INSTITUTE OF LOGIC AND INTELLIGENCE
SOUTHWEST UNIVERSITY
CHONGQING
2010

Argumentation Logics

Author:

HENRY PRAKKEN

UTRECHT UNIVERSITY & UNIVERSITY OF GRONINGEN

Contents

Preface

Logic deals with the formal principles and criteria of validity of patterns of inference. This reader discusses logics for a particular group of patterns of inference, viz. inferences that are not absolutely certain, but that can still be rationally made as long as they cannot be defeated on the basis of information to the contrary. Such patterns can be found in commonsense reasoning, i.e., the inferences humans make in their daily life, but also, for example, in legal and medical reasoning and in public debate. This kind of reasoning lacks one important property of 'standard' or 'deductive' reasoning, viz. the property of *monotonicity* of the consequence notion. When inferences are not absolutely certain, it may happen that conclusions inferrable from a particular body of information, are not inferrable from an extended body of information, since the new information gives rise to defeaters of the inference. Hence logics for such inference patterns are often called *nonmonotonic logics*.

Nonmonotonic notions of logical consequence have been studied in artificial intelligence since 1980, when the *Artificial Intelligence* journal published a special issue on nonmonotonic logic. Several nonmonotonic logics were proposed in this issue, the best-known of which are default logic (Reiter, 1980) and circumscription (McCarthy, 1980). For an introduction to nonmonotonic logic see Antoniou (1997). This reader discusses a third kind of nonmonotonic logic, based on the notion of *argumentation*. The first argumentation logics were proposed by the philosopher John Pollock (Pollock, 1987) and the AI researcher Ronald Loui (Loui, 1987). In 1995 Phan Minh Dung showed in an influential paper that argumentation can be seen as a general framework for nonmonotonic logics (Dung, 1995). Argumentation logics have also been shown to be an important component of multi-agent interaction; see Prakken (2006) for an overview. A recent book with surveys on logical and dialogical aspects of argumentation is Rahwan and Simari (2009).

This reader is based on Chapters 4-7 of Prakken (2010), which is in turn based on Prakken and Vreeswijk (2002) (Chapters 1 and 2 of this reader), on Prakken (2010) (Chapter 4) and on Vreeswijk and Prakken (2000) (Section 3.3). Exercises can be found at the end of the chapters; answers to these exercises are given in a separate text.

Chapter 1

Argumentation logics: introduction

This chapter introduces the idea of argumentation-based inference, illustrates it with an example from legal reasoning and then briefly describes the basic building blocks of argumentation logics.

1.1 Argumentation-based inference: the main idea

Introductory textbooks to logic often portray logically valid inference as 'foolproof' reasoning: an argument is valid if the truth of its premises guarantees the truth of its conclusion. However, we all construct arguments from time to time that are not foolproof in this sense but that merely make their conclusion plausible when their premises are true. For example, if we are told that John and Mary are married and that John lives in Amsterdam, we conclude that Mary will live in Amsterdam as well since we know that usually married people live where their spouses live. Sometimes such arguments are overturned by counterarguments. For example, if we are told that Mary lives in Rome to work at the foreign offices of her company for two years, we have to retract our previous conclusion that she lives in Amsterdam. However, as long as such counterarguments are not available, we are happy to live with the conclusions of our fallible arguments. The question is: are we then reasoning fallaciously or is there still logic in our reasoning?

The answer to this question has been given in three decades of research in Artificial Intelligence on nonmonotonic reasoning, partly inspired by earlier developments in philosophy, e.g. Toulmin (1958); Pollock (1974); Rescher (1977); Walton (1996). At first sight it might be thought that patterns of nonmonotonic reasoning are a matter of applying probability theory. However, many such patterns cannot be analysed in a probabilistic way. In the legal domain this is particularly clear: while reasoning about the facts can (at least in principle) still be regarded as probabilistic, reasoning about normative issues clearly is of a different nature. Moreover, even in matters of evidence reliable numbers are usually not available so that the reasoning has to be qualitative.

Argumentation logics model nonmonotonic reasoning as the construction and comparison of arguments for and against a certain claim. Just as in deductive reasoning, arguments must instantiate inference schemes but only some of these schemes capture foolproof reasoning: in the present account deductive logic turns out to be the special case of arguments that can only be attacked on their premises.

1.2 Motivating example

We shall illustrate the idea of argumentation-based inference with a dispute between two persons, A and B. They disagree on whether it is legally acceptable for a newspaper to publish a certain piece of information concerning a politician's private life. Let us assume that the two parties have reached agreement on the following points.

(1) The piece of information I concerns the health of person P;
(2) P does not agree with publication of I;
(3) Information concerning a person's health is information concerning that person's private life

A now states the legal rule that

(4) Information concerning a person's private life may not be published if that person does not agree with publication.

and A says "So the newspapers may not publish I" (Fig. 1.1, page 9). Although B accepts principle (4) and is therefore now committed to (1-4), B still refuses to accept the conclusion that the newspapers may not publish I. B motivates her refusal by replying that:

(5) P is a cabinet minister
(6) I is about a disease that might affect P's political functioning
(7) Information about things that might affect a cabinet minister's political functioning has public significance

Furthermore, B maintains that there is also a legal rule that

(8) Newspapers may publish any information that has public significance

B concludes by saying that therefore the newspapers may write about P's disease (Fig. 1.2, page 10). A agrees with (5–7) and even accepts (8) as a legal rule, but A does not give up his initial claim. Instead he tries to defend it by arguing that he has the stronger argument: he does so by arguing that in this case

(9) The likelihood that the disease mentioned in I affects P's functioning is small.
(10) If the likelihood that the disease mentioned in I affects P's functioning is small, then rule (4) has priority over rule (8).

Thus it can be derived that the legal rule used in A's first argument has priority over the legal rule used by B (Fig. 1.3, page 10), which makes A's first argument stronger than B's, so that it follows after all that the newspapers should be silent about P's disease.

Let us examine the various stages of this dispute in some detail. Intuitively, it seems obvious that the accepted basis for discussion after A has stated (4) and B has accepted it, viz. (1,2,3,4), warrants the conclusion that the piece of information I may not be published. However, after B's counterargument and A's acceptance of its premises (5-8) things have changed. At this stage the joint basis for discussion is (1-8), which gives rise to two conflicting arguments. Moreover, (1-8) does not yield reasons to prefer one argument over the other: so at this point A's conclusion has ceased to be warranted. But then A's second argument, which states a preference between the two conflicting legal rules, tips the balance in favour of his first argument: so after the basis for discussion has been extended to (1-10), we must again accept A's claim as warranted.

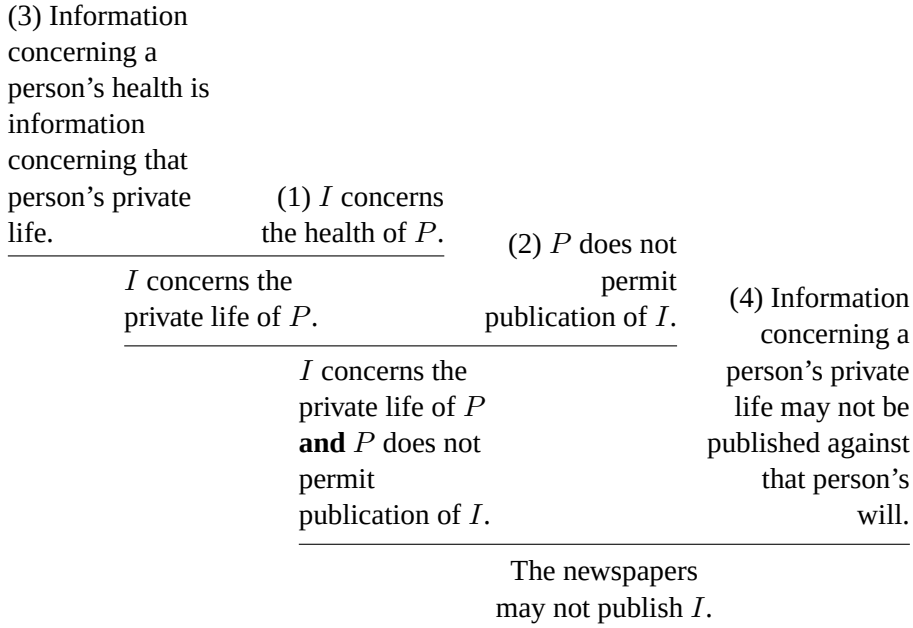


Figure 1.1: A's argument.

Logical systems that formalise this kind of reasoning are called 'argumentation logics', or 'argumentation systems'. As the example shows, these systems lack the monotonicity property of 'standard', deductive logic (say, first-order predicate logic, FOL). According to FOL, if A's claim is implied by (1–4), it is surely also implied by (1–8). From the point of view of FOL it is pointless for B to accept (1–4) and yet state a counterargument; B should also have refused to accept one of the premises, for instance, (4).

Does this mean that our informal account of the example is misleading, that it conceals a subtle change in the interpretation of, say, (4) as the dispute progresses? This is not so easy to answer in general. Although in some cases it might indeed be best to analyse an argument move like B's as a reinterpretation of a premise, in other cases this is different. In actual reasoning, rules are not always neatly labelled with an exhaustive list of possible exceptions; rather, people are often forced to apply 'rules of thumb' or 'default rules', in the absence of evidence to the contrary, and it seems natural to analyse an argument like B's as an attempt to provide such evidence to the contrary. When the example is thus analysed, the force of the conclusions drawn in it can only be captured by a consequence notion that is nonmonotonic: although A's claim is warranted on the basis of (1–4), it is not warranted on the basis of (1–8).

