinction between two kinds of arguments that are not justified. If we look at example 2.2.3, we can see that even though that *B* defeats *A*, *A* is justified through the reinstatement through *C*. The next example is an addition to example 2.2.4: Example 2.3.6 (Zombie arguments)

Let *A*, *B*, *C* be arguments and *A* defeats *B*, *B* defeats *A* and *B* defeats *C*.



A specific example would be

A = 'Nixon was not a pacifist since he was a republican'

B = 'Nixon was a pacifist since he was a qaker, and he had no gun since he was a pacifist'

C = 'Nixon had a gun since he lived in Chicago'

Based on definition 2.3.3, non of the three arguments are justified. *A* and *B* are still the same as in example 2.2.4. *C* lastly is defeated by *B*. Contrary to example 2.2.3, *B* is not defeated by any justified argument, even though it is not justified. Thus *B* can still prohibit *C* from being justified: There is no justified argument reinstating *C* through defeating *B*. Arguments like *B* can be called 'zombie arguments': *B* is not justified (therefore not alive, but is not entirely dead either). *B* has a mediate state, where it is still able to influence other arguments.

We will now define that state of zombie arguments as 'defensible':

Definition 2.3.7 (Overruled and defensible arguments in grounded semantics)

In the scope of grounded semantics, an argument is:

- *overruled* if it is not justified and a justified argument defeats it.
- *defensible* if it is not justified and not overruled.

Self-defeating arguments

Let us now take a look on how definition 2.3.2 handles self-defeating arguments:

Example 2.3.8 Let *A*,*B* be two arguments, where *A* defeats *A* and *A* defeats *B*.



Here we again have $F(\emptyset) = \emptyset$. Both *A* and *B* are defensible, because neither of them is defeated by a justified argument. One might think that this is unnecessary, because self-defeating arguments ought to be overruled. In Chapter 4 one will see that a correct analysis of self-defeating arguments is only given through an explicit inner structure of arguments.

Unique status assignments: problems

The unique-assignment system can be executed in a mathematical way, so that it delivers intuitive results in various cases. With that come a few problems:

Example 2.3.9 (Floating arguments)

Let *A*, *B*, *C*, *D* be arguments, where *A* defeats *B*, *B* defeats *A*, *A* defeats *C*, *B* defeats *C* and *C* defeats *D*.



Because all of the arguments are defeated, all of them are defensible (through definition 2.3.3). One could on the other hand argue, that since *C* is defeated by *A* & *B*, *C* ought to be overruled. The cause for that is regarding *C*'s status, the conflict between *A* and *B* needs no solving. *C*'s status is 'floating' on that of *A* and *B*. If *C* might be overruled, then *D* would be justified, due to *C* being its only defeater.

A version of this is the following instance of default reasoning. Before we start we need to presume that the structure of arguments has a conclusion and they have subarguments.

Example 2.3.10 (Floating conclusions)

Let A^- , A, B^- , B be arguments, where A^- and B^- are defeating one another and A and B have the same conclusion.



A possible configuration would be

- A^{-} = Simon is German because he was born in Munich
- B^- = Simon is Italian because he has an Italian surname
- *A* = Simon likes playing soccer because he is Italian
- B = Simon likes playing soccer because he is German

The main issue no matter of $A^{-'s}$ and $B^{-'s}$ conflict, the conclusion will always be Simon likes playing soccer. With that it appears like it is justified to assume this conclusion as true, despite it not being supported by a justified argument. It sort of floats on the status of A^{-} and B^{-} .

Regardless of whether the unique assignment approach is able to conquer floating arguments and conclusions, one can do so by having multiple status assignments.

2.4 Relation between different semantics

Another approach for dealing with conflicting arguments of equal strength, is them producing two different status assignments: In that case one has to be justified to the harm of the other one. Now an argument is *justified* if it has this status in every status assignment. For this way there are two approaches called 'stable' and 'preferred semantics'.

Stable semantics

This first way takes definition 2.2.2 and uses the extension of multiple assignments:

Definition 2.4.1 (stable status assignments)

Let $AF = \langle Args, defeat \rangle$ be an AAF, where In and Out are two subsets of Args. Therefore (In,Out) is a *stable status assignment* regarding AF, if $In \cap Out = \emptyset$ and $In \cup Out = Args$ and for all $A \in Args$ it maintains that:

- (1) *A* is *in* (meaning $A \in In$), if every defeater of *A*, if there is one, is *out*.
- (2) *A* is *out* (meaning $A \in Out$), if there is a defeater of *A*, which is *in*.

These two terms (1) and (2) are taken from definition 2.2.2. With this definition 2.4.1, we can connect a stable status assignment to a *stable argument extension*, which holds every argument that is *in* the status assignment.

Definition 2.4.2 (Stable argument extension)

A set of arguments is called *stable argument extension*, if for a stable status assignment this is a set, where every argument has the status *in*.

Stable argument extensions correspond with Dung's *stable extensions* [24]. He in fact has a different but comparable notion of a conflict-free set of arguments:

Definition 2.4.3 (Conflict-free sets)

A set of arguments *S* is *conflict-free*, if there is no argument in *S* defeating another argument in *S*.

Definition 2.4.4 (Stable extensions)

A set of arguments *S* is a *stable extension*, if *S* is conflict-free and all arguments that are not in *S*, are defeated by *S*.

Proposition 2.4.5 The stable argument extensions generated by definition 2.4.1 are exactly the stable extensions specified in definition 2.4.4.

Proof. \Rightarrow : Let (*In*,*Out*) be a stable status assignment. To be verified:

1. *In* is conflict free. Through contradiction, we say that *In* holds arguments *A*,*B*, where *A* defeats *B*. By section (2) of definition 2.4.1, *B* is in *Out*. But given that $In \cap Out = \emptyset$, *B* is not in *In*. This is a contradiction. Therefore there is no *A*,*B*, where *In* is conflict-free.

14

2.4. RELATION BETWEEN DIFFERENT SEMANTICS

2. *In* defeats every argument outside *In*. Due to stable status assignments allocating a status to every argument in *Args* and $In \cap Out = \emptyset$, all arguments that are not in *In* are in *Out*. Subsequently by section (2) of definition 2.4.1 all those arguments are defeated by an argument in *In*.

⇐ : Let *S* be a stable extention. To be verified: (*S*,*Args*/*S*) is a stable status assignment. It should be mentioned, that this is a separation of *Args*, so $In \cap Out = \emptyset$ and $In \cup Out = Args$. Hence we need to prove that (1) and (2) from definition 2.3.1 are fulfilled.

1. (1) from definition 2.3.1 is fulfilled like this: If $A \in S$ (*S* is conflict free), then no $B \in S$ defeats *A*, and with that every defeater of *A* is in *Args/S*. And if every defeater of *A* is in *Args/S*, then *A* is not able to be in *Args/S*, given that no defeater of $A \in S$. Therefore *A* is in *S*.

2. (2) from definition 2.3.1 is fulfilled like this: Let $A \in Args/S$. Thus given that *S* defeats every argument around it, *A* has a defeater in *S*. And if *A* has a defeater in *S* (and *S* is conflict free), $A \in Args/S$.

From now on we ought to apply the term *stable extension* for stable argument extensions and Dung's stable extension.

Taking a look at two past examples, 2.2.3 only has one stable extension, namely $\{A,C\}$, but 2.2.4 has two, because of these two status assignments:



If we correctly remember, an argumentation system defines, if it is justified to accept an argument. How does that relate to *A* and *B* in example 2.2.4 ? Since these two are *in* a stable status assignment and not *out* in the other, it must be concluded that regarding the stable semantics none of the two are justified. This is because of the following definition:

Definition 2.4.6 (Justified arguments in stable semantics)

In reference to stable semantics, an argument is *justified*, if it is *in* every stable status assignment.

On the contrary, it is not enough: similar to the unique-status-assignment method, there can be a distinction between two diverse categories of not justified arguments. Various arguments are in no stable status assignment, however some are in extensions. The two categories are *overruled* and *defensible* arguments.

Definition 2.4.7 (Overruled and defensible arguments in stable semantics)

Regarding to stable semantics, an argument is:

- *overruled*, if it is *out* in every stable status assignment.
- *defensible*, if it is *in* in a few but not every stable status assignment.

That being said, the approach of unique- and multiple-assignments are not equivalent. In example 2.3.9 *A* and *B* make a defeat loop, which means (based on multiple assignment approach) *A* or *B* is assigned *in* but not both. The mentioned defeat relation produces two stable status assignments:



Although every argument in the unique-assignment approach is defensible, now, whilst *A*,*B* are defensible, *D* is justified and *C* overruled.

These multiple status assignments also allow capturing floating conclusions. This can be achieved through the definition of a justified formula ϕ , where every extension holds an argument for ϕ . The structure of arguments and other resulting notions for formulas will be further discussed in Chapter 4.

Preferred semantics

Another version of the multiple-status-assignments system is unnecessary, given that a stable extension is conflict-free, it displays a clear perspective. At the same time it is the best perspective, considering that an argument can only be either accepted or rejected. Stable semantics are in general the most 'offensive' type of semantics, seeing that a stable extension defeats all arguments outside of it. Here it does not matter if these arguments are or are not aggressive towards the extension.

Because of that, not every argumentation framework has stable extensions. The example below displays that and also has an 'odd loop' of defeat relations.

Example 2.4.8 (Odd loop)

Let *A*,*B*,*C* be arguments, where *A* defeats *C*, *C* defeats *B*, *B* defeats *A*.



Right here, definition 2.4.1 has a few problems, since this graphic has no stable status assignments.

1. If *A* is *in*, then *C* is out (*A* defeats *C*). If *C* is *out*, *B* is *in* but *A* is *out* (*B* defeats *A*). This is a contradiction.

2. If *A* is *out*, then *C* is *in* (*A* only defeater of *C*). Therefore *B* is *out* (*C* defeats *B*). However *A* is then *in* (*B* defeats *A*). This is a contradiction.

2.4. RELATION BETWEEN DIFFERENT SEMANTICS

It should be mentioned that self-defeating arguments are a special case. Therefore argumentation frameworks with a self-defeating argument might not hold a stable status assignment.

For these examples to have a multiple-assignment semantics, we need to define *partial* status assignments.

Definition 2.4.9 ((Preferred) status assignments)

Let $AF = \langle Args, defeat \rangle$ be an AAF and In, Out two subsets of Args. The tuple (In, Out) then is a *status assignment* according to AF, if $In \cap Out = \emptyset$ and $\forall A \in Args$:

- 1. *A* is *in* ($A \in Args$), if every defeater of *A* (if there is one) is *out*.
- 2. *A* is *out* ($A \in Args$), if *A* has a defeater that has the status *in*.