Argumentation logics are the most direct attempt to formalise examples like the above one, by defining notions like argument, counterargument, attack and defeat, and by defining nonmonotonic consequence in terms of the interaction of arguments for and against certain conclusions. This approach was initiated by the philosopher John Pollock (Pollock, 1987), based on his earlier work in epistemology, e.g. (Pollock, 1974), and the AI researcher Ronald Loui (Loui, 1987).

One application of argumentation logics is to to formalise 'quick-and-dirty' commonsense reasoning with empirical generalisations. In everyday life people often rea-

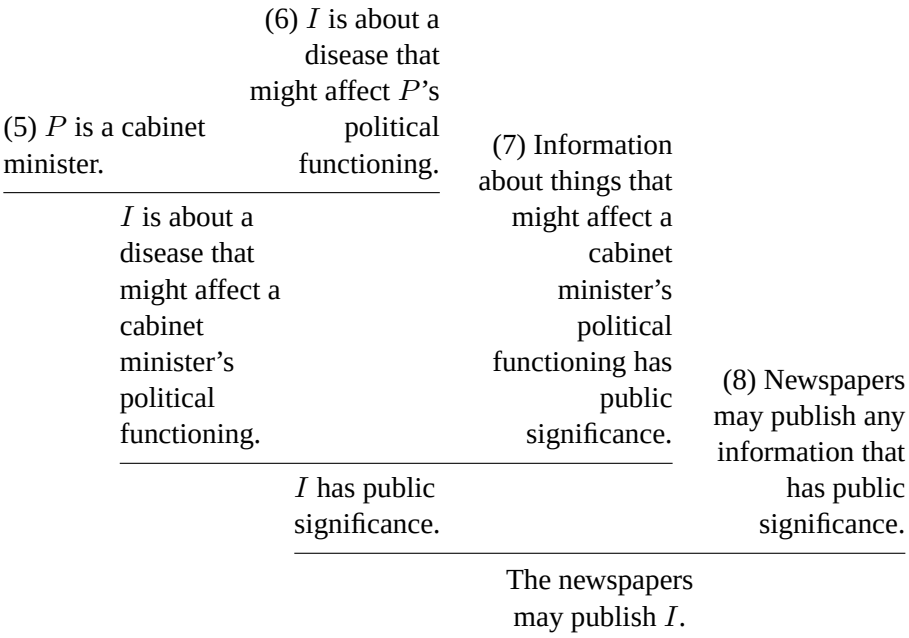


Figure 1.2: B's argument.

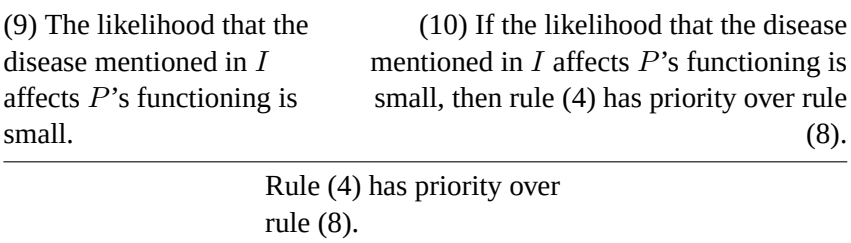


Figure 1.3: A's priority argument.

son with generalisations such as 'Birds fly', 'Italians usually like coffee', 'Chinese usually do not like coffee', 'Witnesses usually speak the truth' or 'When the streets are wet, it must have rained'. In commonsense reasoning, people apply such a generalisation if nothing is known about exceptions, but they are prepared to retract a conclusion if further knowledge tells us that there is an exception (for instance, a given bird is in fact a penguin, a witness has a reason to lie or the streets are wet because they are being cleaned).

However, argumentation systems have wider scope than just reasoning with such empirical generalisations. Firstly, argumentation systems can be applied to any form of reasoning with contradictory information, whether the contradictions have to do with generalisations and exceptions or not. For instance, the contradictions may arise from reasoning with several sources of information, or they may be caused by disagreement about beliefs or about moral, ethical or political claims. Moreover, it is important that several argumentation systems allow the construction and attack of arguments that are traditionally called 'ampliative', such as inductive, analogical and abductive arguments; these reasoning forms fall outside the scope of most other nonmonotonic logics.

One domain in which argumentation systems have become popular is legal reason-

ing. This is not surprising, since legal rules often have exceptions or are in conflict with each other (see the above example). Also, legal reasoning often takes place in an adversarial context, where notions like argument, counterargument, rebuttal and defeat are very common. Argumentation systems have also been applied in, for instance, the medical domain and in multi-agent models of negotiation and collaboration.

1.3 Argumentation logics: a conceptual sketch

In this section we give a conceptual sketch of the general ideas behind argumentation logics. First we sketch the general idea, and then we discuss the five main elements of such logics.

1.3.1 The general idea

Argumentation systems formalise nonmonotonic reasoning as the construction and comparison of arguments for and against certain conclusions. The idea is that the construction of arguments on the basis of a theory is monotonic, i.e., an argument stays an argument if the theory is enlarged with new information. Nonmonotonicity is explained in terms of the interactions between conflicting arguments: it arises from the fact that the new information may give rise to stronger counterarguments, which defeat the original argument. For instance, in case of Tweety the penguin we may construct one argument that Tweety flies because it is a bird, and another argument that Tweety does not fly because it is a penguin, and then we may prefer the latter argument because it is about a specific class of birds, and is therefore an exception to the general rule.

1.3.2 Five elements of argumentation systems

Argumentation systems contain the following five elements (although sometimes implicitly): an underlying logical language, definitions of an argument, of conflicts between arguments and of defeat between arguments and, finally, a definition of the dialectical status of arguments, which can be used to define a nonmonotonic notion of logical consequence.

A logical language

Argumentation systems are built around an underlying logical language and an associated notion of logical consequence, defining the notion of an argument. As noted above, the idea is that this consequence notion is monotonic: new information cannot invalidate arguments as arguments but can only give rise to new counterarguments. Some argumentation systems assume a particular logic, while other systems leave the underlying logic partly or wholly unspecified; thus these systems can be instantiated with various alternative logics, which makes them frameworks rather than systems.

Arguments

The notion of an argument corresponds to a tentative proof (or the existence of such a proof) in the underlying logic. As for the layout of arguments, in the literature on argumentation systems three basic formats can be distinguished, all familiar from the logic literature. Sometimes arguments are defined as a tree of inferences grounded in

the premises, and sometimes as a sequence of such inferences, i.e., as a deduction. Finally, some systems simply define an argument as a premises - conclusion pair, leaving implicit that the underlying logic validates a proof of the conclusion from the premises.

The notions of an underlying logic and an argument still fit with the standard picture of what a logical system is. The remaining three elements are what makes an argumentation system a framework for nonmonotonic reasoning.

Conflicts between arguments

The first is the notion of a *conflict* between arguments (also used are the terms 'attack' and 'counterargument'). In the literature, three types of conflicts are discussed. Firstly, arguments can be attacked on one of their premises, with an argument whose conclusion negates that premise. For example, an argument 'Tweety flies, because it is a bird' can be attacked by arguing that Tweety is not a bird. This kind of attack will in Chapter 4 be called *undermining* attack. The second type of attack is to negate the conclusion of an argument, as in 'Tweety flies, because it is a bird' and 'Tweety does not fly because it is a penguin' (cf. the left part of Fig. 1.4). Finally, when an argument uses a non-

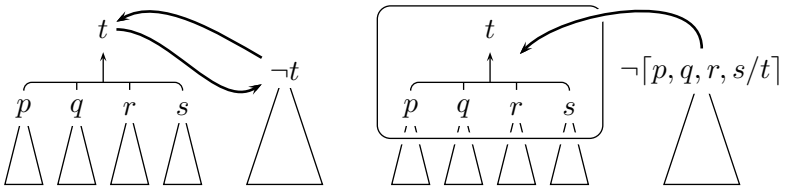


Figure 1.4: Rebutting attack (left) vs. undercutting attack (right).

deductive, or *defeasible* inference rule, it can be attacked on its inference by arguing that there is a special case to which the inference rule does not apply (cf. the right part of Fig. 1.4). After Pollock (1974, 1987), this is usually called *undercutting* attack. Unlike a rebutting attack, an undercutting attack does not negate the conclusion of its target but just says that its conclusion is not supported by its premises and can therefore not be drawn. In order to formalise this type of conflict, the rule of inference that is to be undercut (in Fig. 1.4: the rule that is enclosed in the dotted box, in flat text written as $p, q, r, s/t$) must be expressed in the object language: $\lceil p, q, r, s/t \rceil$) and denied: $\neg\lceil p, q, r, s/t \rceil$. [1] While all arguments can be attacked on their premises, only defeasible arguments can be attacked on their conclusion or inference. The reason why deductive arguments cannot be rebutted or undercut is that deductive inferences are by definition truth-preserving, i.e., the truth of their premises guarantees the truth of their conclusion, so the only way to disagree with the conclusion of a deductive argument is to deny one of its premsies. By contrast, the conclusion of a defeasible argument can be rejected even if all its premises are accepted. In Chapter 4 the difference between deductive and defeasible inference rules will be formalised and several examples of defeasible rules will be discussed. For now, consider the following example of a defeasible argument applying the principle of induction: the argument 'Raven $_{101}$ is black since the observed ravens raven$_1$ ...raven$_{100}$ were black' is undercut by an argument 'I saw raven$_{102}$, which was white'.