A *preferred* status assignment (*In*,*Out*) is a maximised set of arguments that is labelled *in* (if there is no status assignment (*In'*,*Out'*) where $In \subset In'$).

That being said, in example 2.4.8 the preferred semantics assigns it a unique preferred status assignment ((\emptyset , \emptyset)). The definitions 2.4.6 and 2.4.7 can as well be specified for preferred semantics, through the substitution of 'stable' by 'preferred'. In these preferred semantics there are other options for the definition of defensible and overruled argument, due to every status assignment of an argument status being able to be either *in*, *out* or undefined. Contrary to stable semantics, we can now have an argument being *in* in a few assignments but not *out* in any assignment.

Coming back to preferred extensions, it rather is defined with the notion of an admissable set (Dung [24]).

Definition 2.4.10 (conflict-free and admissable set)

- (1) Let a set of arguments *S* be *conflict-free*, if no argument in *S* defeats an argument in *S*.
- (2) Let a set of arguments *S* be *admissable*, if *S* is conflict-free and every argument in *S* is acceptable regarding *S*.

Taking a look at example 2.1.3, we can see that the set \emptyset , {*C*}, {*A*,*C*} are admissable but every other subset of {*A*,*B*,*C*} is not admissable.

Definition 2.4.11 (Preferred extension)

Let a conflict-free set of arguments *S* be a *preferred extension*, if *S* is the largest (regarding set inclusion) admissable set.

Preferred status assignments correlate exactly with these preferred extensions (Caminada (2006) [21]).

Proposition 2.4.12

1. If (*In*,*Out*) is a status assignment, then *In* is an admissable set.

2. Let Out(E) be a set of arguments, with every argument defeating *E*. If *E* now is a preferred extension, (E,Out(E)) is a status assignment.

3. (*In*,*Out*) is a preferred status assignment, if *In* is a preferred extension.

It can be derived from definition 2.4.11 that:

Proposition 2.4.13 (Dung (1995) [24])

Any AAF has a minimum of one preferred extension.

Grounded status assignments

Grounded semantics can be written as status assignments, where the assignments are minimal towards the following.

Definition 2.4.14 (Minimal status assignments)

Let *S* be a status assignment (*In*,*Out*) that is *minimal*, if there exists no status assignment S' = (In',Out'), where $In' \cup Out' \subset In \cup Out$.

Proposition 2.4.15 (Caminada (2006) [21])

Let *S* be the grounded extension of *AF*, if (*S*,*Out*) is a grounded status assignment of *AF*.

Self-defeat in preferred semantics

Finally in what way do preferred semantics process self-defeating arguments? They (same as in grounded semantics) actually hinder other arguments from being justified. For a more illustrative perspective, one can take a look at example 2.1.8. *B* is not admissable and with that \emptyset is the only preferred extension. As already mentioned an entire analysis of self-defeat with the need of a clear internal structure of arguments is made in Chapter 4.

Relationship between grounded, stable and preferred semantics

In the following a presentation of various results on the relation between these semantics proven by Dung (1995) [24] is given.

Proposition 2.4.16

Every stable extension is preferred, but some preferred extensions are not stable.

Proof. First of all, every stable extension is a preferred extension plus example 2.3.8 proves that conversely it does not work. An empty set is a preferred extension of that *AF*, but it is not stable.

These next results are being listed without a proof.

(1) The grounded extension is included in the junction of every preferred extension.

18

- (2) If an *AAF* does not generate infinite paths $A_1,...,A_n,...$ due to the defeat graph, so that A_{i+1} defeats A_i then it holds only one stable extension, that is also grounded and preferred.
- (3) Lastly Dung determined certain conditions, where preferred and stable semantics correspond. One vital condition would be that an *AAF* does not hold odd defeat loops.

To close out this chapter a comparison will be drawn between the unique- and multipleassignment approach. One could say that the difference between these two approaches illustrates the contrast between the 'skeptical' and 'credulous' view. If there is an unsolvable conflict, a 'skeptic' avoids coming to a conclusion where a 'credulous' thinker would draw an arbitrary conclusion and dive into their impacts. The comparison 'skeptical'-'credulous' and unique- and multiple-status-assignment are independent from eachother: If there is a decision on accepting a justified belief, it is important how the certain arguments are assessed in the end, given these assignments. This assessment entails the notions of 'justified' and 'defensible', which encaptures the contrast towards 'skeptical'-'credulous'. Therefore the contrast of justified-defensible arguments can be applied in both the unique- and multiple-assignment approach, which makes these approaches unrelated to the 'skeptical'-'credulous' contrast.

Using skeptical reasoning is advocated by saying there are two arguments of equal strength, whereas using the credulous reasoning could be advocated by saying one person has to act, no matter if s/he has clear reason to evaluate on what to act on. Taking a look at their result, both approaches basically vary in their handling of floating arguments and conclusions. Regarding these instances, one does not know which approach is the right one. Contrary to 'right' and 'wrong', it is often spoken of 'senses', where there can be a justified argument/conclusion. In example 2.3.10 where Simon likes soccer is justified in a different sense than example 2.2.3, where the result is Tweety flies.

2.5 Argument-based reconstruction of nonmonotonic logics

Dung's *AAF* is not limited towards argumentation systems, but can also reformulate other nonmonotonic logics in argument-based circumstances. Thereby one can match these logics regarding general theory: one can study the differences between them and what these differences lead to. With that one can also derive other versions for these logics.

Let us take the default logic, which will be similarly rebuilt according to Dung [24]. One could do that by the definition of an argument being a bounded *process* as defined in Antoniou [5]. It is said that processes are sequences of defaults without multiple occurrences, where the precondition of every default is logically implied through the union of the knowledge *W* and the resultants of every prior default in that sequence. A process *closed*, if there can be no default attached to the sequence and it is *successful*, if every premise is compatible with the result. Since arguments are generally constructed for a proof of a conclusion, processes now being arguments have no need of being closed. There is no need for being successful, since unsuccessful correlates to being a self-defeated argument.

Definition 2.5.1

Let $\Delta = (W, D)$ be a default theory and let the *AAF AF*(Δ) = $\langle \text{Args}_{\Delta}, \text{defeat}_{\Delta} \rangle$ be defined by:

• $Args_{\Delta} = \{ \Pi | \Pi \text{ is a bounded process of } \Delta \}$

• Π defeats Λ Π' , if $\varphi \in In(\Pi)$ for a $\varphi \in Out(\Pi')$.

A formula φ is a *result* of an argument Π , if $\varphi \in In(\Pi)$.

With that a defeat of an argument can be achieved through acquiring the negation of those assumptions.

With this translation, an analogy can be shown between default logic and stable semantics. Let Δ be a default theory, where

E is a given set of formulas, let Args(E) be the set of every $\Pi \in Args_{\Delta}$, where for every $k \in Out(\Pi)$: $\neg k \cup E$ is constant.

 $S \subseteq Args_{\Delta}$ is a given set, let Concs(S) be the union of every set $In(\Pi_i)$, where $\Pi_i \in S$.

Proposition 2.5.2

For a given default theory Δ :

- Let *S* be a stable extension of *AF*(Δ), let then *Concs*(*S*) be a Reiter-extension of Δ.
- (2) Let *E* be a Reiter-extension of Δ , let then Args(E) be a stable extension of $AF(\Delta)$.

I will not display this proposition's proof, since it is not stated in the literature. In the following Lemma following holds: let *d* be a default, then *Pre(d)*, *Jus(d)* and *Res(d)* indicate *d*'s precondition, justification and resultant.

Lemma 2.5.3

Let *S* be a stable extension of $AF(\Delta)$ and $\Pi \in S$, then:

- (1) every subsequence Π' of Π that is an argument is in *S*
- (2) every argument in *S* is a process.

Proof. (1): Every defeater of Π' is also a defeater of Π . Therefore (Π')'s defeater is not in *S*. Then (by definition of stable extension) $\Pi' \in S$.

(2): Let $\Pi \in S$ and Π is no process. Then, for a subsequence $\Pi[i]$ of Π and $d_i \in \Pi$, the negation of a certain $j \in Jus(d_i)$ is in $In(\Pi[i])$. Therefore $\Pi[i]$ is the defeater of Π . On the other hand, $\Pi[i] \in S$ (1). But *S* is not conflict-free. This is a contradiction.

Example 2.5.4

Let $\Delta_1 = (W, D)$ where $W = \{p\}$ and

$$D = \left\{ d_1 : \frac{p:q \wedge r}{q}, \quad d_2 : \frac{q:s}{t}, \quad d_3 : \frac{p:u \wedge \neg t}{\neg t} \right\}$$

Let $AF(\Delta_1)$ consist of:

20

 $\begin{array}{l} A = \varnothing \\ B = d_1 \\ C = d_1, d_2 \\ D = d_3 \\ E = d_1, d_3 \\ F = d_3, d_1 \\ G = d_1, d_3, d_2 \\ H = d_3, d_2, d_1 \end{array}$

The following figure shows the defeat releation and implies that G, H defeat every other argument except A.



In the end, Dung's work was a significant creation in the field of defeasible argumentation. It included a clear general framework for investigating the different argumentation systems. A great benefit is that it can be used on diverse nonmonotonic logics, since he showed how to transpose them. His proof theories for different argument-based semantics can be utilized towards the systems that are instances of these semantics. The abstract essence of Dung's framework gives room for the particular developer, where they can define the internal structure of an argument, how the arguments conflict and the source of the defeat relation.

In the third chapter a particular framework will be displayed, where these elements have been specified.

CHAPTER 3

Argumentation Logic: Applications

Up til now, the focus was on how to define the basic properties of *sets* of arguments, where we excluded methods of identifying if an argument is included in a certain set. In the following chapter a more formal way of argumentation is displayed. Here we will study the status of *individual* arguments.

3.1 General ideas

Let there be an argument in an *AAF*, how can its status be studied? This question has been taken on by various argumentation systems, where the basic concept can be explained as follows: Let there be an argument game amongst two parties or players, a pro- and an opponent of an argument. In a dispute the two players have alternating moves. The proponent begins with an argument, that needs to be tested, and every succeeding move entails an argument defeating a certain move from the opposing player. The starting argument holds a specific dialectical status, if the proponent wins after any given move from the opponent (winning strategy).

The explicit rules of the game rely on the game semantics. A general winning measure is if one player has no moves, the other one wins. On the other hand, there are many more:

- Do there have to be only strictly defeating or also weakly defeating moves ?
- Can one repeat a move ?
- Can a player backtrack?
- Can a player defeat or be defeated by a preceding move of his?