[1] Ceiling brackets around a meta-level formula denote a conversion of that formula to the object language, provided that the object language is expressive enough to enable such a conversion.

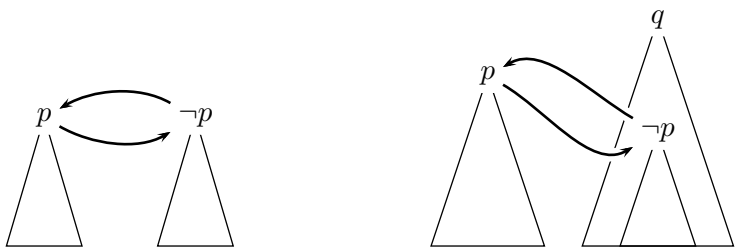


Figure 1.5: Direct attack (left) vs. indirect attack (right).

Note, finally, that all three kinds of attack have a direct and an indirect version; indirect attack is directed against a subconclusion or a substep of an argument, as illustrated by Figure 1.5 for indirect rebutting.

Defeat between arguments

The notion of conflicting, or attacking arguments does not embody any form of evaluation; evaluating conflicting pairs of arguments, or in other words, determining whether an attack is successful, is another element of argumentation systems. It has the form of a binary relation between arguments, standing for 'attacking and not weaker' (in a weak form) or 'attacking and stronger' (in a strong form). The terminology varies: some terms that have been used are 'defeat', 'attack' and 'interference'. Other systems do not explicitly name this notion but leave it implicit in the definitions. In this text we shall use 'defeat' for the weak notion and 'strict defeat' for the strong, asymmetric notion. Note that the several forms of attack, rebutting vs. assumption vs. undercutting and direct vs. indirect, have their counterparts for defeat.

Argumentation systems vary in their grounds for determining the defeat relations. Often only domain-specific criteria are available, which, moreover, are often defeasible. For this reason argumentation systems have been developed that allow for defeasible arguments on these criteria. To give some examples of domain-specific criteria, in domains where observations are important, defeat may depend on the reliability of tests, observers or sensors. In advice giving or consultancy, defeat may be determined by the level of expertise of the advisors or consultants. And in legal applications, defeat may depend on the legal hierarchy among statutes, on the court's level of authority, or on social or moral values. Our example in the introduction contains an argument on the criteria for defeat, viz. A's use of a priority rule (10) based on the expected consequences of certain events. This argument might, for instance, be attacked by an argument that in case of important officials even a small likelihood that the disease affects the official's functioning justifies publication, or by an argument that the negative consequences of publication for the official are small.

The dialectical status of arguments

The notion of defeat is a binary relation on the set of arguments. It is important to note that this relation does not yet tell us with what arguments a dispute can be won; it only tells us something about the relative strength of two individual conflicting arguments. The ultimate status of an argument depends on the interaction between all available arguments: it may very well be that argument B defeats argument A, but

that B is itself defeated by a third argument C; in that case C 'reinstates' A (see Figure 1.6)[2]. Suppose, for instance, that the argument A that Tweety flies because it is a

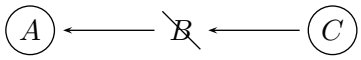


Figure 1.6: Argument C reinstates argument A.

bird is regarded as being defeated by the argument B that Tweety does not fly because it is a penguin (for instance, because conflicting arguments are compared with respect to specificity). And suppose that B is in turn defeated by an argument C, attacking B's intermediate conclusion that Tweety is a penguin. C might, for instance, say that the penguin observation was done with faulty instruments. In that case C reinstates argument A.

Therefore, what is also needed is a definition of the dialectical status of arguments on the basis of all the ways in which they interact. Besides reinstatement, this definition must also capture the principle that an argument cannot be justified unless all its subarguments are justified. There is a close relation between these two notions, since reinstatement often proceeds by indirect attack, i.e., attacking a subargument of the attacking argument (as illustrated by Figure 1.5). It is this definition of the status of arguments that produces the output of an argumentation system: it typically divides arguments in at least two classes: arguments with which a dispute can be 'won' and arguments with which a dispute should be 'lost'. Sometimes a third, intermediate category is also distinguished, of arguments that leave the dispute undecided. The terminology varies here also: terms that have been used are justified vs. defensible vs. defeated (or overruled), defeated vs. undefeated, in force vs. not in force, preferred vs. not preferred, etcetera. Unless indicated otherwise, we shall use the terms 'justified', 'defensible' and 'overruled' arguments.

These notions can be defined both in a 'declarative' and in a 'procedural' form. The declarative form, usually with fixed-point definitions, just declares certain sets of arguments as acceptable, (given a set of statements and evaluation criteria) without defining a procedure for testing whether an argument is a member of this set; the procedural form amounts to defining just such a procedure. Thus the declarative form of an argumentation system can be regarded as its (argumentation-theoretic) semantics, and the procedural form as its proof theory. Note that it is very well possible that, while an argumentation system has an argumentation-theoretic semantics, at the same time its underlying logic for constructing arguments has a model-theoretic semantics in the usual sense, for instance, the semantics of standard first-order logic, or a possible-worlds semantics of some modal logic.

EXERCISE 1.3.1 Reinstatement.

1. Extend Figure 1.6 (p. 14) with an argument D, such that D defeats C. Are there arguments that are justified? If so, which arguments? Are there arguments that are reinstated by D? If so, which?

2. Extend the figure just drawn with a fifth argument, E, such that E defeats D. Are there arguments that are justified? If so, which arguments? Are there arguments

[2]While in figures 1.4 and 1.5 the arrows stood for attack relations, from now on they will depict defeat relations.

that are reinstated by D? If so, which? Are there arguments that are reinstated by E? If so, which?

The content of the remaining chapters is as follows. Chapter 2 presents a fully abstract formal framework for the semantics of argumentation systems, which leaves the structure of arguments and the nature of the defeat relation unspecified. Chapter 3 discusses the proof-theory of these abstract argumentation systems in the form of so-called argument games. Chapter 4 then presents an instantiation of the abstract framework with structured arguments and two kinds of inference rules, deductive and defeasible ones. This framework is still partly abstract in that it abstracts from the nature and origin of these rules and from the nature of the logical language.

Chapter 2

A framework for abstract argumentation

This chapter presents a fully abstract framework for the semantics of argumentation, which leaves the internal structure of arguments and the nature of the defeat relation completely unspecified. As input it assumes nothing else but a set (of arguments) ordered by a binary relation (of defeat) and then defines several 'semantics', that is, properties that subsets of the set of all arguments should satisfy to be justified or defensible. Note that such argumentation semantics are, unlike the semantics of, say, standard first-order logic, not based on the notion of truth: since argumentation systems formalise reasoning that is defeasible, they are not concerned with truth of propositions, but with justification of accepting a proposition as true. In particular, one is justified in accepting a proposition as true if there is an argument for the proposition that one is justified in accepting. Argument-based semantics specify the conditions for when this is the case.

The abstract framework was introduced by Dung (1995). Historically, it came after the development of a number of more concrete argumentation systems, such as the systems of Pollock (1987)−(1994) and Vreeswijk (1993) (both to be discussed in Chapter 4). Dung's framework was a breakthrough in several ways. Firstly, it contains a general account of argumentation semantics, applicable to all systems that instantiate his framework. Secondly, it made a precise comparison possible between different systems by translating them into his abstract format. Third, it made a general study of formal properties of systems possible, which are inherited by all systems that instantiate his framework. Finally, all this applies not just to argumentation systems but also to other nonmonotonic logics, since Dung (1995) showed for several such logics how they can be translated into his abstract framework.

2.1 The status of arguments: preliminary remarks

We now start the discussion of abstract argument-based semantics. As explained above, the task of argument-based semantics is to specify the conditions under which it is justified to accept an argument. These conditions assume an 'input' set of arguments, ordered by a binary relation of 'defeat'.[1] The framework is as abstract as possible, leaving both the structure of arguments and the grounds for defeat unspecified.

[1] Dung (1995) uses the term 'attack', but to maintain uniformity throughout this text, we shall use 'defeat'.

We shall call the input of the framework an 'argumentation theory'[2].

Definition 2.1.1 [Abstract argumentation theories.]

1. An *abstract argumentation theory* (AAT) is a pair $\langle Args, defeat \rangle$, where $Args$ is a set of arguments, and *defeat* a binary relation on $Args$.

2. We say that a set S of arguments defeats an argument A iff some argument in S defeats A; and S defeats a set S' of arguments iff it defeats a member of S'.

As for applications of the framework, one might think of the set $Args$ as all arguments that can be constructed in a given logic from a given set of premises (although this is not always the case; see the discussion below in Chapter 4 of 'partial computation'). Unless stated otherwise, we shall below implicitly assume an arbitrary but fixed argumentation theory. Recall that we read 'A defeats B' in the weak sense of 'A conflicts with B and is not weaker than B'; so in some cases it may happen that A defeats B and B defeats A. If A defeats B, then if B does not defeat A we say that A *strictly defeats* B, otherwise A *weakly defeats* B.

Let us now concentrate on the task of defining the notion of a justified argument. Which properties should such a definition have? A simple definition is the following.