These are individual question that need to be resolved independently for each side. A basic concept of dialectical proof theories is dialectical asymmetry. Each player of an argument game has a different goal: the proponent tries to build a dialectical proof, whereas the opponent tries to impede that. Generally speaking the proponent is constructive and the oppponent destructive, which results in certain rules for both sides. With that comes a liability that affects one player more than the other. Which one can be figured out through the reasoning: if there is a skeptical reasoning the proponent has more of a liability, whereas in a credulous reasoning the opponent has more of a liability.

To be more concrete, a dialectical proof theory has the structure of an argument game controlling a *dispute* between two *players/partys*, the proponent *P* and the opponent *O* of an argument. Let *p* be a player and \overline{p} the other player. The players act in alternating *moves*, moving one argument every turn. There is a *protocol* function included for specifying the *legality* of moves. It does so, by specifying at every point in a dispute which arguments can be moved. In the end, a *winning criterion* is a partial function that identifies the winner of a dispute, if there is one. This argument game is a *zero-sum game*, where if there is a winner, the other player has to lose.

All of these notions are now being defined in this manner:

Definition 3.1.1 (Moves, dispute and protocol)

Let $AF = \langle Args, defeat \rangle$ be an argumentation framework, then the following holds:

- Let *M* be a set of *moves* consisting of all pairs (p, A), where $p \in \{P, O\}$ and $A \in Args$; (p, A) is a move *M* and we can represent *p* through pl(m) and *A* through s(m).
- Let *M* ≤ ∞ be a set of *disputes* and a set of every sequence of *M* and *M* <∞ the set of *finite disputes* and the set of every finite sequence from *M*.
- A function, which identifies every *legal move* at every stage of a dispute, is called a *protocol*. Mathematically, a function Pr with a domain that is a nonempty subset D of $M^{<\infty}$ and that takes subsets of M as values, is called a protocol.

 $Pr: D \longrightarrow Pow(M)$

where $D \subseteq M^{<\infty}$. *D*'s elements are so-called *legal finite disputes*. The elements included in Pr(d) are the so-called moves allowed after *d*. If *d* is a legal dispute and $Pr(d) = \emptyset$, then *d* is a *terminated* dispute. For every finite dispute *d* and move *m*, *Pr* needs to fulfill the following conditions:

- (1) If $d,m \in D$, then $d \in D$ and $m \in Pr(d)$;
- (2) If $m \in Pr(d)$, then if *d* is of even length pl(m) = P. Otherwise pl(m) = O.
- A partial function of type $W: D \longrightarrow \{P, O\}$.

The essential parts of this definition are the protocol and the winning criterion, because the dialectical proof theories deviate in these two elements. In the following, we shall define an abstract game-theoretic notion of defeasible provability, which is done so through the notion of a strategy. A strategy for a certain player is displayed in a game through a tree of disputes, where for every available move from the opposing player an individual answer is given.

Definition 3.1.2 (Strategy)

- Let *p* be a player, where his *strategy* is a tree of disputes only after *p*'s moves, and containing all of *p*'s legal replies.
- (2) Let *p* have a *winning* strategy, if *p* wins every dispute in that strategy.

When defining a winning criterion, that the other player has no legal moves, one can see that a winning strategy is where a player has every last move of every branch in that tree. Defined by a protocol *X*, defeasible provability is being defined here:

Definition 3.1.3 (Provability)

Let *A* be a *defeasibly provable* argument (regarding X-game), if the proponent has winning strategy in a dispute, where *A* is the root and fulfills the protocol of *X*.

3.2 Dialectics for grounded and preferred semantics

In the following section the proof theory, that assesses if a specific argument is in the grounded extension of a given *AF*, is being discussed. The dialectical asymmetry benefits the opponent, since only justified arguments are in the grounded extension. Furthermore the proponent cannot repeat his arguments and backtracking is prohibited for both players.

Definition 3.2.1 (Proof theory for grounded semantics)

Let a dispute fulfill the *G*-game protocol, if these conditions are fulfilled:

- (1) Moves *m* are legal if besides definition 3.1.1 they fulfill these conditions:
 - (a) The proponent does not repeat one of his moves; plus
 - (b) The moves from the proponent (not his first) strictly defeat the last move from the opponent; and
 - (c) The move from the opponent defeats the last move from the proponent.
- (2) A player has won a dispute, if the opposing player has no legal moves.

A dispute that fulfills the *G*-game protocol is a so-called *G*-dispute.

Example 3.2.2 Let *A*, *B*, *C*, *D* be arguments, where *B*, *D* defeat *A* and *C* defeats *B*. Then a *G*-dispute may run on *A* like:

P : *A*, *O*: *P*, *P*: *C*

P in this dispute is trying to show that *A* is justified. *B*, *D* both defeat *A*, which leads to *O* having two choices to respond towards *A*. *O* decides to respond with *B*, and then *C* is the only argument defeating *B* (*P* cannot use *C* in third move). Now there are no arguments against *C*, where *O* cannot move and lose as a consequence.

On the other hand, this is not a certain result for *O*; that loss was due to her inefficient play. An optimal strategy for *O* would be

P: A, O: D

With that *P* has no answer and *O* wins. In the end *P* has no winning strategy. *P* can only win the first dispute if *O* chooses the wrong argument (*B*) as a response towards *A*. *O* is actually able to win every game, given the right moves are made. *O* has a winning strategy.

Example 3.2.3

Let there be the two strategies for *P* (shown below), where the left tree based upon an AF_1 with $Args = \{A, B, C, D, E, F, G\}$ and *defeat* illustrated through the arrows. Since *O* in the end has no move in every dispute, *P* has a winning strategy. An argument *A* is provable regarding AF_1 . The tree on the right hand side is built according to an extension of AF_1 into AF_2 , because of the addition of *H*, *I*, *J* to Args and the corresponding defeat relations (extension is in the dotted box). Since *O* can win a dispute, this is not a winning strategy for *P*, so *A* in this context is not provable regarding AF_2 . Let us take a look at an example from Prakken [59]:



The left hand figure display "A is provable" and the right hand figure "A is not provable". There is an important point about the non-repitition (definition 3.2.1 (section 1a))): This condition does not influence the provability of an argument (O can reply with its own move a second time), but it prohibits infinite disputes given a finite set of *Args*, which is useful for computational purposes. This point also applies to the condition, where P's arguments need to be strictly defeating. If an argument is simply defeating, it does not change the provability but might cause infinite disputes.

The following proposition is made for the relation between grounded semantics and its proof theory:

Proposition 3.2.4 (Wholeness and completeness of G-game)

Let an argument be in the grounded extension of an *AF*, if it is defeasibly provable based on *AF* in the *G*-game.

As a side note, completeness does not insinuate semi-decidability (that there is an algorithm that can genereate a provable formula). If constructing certain arguments is logically not decidable, then searching for counterarguments is generally not semi-decidable, due to the search being a check.

Now that the discourse about dialectical proof theory for grounded semantics is completed, we now take a look at a dialectical proof theory for credulous reasoning, especially for pre-

ferred semantics.

Dialectics for preferred semantics

In the following, I will display a so-called *P*-game (the *P* has no relation to the proponent *P*). This *P*-game was introduced by Vreeswijk and Prakken (2000) [78], and is a credulous proof theory for preferred semantics. As a small notational correction, we will now indicate a defeat relation with \leftarrow . The following graph aswell as the following notions of *straight failure, straight success, even loop success, odd loop failure* and *backtracking* are taken from their scripts:

Example 3.2.5

Let $\mathcal{A} = \langle X, \leftarrow \rangle$ be a pair with arguments

 $X = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,p,q\}$

and \leftarrow as shown in the figure below an example of an *AAF*. It illustrates various cases, and will consequently be used as a running example throughout this chapter.



Figure 3.1: Defeat relations in the running examples

Let us now take a closer look at *a* and check if it is preferred, in other words if it is included in a preferred extension. We know that it is enough to prove that the argument is admissable. We start $S = \{a\}$ and, if *a* has defeaters, finding additional arguments to form *S* into an admissible set.

Example 3.2.6 (Straight failure)

Let us take the argument system from figure 3.1 and *P*'s duty is to prove that *a* is preferred. Firstly *P* puts forward *a*:



If there is no defeater of *a*, *S* = *a* is admissable, and *P* wins. On the contrary, because of $a \leftarrow h$, *O* sends *h*:

P now has to defend *a* through searching for an argument against *h*. Since there is no such argument, *P* cannot build a admissable for *a* and with that *a* is not preferred.

a

Example 3.2.7 (Straight success)

Let *P* prove that *b* is admissable. Firstly, *P* puts forward *b*:



O can defeat *b* through *d*:



P can defend this with *g*:



With the fail of *O*'s attack on *b*, *O* attacks this time with *e*:

28



P can defend *b* again, through *h*. Because *O* has no other legal argument against *b*, *P* can now close *S*:



Example 3.2.8 (Even loop success)

Let *P* try to prove that *f* is admissable:

Firstly *P* puts forward *f*:



O then defeats *f* with *n*:

There after *P* defends this attack through *i*:



Next *O* defeats *i* with *j*:



Here *P* defends *i* with *i* (*i* is then self-defending). This is bad for *O*, since there are no further legal arguments defeating *f* or *i*:



Example 3.2.8 displays a need for *P* to be able to repeat his arguments, while *O* should not be able to repeat his arguments (at the minimum in that "line of dispute").

Example 3.2.9 (Odd loop failure)

Let *P* try to prove that *m* is admissable:

Firstly *P* puts forward *m*:



O then defeats *m* with *l*:



There after *P* defends this attack through *p*:



Next *O* defeats *p* with *h*:



Here *P* backtracks and withdraws *p* from *S*. Instead he defends *l* with *k*:



Now *O* can defeat *k* with *m* (this displays an inconsistency in *S*):



P here has no argument against *l* and *m*, and with that cannot close *S* to an admissable set. *m* is not contained in an admissable set. *P* in this situation cannot reply with *l* towards *m*, because then the constructed set around *m* is not conflict free (and then not admissable). Concluding *P* cannot be allowed to repeat *O*'s moves. It is also shows that *O* should be able to repeat *P*'s moves (forces conflict in *P*'s position).

Example 3.2.10 (Backtracking)



In this figure we can see that O needs to be able to backtrack. Let P start with a, O defeats that with d and P defends a with e. Should O defeat e with b, P can defend that attack by replying with e itself. On the other hand O can backtrack to a, and defeat it with c, where P can defend a only through b and thus repeats O. We did come to the conclusion in Example 3.2.9 that P should not be able to backtrack, so if O backtracks, it can show P's line of argumentation is not conflict-free.