Definition 2.1.2 Arguments are either justified or not justified.

1. An argument is *justified* iff all arguments defeating it (if any) are not justified.

2. An argument is *not justified* iff it is defeated by an argument that is justified.

This definition works well in simple cases, in which it is clear which arguments should emerge victorious, as in the following example.

Example 2.1.3 Consider three arguments A, B and C such that B defeats A and C defeats B:

$$A \longleftarrow B \longleftarrow C$$

A concrete version of this example is

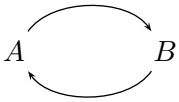
$A =$ 'Tweety flies because it is a bird'
$B =$ 'Tweety does not fly because it is a penguin'
$C =$ 'The observation that Tweety is a penguin is unreliable'

C is justified since it is not defeated by any other argument. This makes B not justified, since B is defeated by C. This in turn makes A justified: although A is defeated by B, A is reinstated by C, since C makes B not justified.

In other cases, however, Definition 2.1.2 is circular or ambiguous. In particular when arguments of equal strength interfere with each other, it is unclear which argument should remain undefeated.

Example 2.1.4 (Even cycle.) Consider the arguments A and B such that A defeats B and B defeats A.

[2]Dung says 'argumentation framework'.



A concrete example is

$A =$ 'Nixon was a pacifist because he was a quaker'
$B =$ 'Nixon was not a pacifist because he was a republican'

Can we regard A as justified? Yes, we can, if B is not justified. Can we regard B as not justified? Yes, we can, if A is justified. So, if we regard A as justified and B as not justified, Definition 2.1.2 is satisfied. However, it is obvious that by a symmetrical line of reasoning we can also regard B as justified and A as not justified. So there are two possible 'status assignments' to A and B that satisfy Definition 2.1.2: one in which A is justified at the expense of B, and one in which B is justified at the expense of A. Yet intuitively, we are not justified in accepting either of them.

In the literature, two approaches to the solution of this problem can be found. The first approach consists of changing Definition 2.1.2 in such a way that there is always precisely one possible way to assign a status to arguments, and which is such that with 'undecided conflicts' as in our example both of the conflicting arguments receive the status 'not justified'. The second approach instead regards the existence of multiple status assignments not as a problem but as a feature: it allows for multiple assignments and defines an argument as 'genuinely' justified if and only if it receives this status in all possible assignments. The following two sections discuss the details of both approaches.

First, however, another problem with Definition 2.1.2 must be explained, having to do with self-defeating arguments.

Example 2.1.5 (Self-defeat.) Consider an argument L, such that L defeats L (Figure 2.1). Suppose L is not justified. Then all arguments defeating L are not justified, so by clause 1 of Definition 2.1.2 L is justified. Contradiction. Suppose now L is justified. Then L is defeated by a justified argument, so by clause 2 of Definition 2.1.2 L is not justified. Contradiction.

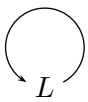


Figure 2.1: A self-defeating argument.

Thus, Definition 2.1.2 implies that there are no self-defeating arguments. Yet in ordinary discourse examples of self-defeating arguments can be found, as in the following example.

Example 2.1.6 (The Liar.) An elementary self-defeating argument can be fabricated on the basis of the so-called *paradox of the Liar*. There are many versions of this paradox. The one we use here, runs as follows:

> Dutch people can be divided into two classes: people who always tell the truth, and people who always lie. Hendrik is Dutch monk, and from Dutch

> monks we know that they tend to be consistent truth-tellers. Therefore, it is reasonable to assume that Hendrik is a consistent truth-teller. However, Hendrik *says* he is a liar. Is Hendrik a truth-teller or a liar?

The Liar-paradox is a paradox, because either answer leads to a contradiction.

1. Suppose that Hendrik tells the truth. Then what Hendrik says must be true. So, Hendrik is a lier. Contradiction.
2. Suppose that Hendrik lies. Then what Hendrik says must be false. So, Hendrik is not a lier. Because Dutch people are either consistent truth-tellers or consistent liers, it follows that Hendrik always tells the truth. Contradiction.

From this paradox, a self-defeating argument L can be made out of (1):

$$\dfrac{\dfrac{\dfrac{\text{Dutch monks tend to be consistent truth-tellers} \qquad \text{Hendrik is a Dutch monk}}{\text{Hendrik is a consistent truth-teller}} \qquad \text{Hendrik says: ``I lie''}}{\text{Hendrik lies}}}{\text{Hendrik is } \textbf{not} \text{ a consistent truth-teller}}$$

If the argument for "Hendrik is *not* a consistent truth-teller" is as strong as its subargument for "Hendrik is a consistent truth-teller," then L defeats one of its own subarguments, and thus is a self-defeating argument.

In conclusion, it seems that Definition 2.1.2 needs another revision, to leave room for the existence of self-defeating arguments. Below we shall discuss for each particular semantics how it deals with self-defeat.

2.2 The unique-status-assignment approach

We now discuss an approach that changes Definition 2.1.2 in such a way that there is always precisely one possible way to assign a status to arguments. This 'unique-status-assignment' approach can best be explained by the way it formalises 'reinstatement' (see above, Section 1.3). It does so by combining a notion of *acceptability* with a fixed-point operator. Recall that an argument that is defeated by another argument can only be justified if it is reinstated by a third argument, viz. by a justified argument that defeats its defeater. Part of this idea is captured by the notion of *acceptability* (which, by the way, is also relevant for the multiple-status-assignments approach, as we shall see below in Section 2.3).

Definition 2.2.1 [Acceptability.] An argument A is *acceptable* with respect to a set S of arguments iff each argument defeating A is defeated by S. When A is acceptable with respect ot S, we also say that S *defends* A.

The arguments in S can be seen as the arguments capable of reinstating A in case A is defeated. To illustrate acceptability, consider again Example 2.1.3: A is acceptable with respect to $\{C\}$, $\{A, C\}$, $\{B, C\}$ and $\{A, B, C\}$, but not with respect to $\varnothing$ and $\{B\}$.

The notion of acceptability is not yet sufficient. Consider in Example 2.1.4 the set $S = \{A\}$. It is easy to see that A is acceptable with respect to S, since all arguments defeating A (viz. B) are defeated by an argument in S, viz. A itself. Clearly, we do not want that an argument can reinstate itself, and this is the reason why, to obtain a unique status assignment, a fixed-point operator must be used.

Intermezzo: fixed point operators Below we need some basics on fixed-point operators. Let S be a set and $O : Pow(S) \longrightarrow Pow(S)$ be an operator which for any subset of S returns a subset of S. $T \subseteq S$ is a *fixed point* of O iff $O(T) = T$. It is known that if O satisfies certain properties, it has a *least fixed point*, i.e. a fixed point which is a subset of all other fixed points of O. The most important of these properties is monotonicity, which is that $O(T) \subseteq O(T')$ whenever $T \subseteq T'$.

Consider now the following operator, which for each set of arguments returns the set of all arguments that are acceptable to it.

Definition 2.2.2 [Grounded semantics.] Let AAT be an abstract argumentation theory, and let $S \subseteq Args_{AAT}$. Then the operator F^{AAT} is defined as follows:

- $F^{AAT}(S) = \{A \in Args_{AAT} \mid A \text{ is acceptable with respect to } S\}$

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

It can be shown that the operator F has a least fixed point, so that the notion of a grounded extension is well-defined[3]. (The basic idea is that if an argument is acceptable with respect to S, it is also acceptable with respect to any superset of S, so that F is monotonic.) Self-reinstatement can then be avoided by defining the set of justified arguments as that least fixed point. Note that in Example 2.1.4 the sets $\{A\}$ and $\{B\}$ are fixed points of F but not its least fixed point, which is the empty set. In general we have that if no argument is undefeated, then $F(\varnothing) = \varnothing$.

These observations allow the following definition of a justified argument.

Definition 2.2.3 [Justified arguments in grounded semantics.] An argument is *justified* with respect to grounded semantics iff it is a member of the grounded extension.

In applying these definitions, it is useful to know that the least fixed point of F can be approximated, and under certain conditions even obtained, by iterative application of F to the empty set.

Proposition 2.2.4 Dung (1995) Consider the following sequence of arguments.

- $F^0 = \varnothing$
- $F^{i+1} = \{A \in Args \mid A \text{ is acceptable with respect to } F^i\}$.

Let $F^\omega = \cup_{i=0}^{\infty}(F^i)$. The following observations hold.

[3]Below the superscript of F will usually be omitted.

1. All arguments in F^ω are justified.

2. If each argument is defeated by at most a finite number of arguments, then an argument is justified iff it is in F^ω.

Proof: (1) follows from the facts that F^ω is included in the least fixed point of F and that if an argument is acceptable with respect to S, it is also acceptable with respect to any superset of S. For (2), assume that each argument has at most a finite number of defeaters. Let $S_0 \subseteq \ldots \subseteq S_n \subseteq \ldots$ be an increasing sequence of sets of arguments, and let $S = S_0 \cup \ldots S_n \cup \ldots$. Let $A \in F(S)$. Since there are only finitely many arguments which defeat A, there exists a number m such that $A \in F^m(S)$. Therefore, $F(S) = F(S_0) \cup \ldots F(S_n) \cup \ldots \Box$

Note that if the condition of (2) does not hold, it is possible that $F^\omega \subset F(F^\omega)$.