Here is a summary on who should be able to repeat moves:

Proposition 3.2.11

- (1) *P* should be able to repeat itself (if available), since *O* might not find a defeater for *P*'s repeated argument. In that case *P*'s repitition completes a circle with even length, where *P* has admissable arguments.
- (2) *O* should be able to repeat *P* (if available), since that would display that *P*'s arguments are not conflict free.
- (3) *P* should not be able to repeat *O* since that would create a conflict in *P*'s arguments.
- (4) *O* should not be able to itself, since *P* has already defeated this particular argument.

3.3 Simplification of the P-game

In this section, the argument game will be defined for the preferred semantics:

- A *dispute line* is a dispute without backtracked moves
- An eo ipso is a previous argument from the other player

Definition 3.3.1 (Proof theory for preferred semantics)

Let there be a dispute, that fulfills the *P*-game protocol if the following holds:

- (1) In addition to definition 3.1.1, moves are considered illegal, if
 - (a) a move from *P* responds to a previous move from *O*
 - (b) a move from *O* responds to a past move from *P*
 - (c) a move defeats an argument that it replies to
 - (d) *P* does not reiterate *O*'s moves
 - (e) O does not repeat one of its moves from that dispute line
 - (f) there a two moves to a particular move that have the same content
- (2) *O* wins the dispute, if he does an *eo ipso* or *P* has no legal moves. Other than that *P* wins.

If a dispute fulfills these rules and conditions of the the *P*-game it is called a *P*-dispute.

Another point is that if there is an infinite dispute, *P* has won automatically.

Because the *P*-game gives *O* the opportunity to backtrack, whilst a *P*-dispute, a tree of possible dispute lines is being built. In general, we have two ways of displaying a *P*-dispute: through a *linear* structure, where the arguments are listed according to when they were moved, and through a *tree* structure, where every edge shows which argument replies to which. There should not be a misconception between a dispute in tree form and a strategy in tree form: in the definition 3.1.2, a so-called latter tree has edges, that are between two arguments and displays that the child argument is the reply to the parent argument.

Proposition 3.3.2 (Wholeness and completeness of the *P*-game)

Let an argument be in a preferred extension of an *AF*, if it is defeasibly provable regarding *AF* in the *P*-game.

Proof. In the following, an argument *a* is *defended* in a dispute, if that dispute starts with *a* and *P* wins it. Through the definition of preferred extensions it is enough to display that an argument is admissable, if it is able to defend in every dispute.

So firstly let *a* be defended in all disputes, even in those where *O* has contradicted ideally. E.g. let *A* be the defending argument of *a* that is being used by *P*. Should *A* not be conflict-free then for a certain a_i , $a_j \in A$, $a_i \leftarrow a_j$, and *O* might have done an *eo ipso*, which is not true. Should *A* not be admissable, then $a_i \leftarrow b$ for certain $a_i \in A$ whilst $b \rightarrow A$. Here

3.3. SIMPLIFICATION OF THE P-GAME

O would have utilized b as a winning argument, which does not happen. Therefore A is admissable.

In contrast, let $a \in A$ and A admissable. P is now able to win every dispute when starting with a and using only arguments from A as replies. Given that P uses only A's arguments, O is not able to win through *eo ipso*, since A is conflict-free. a can with that be defended in that dispute.

Lastly a disadvantage of the *P*-game is that proofs are able to be infinite. In particular if an argument has an infinite number of defeaters, but also other proofs as shown in example 2.3.5. In any case one can prove that with a finite set of arguments every proof is finite.

CHAPTER 3. ARGUMENTATION LOGIC: APPLICATIONS

CHAPTER 4

Argumentation Logic: Argumentation with structured arguments

As already mentioned, Dung's abstract framework was a major development in the field of argumentation. Despite that, its fully abstract nature leads to the framework not being suitable for displaying certain argumentation problems. It is rather suited for analysing argumentation formalisms and developing a metatheory of those systems. If existing applications of argumentation-based inference have to be formed, Dung's approach needs to be improved, specifically the argument's structure and the defeat relation. The following chapter will display Dung's method through taking an undefined logical language and establish arguments in view of inference trees. These trees are made through using two different inference rules: deductive/strict and defeasible rules. As already mentioned in Section 1.3, the notion of an argument through an inference tree gives three possible ways to attack an argument: attacking an inference, a premise or a conclusion. For a resolve for these conflicts, we need priorities, that lead to three kinds of defeat: undermining, rebutting and undercutting defeat. Before we start describing them, we need to make presumptions on that logical language, specifically that well-formed formulas are opposite of other wellformed formulas. Aside from being abstract, the framework can be used for any set of inference rules, if it is separated into strict and defeasible, and for every logical language with an opposite relation.

A very prominent example here is the European ASPIC project [3],[4]. It is predicated on Pollock's [57],[55] and Vreeswijk's [75],[76],[77] structure of arguments, Pollock's defeat relation and work from various scientist in the field of argumentation with prioritised rules. Certain proofs from the following can be found in [62].

4.1 Argumentation systems with structured arguments

In the upcoming section we will redefine the arguments from Dung's argumentation framework and its defeat relation in a structural context (the defeat relation will also have external priority/preference information). The succeeding framework combines two possibilities on how to conquer the defeasibility of reasoning. Others like Bondarenko [19] developed the defeasibility of arguments in the incertitude of their premises, through which attacks are only on the argument's premises. On the other hand, Pollock [55] and Vreeswijk [77] developed the defeasibility of arguments with the uncertainty of their inference rules: the inference rules in these certain logics are either deductive or defeasible, and arguments attacks can only be on the argument's applications of defeasible inference rules. Vreeswijk [76] names these two views *plausible* and *defeasible* reasoning. In more detail he portrays plausible reasoning as sound (e.g. deductive) reasoning on a precarious basis, and defeasible reasoning as unsound (and yet rational) reasoning on a stable basis. The current framework again unites both forms of reasoning, however inside the abstract setting of Dung [24].

The fundamental notion of the current framework is an argumentation system, that continues the notion of a proof system with a separation between strict and defeasible inference rules and a priority ordering on the defeasible inference rules.

Definition 4.1.1 (Argumentation system)

Let there be an *argumentation system* $AS = (\mathcal{L}, -, \mathcal{R}, \leq)$ where

- (1) \mathcal{L} is a logical language,
- (2) $^{-}$ is a contraposition function from \mathcal{L} to $2^{\mathcal{L}}$
- (3) $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict (\mathcal{R}_s) and defeasible (\mathcal{R}_d) inference rules, where $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$,
- (4) \leq is a partial preorder on \mathcal{R}_d .

The syntax of $2^{\mathcal{L}}$ describes the powerset (set of all its subsets) of \mathcal{L} . Amgoud [3] and Caminada & Amgoud [4] suspect that an argument is given through a certain logical unspecified language (except for being closed under classical negation). In the following, this will be universalized in two ways: firstly, asymmetric conflict relations amongst formulas are possible (e.g. contraposition relation from Bondarenko [19]); Secondly, on top of the classical negation, other symmetric conflict relations are possible (e.g. the formulas "mouse" and "cat" can be declared contradictory without an axiom \neg (mouse \cap cat).

Definition 4.1.2 (Logical language)

Let \mathcal{L} (a set) be a logical language and $\overline{}$ a contraposition function from \mathcal{L} to $2^{\mathcal{L}}$. If $\phi \in \overline{\psi}$ and then if $\psi \in \overline{\phi}$, ϕ is a *contrary* to ψ , else ϕ and ψ are *contradictory*. The last case is written as $\phi = -\psi$.

If there is no alternative specification, the contraposition function matches for simpleness with the classical negation. So if ϕ does not begin with a negation then $\neg \phi \in \overline{\phi}$ whilst if ϕ is in the nature of $\neg \psi$ then $\psi \in \overline{\phi}$. Since the notion of negation has been introduced, the same will be done with the notion of consistency:

Definition 4.1.3 (Consistent set)

Let $\mathcal{P} \subseteq \mathcal{L}$ be a *consistent* set, if $\nexists \psi, \phi \in \mathcal{P}$ where $\psi \in \overline{\phi}$, if not it is *inconsistent*.

It should be mentioned, that this is a short form of consistency, dependent on a set including contrary or contradictory formulas. Amgoud and Caminada [4] named this *direct consistency*, and *indirect consistency* the consistency of a closed set under strict inference.

Definition 4.1.4 (Strict and defeasible rules)

Let \mathcal{L} be the set of $\phi_1, ..., \phi_n, \phi$.

- If $\phi_1, ..., \phi_n$ hold then with no exception ϕ holds ($\phi_1, ..., \phi_n \rightarrow \phi$ strict rule).
- If $\phi_1,...,\phi_n$ hold then *probably* ϕ holds ($\phi_1,...,\phi_n \Rightarrow \phi$ *defeasible rule*).

Antecedents of the rule are $\phi_1, ..., \phi_n$ and ϕ its *consequent*.

As it is common in logic, the inference rules are often specified through schemes, where a rule's antecedents and consequent are *metavariables* covering \mathcal{L} . If an *AS* has standard propositional and/or first-order logic, the strict rules could have every valid propositional or first-order inferences. For example:

 $\phi, \psi \to \phi \land \psi \quad (\forall \phi, \psi \text{ propositional formulas}) \\ \forall x P x \to P a \quad (\forall \text{ predicate } P, \text{ constant } a)$

Potential defeasible inference rules can be found in Section 4.2: it is stated there, that the core choice is if these rules are domain-specific or expressing general patterns of reasoning. Arguments are made from a knowledge base, that is expected to have three formulas:

Definition 4.1.5 (Knowledge base)

Let (\mathcal{K}, \leq') be a *knowledge base* of an argumentation system $(\mathcal{L}, {}^-, \mathcal{R}, \leq)$, where $\mathcal{K} \subseteq \mathcal{L}$ and \leq' is a partial preorder on $\mathcal{K} \setminus K_n$. Now $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a$ where each subset is disjoint and

- \mathcal{K}_n is a set of *axioms*; So, an argument cannot be attacked on its axiom's premises.
- *K*_p is a set of *premises*; So, an attack on an argument's ordinary premises is possible, and a defeat is dependent on the comparison of the attacker's and the attacked premise.
- *K*_a is a set of *assumptions*; So, an attack on an argument's assumption is possible, and these attacks are guaranteed to succeed.

In the following the arguments of such a knowledge base are being defined. These arguments can be built through gradually concatenating inference rules into trees. With that arguments have subarguments, that promote halfway conclusions. For a certain argument, there is a function Prem, Conc, Sub, DefRules and TopRule. Prem is a function returning every formula from \mathcal{K} (premises) that constructed the argument, Conc returns the result \conclusion, Sub returns each subargument, DefRules returns every defeasible rule of that argument and TopRule returns the most recent inference rule exercised in that argument.

Definition 4.1.6 (Argument)

Let an *argument* A regarding a knowledge base (\mathcal{K}, \leq') in an argumentation system ($\mathcal{L}, \neg, \mathcal{R}, \leq$) be:

(1) ϕ if $\phi \in \mathcal{K}$ with: Prem(A) = { ϕ }, Conc(A) = ϕ , Sub(A) = ϕ , DefRules(A) = \emptyset , TopRule(A) = undefined.

- (2) $A_1,...,A_n \rightarrow \psi$ if $A_1,...,A_n$ are arguments, where there is a strict rule $Conc(A_1),...,Conc(A_n) \rightarrow \psi$ in \mathcal{R}_s . $Prem(A) = Prem(A_1) \cup ... \cup Prem(A_n)$, $Conc(A) = \psi$, $Sub(A) = Sub(A_1) \cup ... \cup Sub(A_n) \cup \{A\}$, $DefRules(A) = DefRules(A_1) \cup ... \cup DefRules(A_n)$, $TopRule(A) = Conc(A_1),...,Conc(A_n) \rightarrow \psi$.
- (3) $A_1,...,A_n \Rightarrow \psi$ if $A_1,...,A_n$ are arguments, where there is a strict rule $Conc(A_1),...,Conc(A_n) \Rightarrow \psi$ in \mathcal{R}_d . $Prem(A) = Prem(A_1) \cup ... \cup Prem(A_n)$, $Conc(A) = \psi$, $Sub(A) = Sub(A_1) \cup ... \cup Sub(A_n) \cup \{A\}$, $DefRules(A) = DefRules(A_1) \cup ... \cup DefRules(A_n) \cup \{Conc(A_1),...,Conc(A_n) \Rightarrow \psi\}$, $TopRule(A) = Conc(A_1),...,Conc(A_n) \Rightarrow \psi$.

Example 4.1.7

Let there be a knowledge base from an argumentation system where

$$\begin{split} \mathcal{R}_s &\supseteq \{p,q \rightarrow; u,v \rightarrow w\} \\ \mathcal{R}_d &\supseteq \{p \Rightarrow t; s,r,t \Rightarrow v\} \\ \mathcal{K}_n &\supseteq \{q\} \\ \mathcal{K}_p &\supseteq \{p,u\} \\ \mathcal{K}_a &\supseteq \{r\} \end{split}$$

With that we have the following

 $\begin{aligned} & \texttt{Prem}(A_8) = \{p,q,r,u\} \\ & \texttt{Conc}(A_8) = w \\ & \texttt{Sub}(A_8) = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\} \\ & \texttt{DefRules}(A_8) = \{p \Rightarrow t; s, r, t \Rightarrow v\} \\ & \texttt{TopRules}(A_8) = v, u \to w \end{aligned}$

Separating two inference rules and three different premises leads to a separation of four kinds of arguments:

Definition 4.1.8 (Argument properties)

Let an argument A be

- strict if DefRules(A) = Ø;
- defeasible if DefRules(A) ≠ Ø;
- *firm* if $Prem(A) \subseteq \mathcal{K}_n$;
- *plausible* if $Prem(A) \nsubseteq \mathcal{K}_n$.

38

If there is a strict argument for ϕ with every premise extracted from *S*, we write $S \vdash \phi$. On the other hand if there is a defeasible argument for ϕ with every premise extracted from *S*, we write $S \mid \sim \phi$.

For the next part we will now take on the ordering of those arguments. \leq will be a partial preorder, where $A \leq B$ represents '*B* is at least as good as *A*'. *A* < *B* therefore displays $A \leq B$ and $B \leq A$.

Definition 4.1.10 (Admissable argument orderings)

Let A be a set of arguments and a partial preorder \leq , where \leq on A is a *admissable argument preordering (aap)*,

(1) if *A* is both firm and strict, and *B* is either defeasible or plausible, then B < A;

(2) if $A = A_1, ..., A_n \rightarrow \psi$ then for every $1 \le i \le n, A \le A_i$ and for a few $1 \le i \le n, A_i \le A$.

Definition 4.1.11 (Argumentation theories)

Let $AT = (AS, KB, \leq)$ be an *argumentation theory* where AS is an argumentation system, KB is a knowledge base in AS and \leq is an admissable ordering.

Definition 4.1.12 (Last defeasible rules)

Let there be an argument *A*.

- LastDefRules(A) = ∅, if DefRules(A) = ∅
- If A = A₁,..., A_n ⇒ φ, then LastDefRules(A) = {Conc(A₁),..., Conc(A_n) ⇒ φ}, otherwise LastDefRules(A) = LastDefRules(A₁) ∪ ... ∪ LastDefRules(A_n).

This definition can now be used for comparing two arguments:

Definition 4.1.13 (Last link principle)

Let *A*,*B* be arguments, where $A \prec B$, if either

- (1) section 1 of 4.1.10 holds or
- (2) LastDefRules(A) \prec_s LastDefRules(B)
- (3) LastDefRules(A) and LastDefRules(B) are empty and Prem(A) ≺_s Prem(B).

Example 4.1.14 (from Prakken et. al [15])

Let $\mathcal{K}_{p} = \{Snores; Professor\}, \mathcal{R}_{d} = \{Snores \Rightarrow_{r1} Misbehaves; Misbehaves \Rightarrow_{r2} AccessDenied; Professor \Rightarrow_{r3} \neg AccessDenied\}.$

Let us presume that *Snores* <' *Professor* and $r_1 < r_2$, $r_1 < r_3$, $r_3 < r_2$ and let us take the following arguments:

 A_1 : Snores B_1 : Professor A_2 : $A_1 \Rightarrow$ Misbehaves B_2 : $B_1 \Rightarrow \neg AccessDenied$ A_3 : $A_2 \Rightarrow AccessDenied$

We now use the ordering on A_3 and B_2 . The sets for that are LastDefRules(A_3) = { r_2 } and LastDefRules(B_2) = { r_3 }. Since $r_3 < r_2$, we have $B_2 \prec_s A_3$.

Not the last but every uncertain element in an argument is captured by the **weakest-link principle**. *A* here is preferred over *B*, if *A* is preferred to *B* on both of their premises and their defeasible rules.

Definition 4.1.15 (Weakest link principle)

Let *A*,*B* be two arguments. Then A < B, if either the first condition of Definition 4.1.10 holds or

- (1) $Prem(A) \prec_s Prem(B)$; and
- (2) If $DefRules(B) \neq \emptyset$ then $DefRules(A) \prec_s DefRules(B)$.

Attack and defeat

In this section, the thesis discusses the notion of defeat, in the context of two more components: non-evaluative syntactic notions of attack and the preference relation on arguments. To put it simple, defeat is defined through an attack with preference. Looking back at chapter 1, arguments were in inference trees and with that three syntactic forms of attack were available (attack on premise, on conclusion and on inference). These notions will be now combined with the preference order.

First we define how arguments can be attacked. Attacks can be categorised in syntactic categories and do not consider preferences. One attack matches with a case where an argument uses a defeasible rule, that is refuted by a different argument.

Definiton 4.1.17 (Undercutting attack)

Let *A*,*B* be two arguments where *A* undercuts *B* (on *B'*), if $Conc(A) \in \overline{B'}$ for a $B' \in Sub(B)$ being $B''_1, ..., B''_n \Rightarrow \psi$.

Example 4.1.18

In example 4.1.7, A_8 can be undercut in two various ways: by an argument with $\overline{A_5}$, that undercuts A_8 on A_7 , and by an argument with $\overline{A_5}$ as a conclusion, that undercuts A_8 on A_5 . Let us now take $p \Rightarrow t$ with d_1 and s, r, $t \Rightarrow w$ with d_2 , and x extends \mathcal{K}_p , and p, $x \to \neg d_1$ extends \mathcal{R}_s , and q, $x \Rightarrow \neg d_2$ extends \mathcal{R}_d . Therefore A_5 is undercut through

 $A_9: A_{1,x} \rightarrow \neg d_1$

whereas A_7 is undercut through

 $A_{10}: A_{2,x} \Rightarrow \neg d_2$

Attackers that undercut say that there is a specific situation where a defeasible inference rule is not able to be applied, without drawing the opposite conclusion. On the other hand

40

4.1. ARGUMENTATION SYSTEMS WITH STRUCTURED ARGUMENTS

attacks that are rebutting have a contrary/contradictory conclusion towards a defeasible (sub-)conclusion of the attacked argument.

Definition 4.1.19 (Rebutting attack)

Let *A*, *B* be two arguments , where *A* rebuts *B*, if $Conc(A) \in \overline{\phi}$ for a $B' \in Sub(B)$ being $B''_1, ..., B''_n \Rightarrow \phi$. Here *A* contrary-rebuts *B* if Conc(A) is a contrary of ϕ .

Example 4.1.20

Taking example 4.1.7, A_8 can be rebutted regarding A_5 with an argument for \overline{t} and regarding A_7 with an argument for \overline{v} . Now if $\overline{t} = -t$ and the rebutting argument has a defeasible top rule, then A_5 alternately rebuts the argument for \overline{t} . Nonetheless A_8 does not rebut this argument, besides when $w \in \overline{t}$. This displays three reasons, why rebutting is not symmetric: the rebutted argument may have a top rule, rebutting can be contradictory-rebutting and rebutting might be launched on a subargument. On the other hand this displays if the rebutting attack is not contrary-rebutting and the rebutter has a defeasible top rule, the immediately rebutted subargument alternately rebuts its attacker. Let us extend \mathcal{K}_p with x and \mathcal{R}_d with $p, x \Rightarrow \neg t$. With that, the argument

 $A_{11}: A_1, x \Rightarrow \neg t$

rebuts and A_5 rebuts it.

Definition 4.1.21 (Undermining attack)

Let *A*, *B* be two arguments, where *A* undermines *B*, if $Conc(A) \in \overline{\phi}$ for a $\phi \in Prem(B) \mathcal{K}_n$. Here *A* contrary-undermines *B*, if Conc(A) is a contradictory of ϕ or if $\phi \in \mathcal{K}_a$.