In the iterative construction of the set of justified arguments first all arguments that are not defeated by any argument are added, and at each further application of F all arguments that are reinstated by arguments that are already in the set are added. This is achieved through the notion of acceptability. To see this, suppose we apply F for the ith time: then for any argument A, if all arguments that defeat A are themselves defeated by an argument in F^{i-1}, then A is in F^i.

It is instructive to see how this works in Example 2.1.3. We have that

$F^1 = F(\varnothing) = \{C\}$
$F^2 = F(F^1) = \{A, C\}$
$F^3 = F(F^2) = F^2$

The following example, with an infinite chain of defeat relations, provides another illustration.

Example 2.2.5 Consider an infinite chain of arguments $A_1, \ldots, A_n, \ldots$ such that A_1 is defeated by A_2, A_2 is defeated by A_3, and so on.

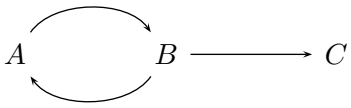
$$A_1 \longleftarrow A_2 \longleftarrow A_3 \longleftarrow A_4 \longleftarrow A_5 \longleftarrow \cdots$$

The least fixed point of this chain is empty, since no argument is undefeated. Consequently, $F(\varnothing) = \varnothing$. Note that this example has two other fixed points, which also satisfy Definition 2.1.2, viz. the set of all A_i where i is odd, and the set of all A_i where i is even.

Defensible arguments

Definition 2.2.3 allows a distinction between two types of arguments that are not justified. Consider first again Example 2.1.3 and observe that, although B defeats A, A is still justified since it is reinstated by C. Consider next the following extension of Example 2.1.4.

Example 2.2.6 (Zombie arguments.) Consider three arguments A, B and C such that A defeats B, B defeats A, and B defeats C.



A concrete example is

$A =$ 'Dixon is no pacifist because he is a republican'
$B =$ 'Dixon is a pacifist because he is a quaker, and he has no gun because he is a pacifist'
$C =$ 'Dixon has a gun because he lives in Chicago'

According to Definition 2.2.3, neither of the three arguments are justified. For A and B this is since their relation is the same as in Example 2.1.4, and for C this is since it is defeated by B. Here a crucial distinction between the two examples becomes apparent: unlike in Example 2.1.3, B is, although not justified, not defeated by any justified argument and therefore B retains the potential to prevent C from becoming justified: there is no justified argument that reinstates C by defeating B. Sometimes arguments like B are called 'zombie arguments': B is not 'alive', (i.e., not justified) but it is not fully dead either; it has an intermediate status, in which it can still influence the status of other arguments.

We shall call the intermediate status of zombie arguments 'defensible'. In the unique-status-assignment approach it can be defined as follows.

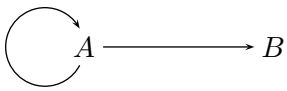
Definition 2.2.7 [Overruled and defensible arguments in grounded semantics.] With respect to grounded semantics, an argument is:

- *overruled* iff it is not justified, and defeated by a justified argument;
- *defensible* iff it is not justified and not overruled.

Self-defeating arguments

How does Definition 2.2.2 deal with self-defeating arguments? Consider the following extension of Example 2.1.5.

Example 2.2.8 Consider two arguments A and B such that A defeats A and A defeats B.

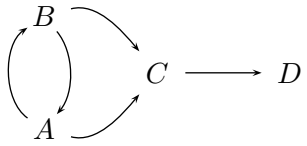


We have that $F(\varnothing) = \varnothing$, so neither A nor B are justified. Moreover, they are both defensible, since they are not defeated by any justified argument. At first sight, it might be thought that this is undesired since it would seem that self-defeating arguments should always be overruled. However, in Chapter 4 we will see that that things are more subtle and that a proper analysis of self-defeating arguments can only be given if the internal structure of arguments is made explicit.

Unique status assignments: problems

We have seen that the unique-assignment approach can be formalised in a mathematically elegant way, and that it produces intuitive results in many cases. However, there are also problems, in particular with examples of the following kind.

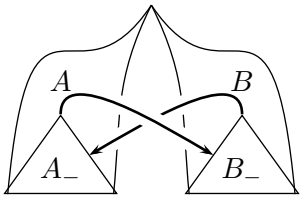
Example 2.2.9 (Floating arguments.) Consider the arguments A, B, C and D such that A defeats B, B defeats A, A defeats C, B defeats C and C defeats D.



Since no argument is undefeated, Definition 2.2.3 tells us that all of them are defensible. However, it might be argued that for C and D this should be otherwise: since C is defeated by both A and B, C should be overruled. The reason is that as far as the status of C is concerned, there is no need to resolve the conflict between A and B: the status of C 'floats' on that of A and B. And if C should be overruled, then D should be justified, since C is its only defeater.

A variant of this example is the following piece of default reasoning. To analyse this example, we must make two assumptions on the structure of arguments, viz. that they have a conclusion and that they have subarguments.

Example 2.2.10 (Floating conclusions.) Consider the arguments A^-, A, B^- and B such that A^- and B^- defeat each other and A and B have the same conclusion.



An intuitive reading is

A^-	=	Brigt Rykkje is Dutch because he was born in Holland
B^-	=	Brigt Rykkje is Norwegian because he has a Norwegian name
A	=	Brigt Rykkje likes ice skating because he is Dutch
B	=	Brigt Rykkje likes ice skating because he is Norwegian

The point is that whichever way the conflict between A^- and B^- is decided, we always end up with an argument for the conclusion that Brigt Rykkje likes ice skating, so it seems that it is justified to accept this conclusion as true, even though it is not supported by a justified argument. In other words, the status of this conclusion floats on the status of the arguments A^- and B^-.

While the unique-assignment approach is inherently unable to capture floating arguments and conclusions, there is a way to capture them, viz. by working with multiple status assignments. To this approach we now turn.

2.3 The multiple-status-assignments approach

A second way to deal with competing arguments of equal strength is to let them induce two alternative status assignments, in both of which one is justified at the expense of the other. In this approach, an argument is 'genuinely' justified iff it receives this status in all status assignments. This approach can be formalised in various ways, of which so-called stable and preferred semantics are the two best-known.

2.3.1 Stable semantics

The first way to allow for multiple status assignments, called stable semantics, is to take Definition 2.1.2 as the basis, and simply use the fact that it allows for multiple assignments. To this end, we turn this definition into one of a 'stable status assignment'.

Definition 2.3.1 [stable status assignments.] A *stable status assignment* on the basis of an abstract argumentation theory $\langle Args, defeat \rangle$ is an assignment to each argument in *Args* of either the status 'in' or the status 'out' (but not both) such that:

1. An argument is *in* iff all arguments defeating it (if any) are out.
2. An argument is *out* iff it is defeated by an argument that is in.

Note that the conditions 1 and 2 are just the conditions of Definition 2.1.2.

Definition 2.3.1 is said to define *stable* status assignments for the following reasons. Firstly, with each stable status assignment a so-called *stable argument extension* can be associated, containing all the arguments that are 'in' in the status assignment.

Definition 2.3.2 [Stable argument extensions.] A set of arguments is an *stable argument extension* iff for some stable status assignment it is the set of all arguments that are assigned the status 'in'.

Now stable argument extensions are what Dung (1995) calls *stable extensions*. In fact, Dung gives another but equivalent definition.

Definition 2.3.3 [Stable extensions.] A conflict-free set S is a *stable extension* iff every argument that is not in S, is defeated by S.

Proposition 2.3.4 The stable argument extensions induced by Definition 2.3.1 are precisely the stable extensions defined by Definition 2.3.3.

Proof: $\Rightarrow$:

Suppose (In, Out) is a stable status assignment. To be proven:

1. In is conflict-free.

   Assume for contradiction that In contains arguments A and B such that A defeats B. Then by condition (2) of Definition 2.3.1 B is in Out. But since $In \cap Out = \varnothing$, we have that B is not in In. Contradiction.

2. In defeats every argument outside In.

   Since stable status assignments assign a status to all arguments in $Args$ and $In \cap Out = \varnothing$, every argument outside In is in Out. Then by condition (2) of Definition 2.3.1 every such argument is defeated by an argument in In.

$\Leftarrow$:

Suppose S is a stable extension. To be proven: $(S, Args/S)$ is a stable status assignment. Note first that by construction this is a partition of $args$. Then it must be vreified that the two labelling conditions of Definition 2.3.1 are satisfied.

1. Condition (1) of Definition 2.3.1 is satisfied as follows. For the only if-part, if $A \in S$ then since S is conflict-free, no $B \in S$ defeats A, so all defeaters of A are in $Args/S$. But then since S is a stable extension, they are all defeated by an argument in S. For the if-part, if all defeaters of an argument A are in $Args/S$, then since S defeats all arguments outside it, they are all defeated by an argument in S.

2. Condition (2) of Definition 2.3.1 is satisfied as follows. For the only-if part, suppose $A \in Args/S$. Then since S defeats all arguments outside it, A is defeated by an argument in S. For the if-part, suppose A is defeated by an argument in S. Then since S is conflict-free, $A \in Args/S$. □

Below we shall use the term *stable extension* both for stable argument extensions and for Dung's stable extensions.