Example 4.1.22 (Example 4.1 from [46])

Let $\mathcal{R}_d = \{r_1, r_2\}$ where

 $r_1 = WearsRing \Rightarrow Married$

 $r_2 = PartyAnimal \Rightarrow Bachelor$

Let $\mathcal{R}_s = \{r_3, r_4\}$ where

 $r_3 = Married \Rightarrow \neg Bachelor$

 $r_4 = Bachelor \Rightarrow \neg Married$

and let $\mathcal{K}_{p} = \{WearsRing, PartyAnimal\}$. Examine the following arguments:

A ₁ : WearsRing	<i>B</i> ₁ : <i>PartyAnimal</i>
$A_2: A_1 \Rightarrow Married$	$B_2: B_1 \Rightarrow Bachelor$
$A_3: A_2 \rightarrow \neg Bachelon$	$B_3: B_2 \rightarrow \neg Married$

Now A_3 rebuts B_3 because of its subargument B_2 whereas B_3 rebuts A_3 regarding its subargument A_2 .

Defeat

Knowing now how attacks on arguments work, we can use the ordering function to determine which arguments will be resulting in defeat. There are no preferences for undercutting attacks, since a weaker undercutter and its stronger victim could be in the identical extension. The same applies to the other two possibilities to attack regarding contraries. The remainder of attacks will be implemented by the argument ordering, to see if a certain attack leads to defeat.

Definition 4.1.23 (Successful rebuttal)

Let *A*, *B* be two arguments, where *A* successfully rebuts *B*, if *A* rebuts *B* on *B'* and if either *A* contrary-rebuts *B'* or $A \neq B'$.

Definition 4.1.24

Let us take example 4.1.22: The conflict between A_3 and B_3 is cleared up through the comparison of A_3 and B_2 and of B_3 and A_2 . When applying the last-link ordering, if $r_1 < r_2$ then $B_2 < A_3$ (A_3 successfully rebuts B_2 and B_3 , but A_3 does not do so with A_2 and A_3). On the other hand, if $r_1 \leq r_2$ and $r_2 \leq r_1$ then $A_2 \leq B_3$ and $B_2 \leq A_3$ (both A_3 and B_3 successfully rebut the other).

Defintion 4.1.25 (Successful undermining)

Let *A*, *B* be two arguments, where *A* successfully undermines *B*, if *A* undermines *B* and if either *A* contrary-undermines *B* or $A \neq B$.

This definition uses the definition of an argument's premise also being a subargument. These three defeat relations can now be formed into on global definition of "defeat":

Definiton 4.1.26 (Defeat)

Let *A*, *B* be two arguments, where *A* defeats *B*, if *A* successfully rebuts/undermines or undercuts *B*. *A* strictly defeats *B*, if *A* defeats *B* and *B* does not defeat *A*.

Example 4.1.27

Taking another look at example 4.1.14, A_3 and B_2 rebut one another. Through the lastlink argument ordering, $B_2 <_{s} A_3$ holds, so A_3 successfully rebuts B_2 , but B_2 does not successfully rebut A_3 . Therefore A_3 strictly defeats B_2 . On the other hand, arguing with the weakest-link ordering we have $A_3 <_{s} B_2$ and thus B_2 strictly defeats A_3 .

Join abstract and structured argumentation

After the past section, we can now connect the structured argumentation with Dung's abstract argumentation theory.

Definiton 4.1.28 (AAF coinciding to AT)

Let AAT_{AT} be an *abstract argumentation theory* coinciding with an *argumentation theory* AT, that is a tuple $\langle Args, Defeat \rangle$ where

42

- Args is an argument set based on AT regarding definition 4.1.6,
- Defeat is a relation regarding Args based on definition 4.1.26.

The upcoming definition utilizes the notions of justified, defensible and overruled arguments: We will write *S*-justified, if an argument is justified based on the semantics *S* (same for defensible and overruled).

Definition 4.1.29 (Status of conclusions)

Given a semantic *S*, an argumentation theory *AT* and a formula $\phi \in \mathcal{L}_{AT}$:

- φ is *S*-*justified* regarding *AT*, if there is one *S*-justified argument based on *AT* with the conclusion φ
- (2) φ is *S*-defensible regarding *AT*, if there is one *S*-defensible argument based on *AT* with the conclusion φ and φ is not *S*-justified in *AT*
- (3) φ is *S*-overruled regarding *AT*, if there is one *S*-overruled argument based on *AT* with the conclusion φ and φ is neither *S*-justified nor *S*-defensible

In addition, the following definition apprehends floating conclusions.

Definition 4.1.30 (Justified conclusions)

 ϕ is *S*-*f*-*justified* regarding *AT* if every *S*-extension of *AT* includes an argument with ϕ as its conclusion.

4.2 Domain-specific vs general inference rules

Depending on the inference rules being domain-specific or general, there are two possible uses for that framework. A common method is to state global patterns of reasoning as inference rules. An example for that would be universal instantiation. On the other hand, domain-specific inference rules are common in nonmonotonic logic and default logic. The disparities between these two views will be explained with the following example:

Let us accept that every Bavarian is German, that Germans tend to like beer and that Hans is German. Through the domain-specific inference rule we can display this information in a propositional language as follows:

 $\mathcal{R}_{s} = \{Bavarian \rightarrow German\} \\ \mathcal{R}_{d} = \{German \Rightarrow tends \ to \ like \ Beer\} \\ \mathcal{K}_{p} = \{Bavarian\}$

On the other hand we need to characterize the two general inference rules in an object language \mathcal{L} . The first one can be characterized through the material implication. The second one is a bit more difficult: here we need to add a connection regarding defeasible conditionals to \mathcal{L} plus a defeasible modus-ponens inference rule for that connection.

 $\mathcal{R}_{s} = \{\phi, \phi \supset \psi \rightarrow \psi \text{ (for every } \phi, \psi \in \mathcal{L}), ...\} \\ \mathcal{R}_{d} = \{\phi, \phi \rightsquigarrow \psi \Rightarrow \psi \text{ (for every } \phi, \psi \in \mathcal{L}), ...\}$

 $\mathcal{K}_{p} = \{Bavarian \supset German, Germans \rightsquigarrow tend to like beer, Bavarian\}$

The inference rules defined by Vreeswijk [76], [77] and Pollock [57], [55]were meant to describe general patterns of reasoning. This correlates very well with inference rules in the standard logic. Pollock in fact devoted a lot of his time towards the field of general patterns of defeasible reasoning, and named it prima facie reasons. Prima facie reasons entail certain reasoning patterns concerning perception, induction, temporal persistence, memory and statistical syllogism. These reasons can be utilized in the current framework as defeasible inference schemes.

Let us take the principle of perception and the principle of memory:

$$d_{\mathrm{p}}(x,\phi)$$
: Sees $(x,\phi) \Rightarrow \phi$

$$d_{\rm m}(x,\phi)$$
: Recalls $(x,\phi) \Rightarrow \phi$

Subsequently every undercutter for d_p declares conditions, where perceptions are uncertain, whereas every undercutter for d_m declares conditions, where memories might be faulty. A common reason for false memories of events is a recollection that is fabricated partially or as a whole through seeing or hearing about that certain event. One could build a global undercutter for distorted memories like

 $u_{\mathrm{m}}(x, \phi)$: DistortedMemory $(x, \phi) \Rightarrow \neg d_{\mathrm{m}}(x, \phi)$

and extend it with information in this way

 $\forall x, \phi (\text{ReadsAbout}(x, \phi) \rightsquigarrow \text{DistortedMemory}(x, \phi))$

The Prima facie reasons are considered family of argument schemes. Walton *et al.* [80] defined the notion of an argument scheme for the field of philosophy but it is of great importance for the computational study of argumentation. These schemes are non-deductive patterns of reasoning, that entail a set of premises and a, from this set drawn, conclusion. Let us take the following example from Walton himself [80]:

E is an expert in domain *D E* asserts that *P* is true *P* is within *D P* is true

From this scheme, these six questions arise:

- (1) How credible is *E* as an expert source ?
- (2) Is E an expert in domain D?
- (3) What did *E* assert that implies *P*?
- (4) Is *E* personally reliable as a source ?
- (5) Is *P* consistent with what other experts assert ?
- (6) Is *E*'s assertion of *P* based on evidence ?

44

One can now validate this reasoning with argument schemes by considering to see them as defeasible inference rules and to see these questions as indicators for counterarguments. Here we can see a direct connection between the three different attacks on arguments and three different types of critical questions of argument schemes. Certain question doubt the premise of a certain argument and with that indicate an undermining attack, whereas others indicate an undercutting attack or a rebutting attack. In our example, the second and third question indicate underminers regarding the first and second premise. The first, fourth and sixth question indicate undercutters, through the case where *E* is biased or incredible for various reasons and for his scientific unfounded statements. Lastly the fifth question indicates a rebutting attack of this scheme.

4.3 Caminada's and Amgoud's Rationality postulates

One can say that Dung's semantics is equivalent to rationality confinements on argument assessment in abstract argumentation frameworks. Through extension of his approach with structured arguments, the question arises if this addition leads to more rationality confinements. This question was answered through Caminada's and Amgoud's [4] "rationality postulates" for their so-called "rule-based argumentation". These specific postulates define confinements on every extension of an argumentation framework similar to an argumentation theory:

- Closure under subarguments: given any argument in an extension, all of its subarguments are aswell in that extension
- Closure under strict rules: given a set of conclusions of every argument in an extention, this set is closed under the strict-rule application
- Direct consistency: given a set of every argument in an extension, this set is consistent
- **Indirect consistency:** given a set of conclusions of every argument in an extension, the closure of this set under the strict-rule application is consistent

Prior to the study of how far the current framework fullfills these postulates, various specifications towards the strict inference rules need to be discussed. We will start with the notion of transposition of strict rules and the closure of sets of strict rules regarding transposition.

Definition 4.3.1 (Transposition)

Let there be a strict rule *s*, where *s* is a *transposition* of $\phi_1, ..., \phi_n \rightarrow \psi$, if $s = \phi_1, ..., \phi_{i-1}, -\psi, \phi_{i+1}, ..., \phi_n \rightarrow -\phi_i$ for a given $1 \le i \le n$.

Definiton 4.3.2 (Transposition operator)

Let there be a set of strict rules \mathcal{R}_s . Then $Cl_{tp}(\mathcal{R}_s)$ is the smallest set where:

- $\mathcal{R}_{s} \subseteq Cl_{tp}(\mathcal{R}_{s})$, and
- if $s \in Cl_{tp}(\mathcal{R}_s)$ and let there be a transposition t of s, where $t \in Cl_{tp}(\mathcal{R}_s)$.

With that, we can now define the subclass of argumentation systems that are closed regarding transposition.

Definition 4.3.3 (Closure under transposition)

Let $(\mathcal{L}, \neg, \mathcal{R}, \leqslant)$ be an argumentation system that is *closed under transposition*, if $\mathcal{R}_s = Cl_{tp}(\mathcal{R}_s)$. In general, an argumentation theory is closed regarding transposition, if its argumentation system is.

Furthermore, Caminada's and Amgoud's work deliver a definition of the closure of a set of formulas under the application of strict rules.

Definition 4.3.4 (Closure from a set including formulas)

Let $\mathcal{P} \subseteq \mathcal{L}$. Then the *closure* of \mathcal{P} regarding the set \mathcal{R}_s of strict rules (which is $Cl_{\mathcal{R}_s}(\mathcal{P})$) is the smallest set where:

- $\mathcal{P} \subseteq Cl_{\mathcal{R}_{s}}(\mathcal{P})$
- if $\phi_1, ..., \phi_n \to \psi \in \mathcal{R}_s$ and $\phi_1, ..., \phi_n \in Cl_{\mathcal{R}_s}(\mathcal{P})$ then $\psi \in Cl_{\mathcal{R}_s}(\mathcal{P})$.

If $\mathcal{P} = Cl_{\mathcal{R}_s}(\mathcal{P})$, then \mathcal{P} is called *closed*.

In the following it is important, if \vdash envoked by an argumentation system satisfies the contraposition.

Definition 4.3.5 (Closure regarding contraposition)

Let a certain argumentation system be *closed regarding contraposition* if for every $S \subseteq \mathcal{L}$, every $s \in S$ and every ϕ , holds that when $S \vdash \phi$ thus $S \setminus \{s\} \cup \{-\phi\} \vdash -s$. Let a certain argumentation theory be closed regarding contraposition, if its argumentation system is.

With that being said, the first and second postulate are supported unconditionally for the current framework.

Proposition 4.3.6

Let the tuple $\langle A, defeat \rangle$ be an *AAT* according to definition 4.1.28 and *E* either its grounded, preferred or stable extension. In that case for every $A \in E$: if $A' \in Sub(A)$ then $A' \in E$.

Proposition 4.3.7

Let the tuple $\langle A, defeat \rangle$ be an *AAT* correlating to an argumentation theory, and *E* either its grounded, preferred or stable extension. In this case {Conc(*A*) | *A* \in *E*} = *Cl*_{*R*_s}({Conc(*A*) | *A* \in *E*}.

Globally thinking, one needs to be careful with the validity of the two consistency postulates. One can say though, if the last-link argument ordering is applied, it is then valid for the systems closed regarding transposition or contraposition. Here the strict closure of the \mathcal{K}_n is stable and "well-formed", which means they accept the deliberate application of assumptions and contraries:

46

Defintion 4.3.8

Let an argumentation theory be *well-formed* if:

- (1) there is not a resultant of a defeasible rule, that is a contrary of the resultant of a strict rule;
- (2) $\phi \in \mathcal{K}_a$ and ϕ is a contrary of ψ , then $\psi \notin \mathcal{K}_n \cup \mathcal{K}_p$ and \mathcal{R} has no rule, that has ψ as a conclusion or result.

The second condition essentially shows, that an assumption is only a contrary/contradictory from another assumption. Let

$$\mathcal{R}_{s} = \{ p \rightarrow q \}, \mathcal{R}_{d} = \{ r \Rightarrow s, t \Rightarrow u \}$$
$$\mathcal{K}_{p} = \{ p, r \}, \mathcal{K}_{a} = \{ v \}$$

be a not well-formed *AT*, where *s* is a contrary towards *q* and *v* a contrary towards *u*. With that, the first condition is violated, because of *A*: $p \rightarrow q$ and *B*: $r \Rightarrow s$.

Theorem 4.3.9

Let the tuple $\langle A, defeat \rangle$ be an *AAT* based on a well-formed *AT* that is closed regarding contraposition or transposition. It also has a last-link argument ordering and a consistent $Cl_{\mathcal{R}_s}(\mathcal{K}_n)$. Let also *E* be either its grounded, preferred or stable extension. Subsequently the set {Conc(A) — $A \in E$ } is consistent.

Theorem 4.3.10

Let the tuple $\langle A, defeat \rangle$ be an *AAT* based on a well-formed *AT* that is closed regarding contraposition or transposition. It also has a last-link argument ordering and a consistent $Cl_{\mathcal{R}_s}(\mathcal{K}_n)$. Let also *E* be either its grounded, preferred or stable extension. Subsequently the set $Cl_{\mathcal{R}_s}(\{Conc(A) - A \in E\})$ is consistent.

Corollary 4.3.11

If both conditions of theorem 4.3.10 are fulfilled, then given *E* a grounded, preferred or stable extension, the set { $\phi \mid \phi$ is a premise of a certain argument in *E*} is consistent.

4.4 Serial and parallel self-defeat

As we already mentioned in section 2.2., a good analysis of self-defeating arguments needs to specify the arguments's structure precisely. Having completed that, this thesis follows that up with the explantion why it is necessary. In the current framework we can have either *serial self-defeat*, where an argument defeats one of its previous steps, or *parallel self-defeat*, where minimum two arguments each have a contradictory conclusion and those arguments are used as the premises of \bot . If argumentation systems are not defined correctly, parallel self-defeat can provoke issues and problems.

Example 4.4.1

Let there be an argument scheme based on a witness testimonial with an undercutter and the witness is not trustworthy:

```
d_w(x, \phi): Says(x, \phi) \Rightarrow \phi
u_w(x, \phi): NotTrustworthy(x) \rightarrow \neg d_w(x, \phi)
```

In this case, let \mathcal{K}_p be Says(*Jeff*, "NotTrustworthy(*Jeff*)"). With that we have:

$$A_1$$
: Says(Jeff, "Incredible(Jeff)")
 A_2 : $A_1 \Rightarrow$ Incredible(John)
 A_3 : $A_2 \rightarrow \neg d_w$ (Jeff, "Incredible(Jeff)")

We now have the self-defeating argument A_3 , because it undercuts itself in A_2 . The grounded and preferred semantic have a certain extension $E = \{A_1\}$. This is the best-case outcome, because, let Jeff talk about a separate topic, e.g. "the accused stabbed the victim with a pair of scissors" if A_3 was overruled, the specific argument that could be created for "the accused stabbed the victim with a pair of scissors" will be justified (all of its preceding defeaters are overruled), but it is from a witness that is regarded as not trustworthy.

Let us now look at an example, where parallel self-defeat causes issues.

Example 4.4.2

Let there be $\mathcal{R}_d = \{p \Rightarrow q; r \Rightarrow \neg q; t \Rightarrow s\}$ and $\mathcal{K} = \{p, r, t\}$, where \mathcal{R}_s entails every propositionally valid legitimate inference.

 $A_{1}: p \qquad A_{2}: A_{1} \Rightarrow q$ $B_{1}: r \qquad B_{2}: B_{1} \Rightarrow \neg q$ $C_{1}: A_{2}, B_{2} \rightarrow \bot \qquad C_{2}: C_{1} \rightarrow \neg s$ $D_{1}: t \qquad D_{2}: D_{1} \Rightarrow s$

In this construction there is a problem: *s* can be a random formula, where a given defeasible argument separate to A_2 and B_2 , like D_2 , might be rebutted from C_2 (dependent on the argument ordering). This problem actually only plays a role for the grounded semantics of Dung's semantics. Because the other two semantics have either A_2 or B_2 , C_2 is not included in these extensions and with that D_2 is included in them. If A_2 and B_2 are not strictly defeating the opposing argument, then neither one of them is in the grounded extension, in order that the extension is not defending D_2 regarding C_2 and so does not include D_2 .

Ultimately one can see that the two types of self-defeat should be treated separately: parallel self-defeating arguments should be overruled in every instance, whereas serial self-defeating arguments should maintain the avoidance of other arguments being justified or defensible.

CHAPTER 5

Argumentation Logic: Conclusion and discussion

5.1 Relevance for present and outlook on future

In the end, this thesis gave an indepth view of the logic-based argumentation framework, which is known as Argumentation Logic. With the application of logic-based argumentation on given problems (from default or legal or commonsense reasoning to logic programming, decision and negotiation), a so-called preference or priority amidst the logical sentences (which in the bigger picture are the premise of an argument) is useful and creates argumentation with structured arguments (preference based argumentation). The field of abstract rule-based argumentation with strict and defeasible rules has a rich history: from the early research of Pollock [57], the transitional stages of Simari and Loui [69], Pollock [58], Vreeswijk [77], Prakken and Sartor [64], to the present collective work on ASPIC⁺. In recent times there has been a use for AL in the field of formalizing psychological theories of story comprehension [23] for the incorporation of contrapositive reasoning with default rules. An overall examination of the extension of AL considering an argumentation framework with structured arguments, with the goal of attaining a more global union of defeasible and strict classical reasoning, is a big part of the forthcoming research. The are already system, that visualize the argumentative reasoning of Argumentation Logic [81] with the goal to assist a dialectical process for the resolution of conflicts. With these new applications and implementations, there are also more questions being

With these new applications and implementations, there are also more questions being posed:

A present research movement in formal argumentation is the union of argumentationbased inference and probability theory. This is clear, given that argumentation was considered as a model for reasoning under uncertainty. A question here would be, how representations of the strength or relative preference of arguments are connected with probability theory. There is work on probabilistic argumentation, which assigns probabilities to arguments in *AAFs*, but it is problematic in a way: these assignments are critical, given that probability theory assigns probabilities to the truth of statements or outcomes of events, and arguments are not either one.

As a contrast to abstract argumentation, computational aspects of rule-based argumentation and its ways of instantiation is very primitive. There is available work on algorithms

50 CHAPTER 5. ARGUMENTATION LOGIC: CONCLUSION AND DISCUSSION

and complexity solutions for rule-based argumentation regarding defeasible rules and preferences.

Another underdeveloped field is argument preference relations and their properties. Namely argument orderings applied on decision theory or probability theory.

The new approach of dialectics has helped knowledge representation and formalisms of reasoning for legal applications and delivered good examples. Namely it has led to the discovery of common foundations for logic programming and procedural argumentative frameworks [63][64]. The general topic of argumentation is a natural device for the characterization of supportive reasoning. With that it is a great method for the goal of AI: to model commonsense reasoning. With a lot of foundational work and great results already achieved, the future looks promising.