Example 2.1.3 has only one stable extension, viz. $\{A, C\}$, while Example 2.1.4 has two, induced by the following two status assignments:

Recall that an argumentation system is supposed to define when it is justified to accept an argument. What can we say in case of A and B in Example 2.1.4? Since both of them are 'in' in one stable status assignment but 'out' in the other, we must conclude that with respect to stable semantics neither of them is justified. This is captured by the following definition:

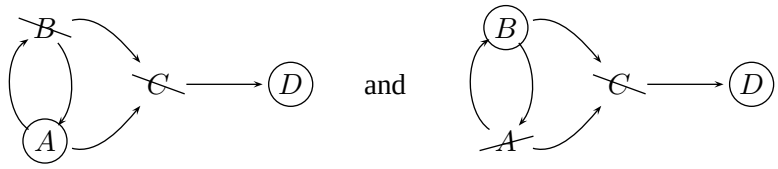
Definition 2.3.5 [Justified arguments in stable semantics.] With respect to stable semantics, an argument is *justified* iff it is 'in' in all stable status assignments.

However, this is not all; just as in the unique-status-assignment approach, it is possible to distinguish between two different categories of arguments that are not justified. Some of those arguments are in no stable status assignment, but others are at least in some extensions. The first category can be called the *overruled*, and the latter category the *defensible* arguments.

Definition 2.3.6 [Overruled and defensible arguments in stable semantics.] With respect to stable semantics, an argument is:

- *overruled* iff it is 'out' in all stable status assignments;
- *defensible* iff it is 'in' in some but not in all stable status assignments.

It is easy to see that the unique-assignment and multiple-assignments approaches are not equivalent. Consider again Example 2.2.9. Argument A and B form an even defeat loop, thus, according to the multiple-assignments approach, either A and B can be assigned 'in' but not both. So the above defeat relation induces stable two status assignments:



While in the unique-assignment approach all arguments are defensible, we now have that, while A and B are defensible, D is justified and C is overruled.

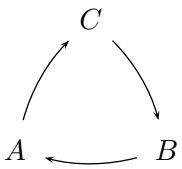
Multiple status assignments also make it possible to capture floating conclusions. Informally, this can be done by defining that a formula φ is justified as 'all extensions contain an argument for φ', rather than as 'there exists an argument for φ that is in all extensions'. In Chapter 4, in which the structure of arguments is formally defined, these alternative consequence notions for formulas will be fully formalised.

2.3.2 Preferred semantics

There is reason to discuss a second variant of the multiple-status-assignments approach. Since a stable extension is conflict-free, it reflects in some sense a coherent point of view. It is also a maximal point of view, in the sense that every possible argument is either accepted or rejected. In fact, stable semantics is the most 'aggressive' type of semantics, since a stable extension defeats every argument not belonging to it, whether or not that argument is hostile to the extension.

This feature is the reason why not all AT's have stable extensions, as the following example shows. It contains an 'odd loop' of defeat relations.

Example 2.3.7 (Odd loop.) Let A, B and C be three arguments, represented in a triangle, such that A defeats C, B defeats A, and C defeats B.



In this situation, Definition 2.3.1 has some problems, since this example has no stable status assignments.

1. Assume that A is 'in'. Then, since A defeats C, C is 'out'. Since C is 'out', B is 'in', but then, since B defeats A, A is 'out'. Contradiction.

2. Assume next that A is 'out'. Then, since A is the only defeater of C, C is 'in'. Then, since C defeats B, B is 'out'. But then, since B is the only defeater of A, A is 'in'. Contradiction.

Note that a self-defeating argument is a special case of Example 2.3.7, viz. the case where B and C are identical to A. This means that argumentation theories containing a self-defeating argument may have no stable status assignment.

To give such examples also a multiple-assignment semantics, we need the notion of a *partial* status assignment.

Definition 2.3.8 [Status assignments.]

1. A *status assignment* on the basis of an abstract argumentation theory $\langle Args, defeat \rangle$ is an assignment to zero or more arguments in *Args* of either the status 'in' or the status 'out' (but not both), satisfying the following conditions:
   (a) An argument is *in* iff all arguments defeating it (if any) are out.
   (b) An argument is *out* iff it is defeated by an argument that is in.
2. A status assignment is *complete* iff it assigns a status to all arguments in $Args$; otherwise it is *partial*.
3. A status assignment is *preferred* iff it is a maximal status assignment.

Sometimes we shall represent a status assignment to a set $Args$ as a pair (S_1, S_2), where S_1 is the set of arguments assigned 'in' and S_2 the set of arguments assigned 'out'.

Corollary 2.3.9 A status assignment is stable iff it is complete.

We must still make the notion of a maximal status assignment precise.

Definition 2.3.10 [Maximal status assignments.] A status assignment $S = (In, Out)$ is *maximal* iff there is no status assignment $S' = (In', Out')$ such that $In \cup Out \subset In' \cup Out$.

To go back to Example 2.3.7, preferred semantics gives it a unique preferred status assignment, viz. $(\varnothing, \varnothing)$.

The notions of justified, overruled and defensible arguments defined in Definitions 2.3.5 and 2.3.6 can be easily defined also for preferred semantics, by uniformly replacing 'stable' by 'preferred'. However, in preferred semantics there are reasonable alternatives for the definitions of defensible and overruled arguments (and conclusions). This is because in each status assignment the status of an argument can be one of three kinds: 'in', 'out' or undefined. Hence there are, unlike in stable semantics, situations where an argument is 'in' in some but not in all assignments but yet not 'out' in any assignment. Likewise, there are situations where an argument is 'out' in some but not in all assignments but yet not 'in' in any assignment. In the remainder of this reader we will for simplicity interpret the notions of defensible and overruled arguments as defined in Definitions 2.3.6.

To return to the notion of preferred extensions, Dung (1995) defines it not in terms of partial status assignments but with the notion of an admissible set, which in turn is defined in terms of acceptability.

Definition 2.3.11 [conflict-free and admissible sets.]

1. A set of arguments is *conflict-free* iff no argument in the set defeats an argument in the set.

2. A set of arguments S is *admissible* iff S is conflict-free and each argument in S is acceptable with respect to S.

Intuitively, an admissible set represents an admissible, or defendable, point of view. In Example 2.1.3 the sets $\varnothing$, $\{C\}$ and $\{A, C\}$ are admissible but all other subsets of $\{A, B, C\}$ are not admissible.

Definition 2.3.12 [Preferred extensions.] A conflict-free set of arguments is a *preferred extension* iff it is a maximal (with respect to set inclusion) admissible set.

There is a one-to-one correspondence between maximal status assignments and preferred extensions.

Proposition 2.3.13

1. If (In, Out) is a status assignment, then In is an admissible set;

2. Let $Out(E)$ be the set of all arguments defeated by E. If E is a preferred extension, then $(E, Out(E))$ is a status assignment;

3. (In, Out) is a maximal status assignment iff In is a preferred extension.

Proof: We first prove the following lemma (which is Lemma 10 of Dung 1995).

(*) If E is an admissible set and A is acceptable wrt E, then $\{A\} \cup E$ is admissible.

Proof of (*): It suffices to show that $\{A\} \cup E$ is conflict-free. Assume for contradiction the contrary. Then there is a $B \in E$ such that either A defeats B or B defeats A. Since E is admissible and A is acceptable wrt E, there is a B' in E such that B' defeats B or B' defeats A. Since E is conflict-free, it follows that B' defeats A. But then there is an argument B'' in E such that B'' defeats B'. Contradiction. □
Proof of (1):
Let (In, Out) be any status assignment and A be any member of In. Observe first that In is conflict-free. Next, all arguments defeating A are in Out, so all arguments defeating A are defeated by In. But then In is an admissible set.
Proof of (2):
Let E be any preferred extension. Condition 1b of Definition 2.3.8 is satisfied by definition of $Out(E)$. To verify condition 1a, observe first that all members of E are acceptable with respect to E, so all their defeaters are in $Out(E)$. Next, let A be any argument such that all its defeaters are in $Out(E)$. Then A is acceptable with respect to E, and by (*), $\{A\} \cup E$ is admissible. But then, since E is maximally admissible, it follows that $A \in E$.
Proof of (3), ⇒:
Consider any maximal status assignment (In, Out). By (1), In is admissible. To prove that In is maximally admissible, assume for contradiction that there is an admissible set $In' \supset In$. By a result of Dung (1995) we may without loss of generality assume that In' is maximally admissible. Then (In', Out') is a status assignment by (2). But

since $In' \supset In$, (In, Out) is not a maximal status assignment. Contradiction.
Proof of (3), $\Leftarrow$:
Assume that E is a preferred extension. By (2), $(E, Out(E))$ is a status assignment. Next, to prove that E is a maximal status assignment, assume for contradiction otherwise, viz. that there is a status assignment (In, Out) such that $In \supset E$. By (1), In is an admissible set. But then E is not maximally admissible. Contradiction. □

It follows from Definition 2.3.12 that:

Proposition 2.3.14 (Dung, 1995) Every abstract argumentation theory has at least one preferred extension.

Proof: We begin by proving that every admissible set is contained in a maximal admissible set. From this the observation follows since the empty set is admissible.

Consider a sequence $S_0 \subseteq \ldots \subseteq S_i \subseteq \ldots$ of admissible sets. Clearly, $S = S_0 \cup \ldots \cup S_i \cup \ldots$ is maximal in this sequence.[4] We prove that S is also admissible by proving that the union of any two elements of S is admissible.

Consider any $S_i, S_j \in S$. Observe first that if $S_i \subseteq S_j$, then since S_j is conflict-free, S_i does not defeat S_j. Suppose next that S_j defeats S_i. Since S_i is admissible, S_i then also defeats S_j. Contradiction. So $S_i \cup S_j$ is conflict-free. Next, since S_i as well as S_j defeats each argument that defeats one of its members, the same holds for $S_i \cup S_j$, so that this set is admissible. □

Grounded status assignments It turns out that grounded semantics can also be formulated in terms of status assignments, namely, as those assignments that are mininal in the following sense.