Bibliography

- L. Amgoud and P. Besnard. A formal analysis of logic-based argumentation systems. Proceedings of the fourth International Conference on Scalable Uncertainty Management, pages 42–55, 2010.
- [2] L. Amgoud and Ph. Besnard. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics*, 23:229–267, 2013.
- [3] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. van Veenen, and G. Vreeswijk. Final review and report on formal argumentation system. *ASPIC IST-FP6-002307*, Deliverable D2.6, 2006.
- [4] L. Amgoud and M. Caminada. On the evaluation of argumentation formalisms. Artificial Intelligence, 171:286–310, 2007.
- [5] G. Antoniou. A tutorial on default logics. ACM Computing Surveys, 31:337–359, 1999.
- [6] K. Atkinson. What should we do ?: Computational representation of presuasive argument in practical reasoning. *PhD Thesis, Department of Computer Science, University of Liverpool*, 2005.
- [7] K. Atkinson and T. J. M. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171:855– 874, 2007.
- [8] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.
- [9] T. J. M. Bench-Capon. Computational models of argument. *Proceedings of Computational Models of Argument (COMMA) 2006*, 144, 2006.
- [10] T. J. M. Bench-Capon, T. Geldard, and P. H. Leng. A method for the computational modelling of dialectical argument with dialogue games. *Artificial Intelligence and Law*, 8:233–254, 2000.
- [11] T. J. M. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150:97–143, 2003.
- [12] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. *Proceedings of the ninth International Conference on Uncertainty in Artificial Intelligence*, pages 411–419, 1993.

- [13] Ph. Besnard and H. Hunter. Constructing argument graphs with deductive arguments: a tutorial. *Argument and Computation*, 5:5–30, 2014.
- [14] F. Bex, T. J. M. Bench-Capon, and K. Atkinson. Did he jump or was he pushed ?: Abductive practical reasoning. *Artificial Intelligence and Law*, 17:79–99, 2009.
- [15] F. Bex, S. Modgil, H. Prakken, and C. Reed. On logical specifications of the Argument Interchange Format. *Journal of Logic and Computation*, 23:951–989, 2013.
- [16] F. J. Bex, P. van Koppen, H. Prakken, and B. Verheij. A hybrid formal theory of arguments, stories and criminal evidence. *Artificial Intelligence and Law*, 18:123–152, 2010.
- [17] L. Birnbaum, M. Flowers, and R. McGuire. Towards an AI model of argumentation. Proceedings of the First AAA/Conference, pages 313–315, 1980.
- [18] D. G. Bobrow. Special issie on non-monotonic logic. Artificial Intelligence, 13, 1980.
- [19] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentationtheoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.
- [20] A. Bondarenko, R. A. Kowalski, and F. Toni. An assumption-based framework for non-monotonic reasoning. *Proceedings of the second International Workshop on Logic Pro*gramming and Nonmonotonic Logic, pages 171–189, 1993.
- [21] M. Caminada. On the issue of reinstatement in argumentation. *Logics in Artificial Intelligence*, 4160:111–123, 2006.
- [22] Martin W. A. Caminada. A discussion game for grounded semantics. Proceedings of the third International Workshop on Theory and Applications of Formal Argumentation, 9524:59–73, 2015.
- [23] I. Diakidou, A. Kakas, L. Michael, and R. Miller. A psychology-inspired approach to automated narrative text comprehension. 14th International Conference on Principles of Knowledge Representation and Reasoning, pages 610–613, 2014.
- [24] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
- [25] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171:642–674, 2007.
- [26] M. Elvang-Göransson, P. Krause, and J. Fox. Acceptability of arguments as 'logical uncertainty'. Proceedings of the second European Conference on Symbolic and Quantative Approaches to Reasoning and Uncertainty, 747 of Lecture Notes in Computer Science:85– 90, 1993.
- [27] T. J. M. Bench-Capon et. al. A History of AI and Law in 50 Papers: 25 Years of the International Conference on AI and Law. *Artificial Intelligence and Law*, 3:215–319, 2012.
- [28] J. Fox and S. Parsons. On using arguments for reasoning about actions and values. Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, 1997.
- [29] D. Gabbay. Editorial comment about "on the difference between aba and aa". *IFCoLog Journal of Logic and its Applications*, 2:1–14, 2015.

- [30] R. Girle, D. L. Hitchcock, P. McBurney, and B. Verheij. Decision support for practical reasoning. a theoretical and computational perspective. *Argumentation machines. New frontiers in argument and computation*, pages 55–83, 2004.
- [31] T. F. Gordon. The pleadings game. Artificial Intelligence and Law, 4:239–292, 1993.
- [32] T. F. Gordon. The Pleadings Game. An artificial intelligence model of procedural justice. *Dordrecht: Kluwer*, 1995.
- [33] T. F. Gordon, H. Prakken, and D. Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.
- [34] T. F. Gordon, H. Prakken, and D. N. Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.
- [35] T. F. Gordon, H. Prakken, and D. N. Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.
- [36] G. Governatori, M. J. Maher, G. Antoniou, and D. Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14:675–702, 2004.
- [37] F. Grasso, A. Cawsey, and R. Jones. Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition. *International Journal of Human-Computer Studies*, 53:1077–1115, 2001.
- [38] H. L. A. Hart. The ascription of responsibility and rights. Proceedings of the Aristotelean Society, 49:171–194, 1949.
- [39] J. Horty, R. H. Thomason, and D. S. Touretzky. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence*, 42:311–348, 1990.
- [40] A. Hunter. A probabilistic approach to modelling uncertain logical arguments. International Journal of Approximate Reasoning, 54:47–81, 2013.
- [41] G. W. Leibniz. Leibniz Selections. New York: Scribner, P. P. Wiener (Ed. Trans.):51, 1951.
- [42] R. P. Loui. Defeat among arguments: a system of defeasible inference. Computational Intelligence, 2:100–106, 1987.
- [43] R. P. Loui. Process and policy: resource-bounded non-demonstrative reasoning. Computational Intelligence, 14:1–38, 1998.
- [44] L. McCarty. Reflections on taxman. an experiment in artificial intelligence and legal reasoning. *Harvard Law Review*, 90:89–116, 1977.
- [45] L. McCarty. An implementation of eisner v. macomber. Proceedings of the fifth international conference on artificial intelligence and law, pages 276–286, 1995.
- [46] S. Modgil and H. Prakken. The ASPIC framework for structured argumentation: a tutorial. *Argument and Computation*, 5:31–62, 2014.
- [47] S. H. Nielsen and S. Parsons. An application of formal argumentation: fusing bayesian networks in multi-agent systems. *Artificial Intelligence*, 171:754–775, 2007a.
- [48] D. Nute. Defeasible reasoning: A philosophical analysis in prolog. Aspects of Artificial Intelligence, pages 251–288, 1988.
- [49] D. Nute. Defeasible logic. Handbook of Logic in Artificial Intelligence and Logic Programming, pages 253–395, 1994.

- [50] Ch. Perelman and L. Olbrechts-Tyteca. The New Rhetoric. A Treatise on Argumentation. University of Notre Name Press, 1969.
- [51] Plato. The Republic. New York :Books, Book I:190, 1943.
- [52] J. L. Pollock. Self-defeating arguments. *Minds and Machines*, 1:367–392, 1991.
- [53] J. L. Pollock. A theory of defeasible reasoning. International Journal of Intelligent Systems, 6:33–54, 1991.
- [54] J. L. Pollock. How to reason defeasibility. *Artificial Intelligence*, 57:1–42, 1992.
- [55] J. L. Pollock. Justification and defeat. Artificial Intelligence, 67:377-407, 1994.
- [56] J.L. Pollock. Knowledge and justification. Princeton University Press, 1974.
- [57] J.L. Pollock. Defeasible reasoning. Cognitive Science, 11:481–518, 1987.
- [58] J.L. Pollock. Cognitive Carpentry. A Blueprint for How to Build a Person. *MIT Press, Cambridge*, 1995.
- [59] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. Synthese, 127:187–219, 2001.
- [60] H. Prakken. Argumentation schemes and generalisations in reasoning about evidence. *Proceedings of the ninth International Conference on Artificial Intelligence and Law,* pages 32–41, 2003.
- [61] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal* of Logic and Computation, 15:1009–1040, 2005.
- [62] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
- [63] H. Prakken and G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.
- [64] H. Prakken and G. Sartor. Argument-based logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
- [65] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6:231–287, 1998.
- [66] H. Prakken and G. Sartor. The three faces of defeasibility in the law. *Ratio Juris*, 17:118–139, 2004.
- [67] H. Prakken and G. Sartor. A logical analysis of burdens of proof. *Legal Evidence and Proof: Statistics, Stories, Logic,* pages 223–253, 2009.
- [68] W. Ross. The right and the good. Oxford University Press, 1930.
- [69] G. R. Simari and R. P. Loui. A mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence*, 53:125–157, 1992.
- [70] D. S. Touretzky. Implicit ordering of defaults in inheritance systems. Proceedings of the fourth National Conference on Artificial Intelligence, pages 322–325, 1984.
- [71] David Touretzky. The mathematics of inheritance systems. Morgan Kaufmann, 1986.

- [72] F. H. van Eemeren, B. Garssen, E. C. W. Krabbe, A. F. Snoeck Henkemans, B. Verheij, and J. H. M. Wagemans. Handbook of argumentation theory. *Dordrecht: Springer*, 2014.
- [73] B. Verheij. Two approaches to dialectical argumentation: admissable sets and argumentation stages. *Proceedings of the Eighth Dutch Conference on Artificial Intelligence* (NAIC-96), pages 357–368, 1996.
- [74] B. Verheij. Jumping to conclusions. a logico-probabilistic foundation for defeasible rule-based arguments. *Logics in artificial intelligence*. 13th European conference, pages 411–423, 2012.
- [75] G. Vreeswijk. Defeasible dialectics: a controversy-oriented approach towards defeasible argumentation. *Journal of Logic and Computation*, 3:317–334, 1993.
- [76] G. Vreeswijk. Studies in defeasible argumentation. Doctoral dissertation Free University Amsterdam, 1993.
- [77] G. Vreeswijk. Abstract argumentation systems. Artificial Intelligence, 90:225–279, 2007.
- [78] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. *Proceedings of the 7th European Workshop on Logics in Artificial Intelligence*, 1919:239–253, 2000.
- [79] D. N. Walton. Metadialogues for resolving burden of proof disputes. Argumentation, 2006b.
- [80] D. N. Walton, C. Reed, and F. Macagno. Argumentation Schemes. Cambridge: Cambridge University Press, 2008.
- [81] D.N. Walton. Visualization tools, argumentation schemes and expert opitnion evidence in law. Law, Probability and Risk Advance Access, pages 1–22, 2007.