Definition 2.3.15 [Minimal status assignments.] A status assignment $S = (In, Out)$ is *minimal* iff there is no status assignment $S' = (In', Out')$ such that $In' \cup Out' \subset In \cup Out$.

Proposition 2.3.16 (Caminada, 2006) S is the grounded extension of AT if and only if (S, Out) is a minimal status assignment of AT.

Self-defeat in preferred semantics Finally, how does preferred semantics deal with self-defeating arguments? It turns out that, just as in grounded semantics, self-defeating arguments can prevent other arguments from being justified. This can be illustrated with Example 2.2.8 (two arguments A and B such that A defeats A and A defeats B). The set $\{B\}$ is not admissible, so the only preferred extension is the empty set. As said above, a full analysis of self-defeat requires that the internal structure of arguments is made explicit; this will be further discussed in Chapter 4, Section 4.4.

2.4 Formal relations between grounded, stable and preferred semantics

We now give some results on the relation between the various semantics proven by Dung (1995).

[4] Strictly speaking, this follows from a result in lattice theory.

Proposition 2.4.1 Every stable extension is preferred, but not vice versa.

Proof: It is clear that each stable extension is a preferred extension. And Example 2.2.8 shows that the reverse does not hold: the empty set is a preferred extension of this argumentation theory, but it is not stable. □

The following results are listed without proofs.

1. The grounded extension is contained in the intersection of all preferred extensions (Example 2.2.9 is a counterexample against 'equal to').

2. If an abstract argumentation theory does not give rise to infinite paths $A_1, \ldots, A_n, \ldots$ through the defeat graph such that each A_{i+1} defeats A_i then it has exactly one stable extension, which is also grounded and preferred. (Note that the even loop of Example 2.1.4 and the odd loop of Example 2.3.7 give rise to such an infinite defeat path.)

3. Finally, Dung (1995) identifies conditions under which preferred and stable semantics coincide. A necessary condition is that an abstract argumentation theory does not contain odd defeat loops.

2.5 Comparing the two approaches

How do the unique- and multiple-assignment approaches compare to each other? It is sometimes said that their difference reflects a difference between a 'skeptical' and 'credulous' attitude towards drawing defeasible conclusions: when faced with an unresolvable conflict between two arguments, a skeptic would refrain from drawing any conclusion, while a credulous reasoner would choose one conclusion at random (or both alternatively) and further explore its consequences. The skeptical approach is often defended by saying that since in an unresolvable conflict no argument is stronger than the other, neither of them can be accepted as justified, while the credulous approach has sometimes been defended by saying that the practical circumstances often require a person to act, whether or not s/he has conclusive reasons to decide which act to perform.

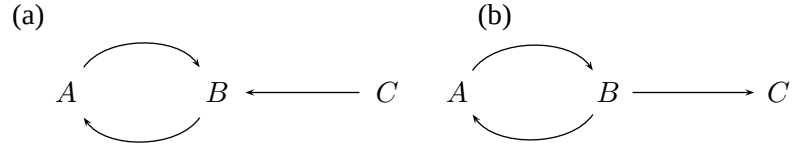
In our opinion the notions of skeptical and credulous reasoning do not exclude but completement each other: whether it is be better to reason skeptically or credulously may depend on the application context. For example, for a judge in a law court the reasoning about whether the suspect is guilty must clearly be skeptical, while for an intelligent software agent faced with two conflicting goals it makes sense to reason credulously, to achieve at least one of the goals. Moreover, it seems wrong to equate the distinction skeptical-credulous with the distinction between the unique- and multiple-status-assignment approach. When deciding what to accept as a justified belief, what is important is not whether one or more possible status assignments are considered, but how the arguments are ultimately evaluated given these assignments. And this evaluation is captured by the qualifications 'justified' and 'defensible', which thus capture the distinction between 'skeptical' and 'credulous' reasoning. And since, as we have seen, the distinction justified vs. defensible arguments can be made in both the unique-assignment and the multiple-assignments approach, these approaches are independent of the distinction 'skeptical' vs. 'credulous' reasoning.

As for their outcomes, the approaches mainly differ in their treatment of floating arguments and conclusions. With respect to these examples, the question easily arises whether one approach is the right one. However, we prefer a different attitude: instead of speaking about the 'right' or 'wrong' definition, we prefer to speak of 'senses' in which an argument or conclusion can be justified. For instance, the sense in which the conclusion that Brigt Rykkje likes ice skating in Example 2.2.10 is justified is different from the sense in which, for instance, the conclusion that Tweety flies in Example 2.1.3 is justified: only in the second case is the conclusion supported by a justified argument. And the status of D in Example 2.2.9 is not quite the same as the status of, for instance, A in Example 2.1.3. Although both arguments need the help of other arguments to be justified, the argument helping A is itself justified, while the arguments helping D are merely defensible. Again it may depend on the application context which sense of justification is the best.

To conclude this chapter, Dung's fully abstract approach was a major innovation in the study of defeasible argumentation, in that it provided an elegant general framework for investigating the various argumentation systems. Moreover, the framework also applies to other nonmonotonic logics, since Dung showed how several of these logics can be translated into argumentation systems. Thus it becomes very easy to formulate alternative semantics for nonmonotonic logics. For instance, default logic (Reiter, 1980), which was by Dung shown to have a stable semantics, can very easily be given an alternative semantics, like preferred or grounded semantics. Moreover, the proof theories that have been or will be developed for the various argument-based semantics immediately apply to the systems that are an instance of these semantics. On the other hand, the fully abstract nature of Dung's framework also leaves much to the developers of particular systems. In particular, they have to define the internal structure of an argument, the ways in which arguments can conflict, and the origin of the defeat relation. In the next chapter a more concrete framework will be discussed in which these elements have been defined.

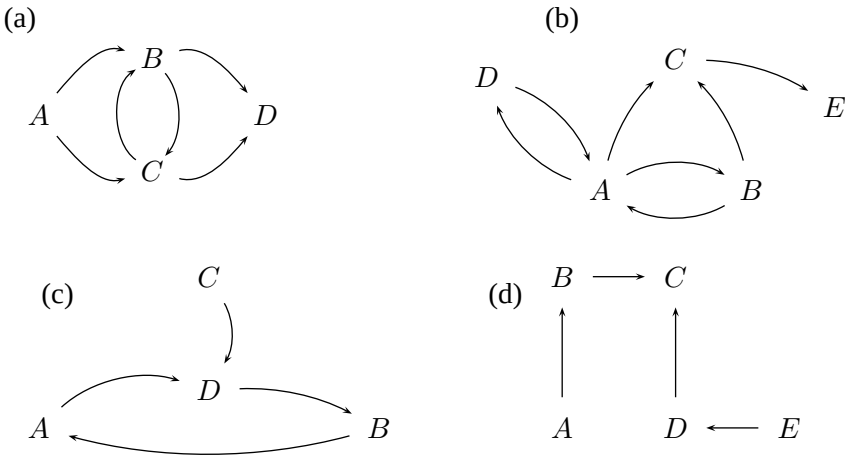
2.6 Exercises

EXERCISE 2.6.1 Determine, if possible, with Definition 2.1.2 which arguments are justified in the following two examples.



EXERCISE 2.6.2 Prove that if no argument of AAT is undefeated, then $F^{AAT}(\varnothing) = \varnothing$.

EXERCISE 2.6.3 Determine the grounded extension of the following defeat graphs. Show in each case its construction as in Proposition 2.2.4.



EXERCISE 2.6.4 Let

- $G(S) = \{A \in Args \mid A \text{ is not defeated by a member of } S\}$

1. Show that, for every set of arguments X, $F(X) = G^2(X)$ $[= G(G(X))]$.
2. Show that G is anti-monotonic. G is anti-monotonic if $A \subseteq B$ implies $G(B) \subseteq G(A)$.
3. Show on the basis of (2) that F is monotonic.
4. Let $\{G_i\}_{i \geq 0}$ be sets of arguments, such that

$$\begin{aligned} G_0 &=_{Def} \varnothing, \\ G_i &=_{Def} G(G_{i-1}). \end{aligned}$$

   Show that $G_0 \subseteq G_2 \subseteq G_4 \subseteq \ldots \subseteq G_5 \subseteq G_3 \subseteq G_1$.

EXERCISE 2.6.5 Determine for each of the defeat graphs in Exercise 2.6.3 which arguments are justified, which are defensible and which are overruled, all according to grounded semantics.

EXERCISE 2.6.6 Prove that S is a stable extension iff $S = \{A \mid A \text{ is not defeated by } S\}$.

EXERCISE 2.6.7 Determine all status assignments in Examples 2.1.3, 2.1.4 and 2.3.7. Which of these assignments are maximal?

EXERCISE 2.6.8 Consider two status assignments $S = (In, Out)$ and $S' = (In', Out')$ to the same argumentation theory such that $In \subset In'$.

1. Does it hold that $Out \subseteq Out'$? If so, give the proof; if not, give a counterexample.
2. Does it hold that $Out \subset Out'$? Again, if so, give the proof; if not, give a counterexample.

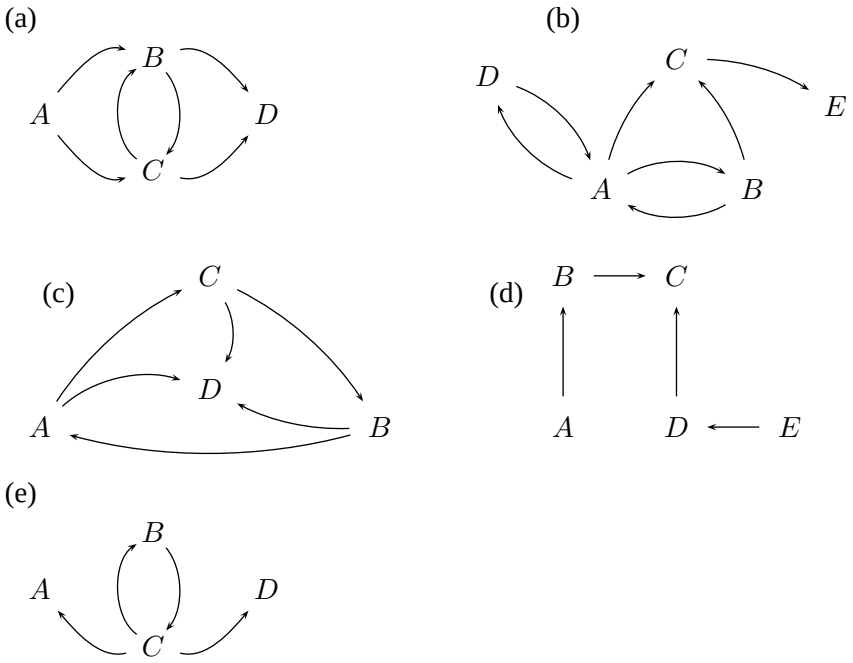
EXERCISE 2.6.9 Determine all status assignments in Examples 2.1.3, 2.1.4 and 2.3.7. Which of these assignments are maximal?

EXERCISE 2.6.10 Give one or more alternative definitions of the notions of defensible and overruled arguments in preferred semantics. Verify for each definition whether it implies that each argument is either justified, or defensible, or overruled. If not, do you regard this as a flaw of your definition?

EXERCISE 2.6.11 Determine the admissible sets in Example 2.3.7. Which of these is or are maximally admissible?

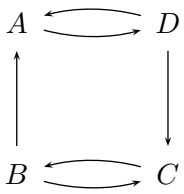
EXERCISE 2.6.12

1. Determine the preferred and stable extension(s) of the following defeat graphs.



2. Determine for each of the above defeat graphs, and with respect to each semantics, which arguments are justified, which are defensible and which are overruled.

EXERCISE 2.6.13 Consider four arguments A, B, C and D such that B strictly defeats A, D strictly defeats C, A and D defeat each other and B and C defeat each other.



Here is a natural-language version, in which the defeat relations are based on which argument uses the more specific of two conflicting defaults.

$A =$ Larry is rich because he is a public defender, public defenders are lawyers, and lawyers are rich;
$B =$ Larry is not rich because he is a public defender, and public defenders are not rich;
$C =$ Larry is rich because he lives in Hollywood, and people who live in Hollywood are rich;
$D =$ Larry is not rich because he rents in Hollywood, and people who rent in Hollywood are not rich.

1. Determine the grounded extension and the preferred extension(s) of this argumentation theory.

2. Determine in both cases which conclusions about Larry's richness are justified. Does the result agree with your intuitions?

Chapter 3

Games for abstract argumentation

So far mainly semantical aspects have been discussed, where the main focus was on characterising properties of *sets* of arguments, without specifying procedures for determining whether a given argument is a member of the set. In this chapter we shall go deeper into proof-theoretical, or procedural aspects of argumentation, where the chief concern is to investigate the status of *individual* arguments. This aspect of argumentation logics is less well-developed than its semantics; much research is ongoing or still to be carried out.

3.1 General ideas

The main question of this chapter is: given an argument from an abstract argumentation theory, how can its status be investigated? Several argumentation systems have tackled this problem in dialectical style. The common idea can be explained in terms of an argument game between two players, a proponent and an opponent of an argument. A dispute is an alternating series of moves by the two players. The proponent starts with an argument to be tested, and each following move consists of an argument that defeats (or in some cases strictly defeats) a move of the other party. The initial argument provably has a certain dialectical status if the proponent has a winning strategy, i.e., if he can win whatever moves the opponent makes.

The exact rules of the game depend on the semantics the game is meant to capture. A common winning criterion is that a player has won if s/he has made the other player run out of moves. However, other criteria are also possible. Other aspects on which choices have to be made are:

- Must moves strictly defeat their target or can they be weakly defeating?
- May moves be repeated?
- May players backtrack?
- May players defeat or be defeated by their own earlier moves?

These choices have to be made independently for both sides.

A natural idea in dialectical proof theories is that of dialectical asymmetry. The players of an argument game have different objectives: proponent wants to build a (dialectical) proof, while opponent wants to prevent proponent from doing so. In other words, while proponent is constructive, opponent is destructive, and this leads to different rules for the two players. Moreover, the burden induced by these rules will be heavier for one player than for the other. Which player has the heavier burden depends

on whether the reasoning is credulous or skeptical: in skeptical reasoning the heavier burden is on proponent, while in credulous reasoning it is on opponent.

Let us now make these informal observations more precise. A dialectical proof theory takes the form of an argument game regulating a *dispute* between two *players*, the proponent P and opponent O of an argument. If p is a player, then $\overline{p}$ denotes the other player. The players *move* alternatingly, moving one argument at each turn. The game has a *protocol* function for determining *legality* of moves, by defining at each point in a dispute which arguments can be moved. Finally, a *winning criterion* is a partial function that determines the winner of a dispute, if any. If one player wins, the other player loses, so the argument game is a so-called zero-sum game.

These notions are formally defined as follows (relative to a given argumentation theory; in the rest of this chapter we shall, unless stated otherwise, implicitly assume an arbitrary but fixed argumentation theory).

Definition 3.1.1 [Moves, disputes and protocols.] Given an argumentation theory $AT = \langle Args, defeat \rangle$ we define the following notions.

- The set M of *moves* consists of all pairs (p, A) such that $p \in \{P, O\}$ and $A \in Args$; for any move (p, A) in M we denote p by $pl(m)$ and A by $s(m)$.
- The set of $M^{\leq\infty}$ of *disputes* is the set of all sequences from M and the set $M^{<\infty}$of *finite disputes* is the set of all finite sequences from M.
- A *protocol* is a function that specifies the *legal moves* at each stage of a dispute. Formally, protocol is a function Pr with domain a nonempty subset D of $M^{<\infty}$ taking subsets of M as values. That is:
  - $Pr : D \longrightarrow Pow(M)$

  such that $D \subseteq M^{<\infty}$. The elements of D are called the *legal finite disputes*. The elements of $Pr(d)$ are called the moves allowed after d. If d is a legal dispute and $Pr(d) = \varnothing$, then d is said to be a *terminated* dispute. Pr must satisfy the following conditions for all finite disputes d and moves m:
  1. $d \in D$ and $m \in Pr(d)$ iff $d, m \in D$;
  2. if $m \in Pr(d)$ then $pl(m) = P$ if d is of even length, otherwise $pl(m) = O$.
- A *winning function* is a partial function of type $W : D \longrightarrow \{P, O\}$.

The crucial elements of this definition are the protocol and the winning criterion. Dialectical proof theories differ only on these two elements.

We now define an abstract game-theoretic notion of defeasible provability, which is the same for all dialectical proof theories. It is defined in terms of the notion of a strategy. A strategy for a player in a dispute game has the form of a tree of disputes that for each possible move of the other player specifies a unique reply.

Definition 3.1.2 [Strategies.]

1. A *strategy* for player p is a tree of disputes only branching after p's moves, and containing all legal replies of $\overline{p}$.
2. A strategy for p is *winning* iff p wins all disputes in the strategy.

If the winning criterion is that the other player has no legal moves, then it is easy to see that a winning strategy for a player is a strategy in which all branches end with a move by that player.

Defeasible provability is now defined as follows, parametrised by a protocol X.

Definition 3.1.3 [Provability.] An argument A is *defeasibly provable in the X-game* iff the proponent has a winning strategy in a dispute with as root the argument A that satisfies protocol X.

3.2 Dialectics for grounded semantics

In this section we discuss a proof theory for determining whether an argument is in the grounded extension of a given argumentation theory. Since a grounded extension only contains justified arguments, the dialectical asymmetry favours the opponent: her moves are allowed to be simply defeating[1], while proponent's moves must be strictly defeating. Moreover, the proponent is not allowed to repeat his arguments. Finally, backtracking is not allowed for both players.

Definition 3.2.1 [Proof theory for grounded semantics.] A dispute satisfies the *G-game* protocol iff it satisfies the following conditions.

1. Moves are legal iff in addition to Definition 3.1.1 they satisfy the following conditions.
   (a) Proponent does not repeat his moves; and
   (b) Proponent's moves (except the first) strictly defeat opponent's last move; and
   (c) Opponent's moves defeat proponent's last move.
2. A player wins a dispute iff the other player has no legal moves.

A dispute satisfying the protocol of the G-game is called a G-dispute.

Example 3.2.2 Let A, B, C and D be arguments such that B and D defeat A, and C defeats B. Then a G-dispute on A may run as follows:

P: A, O: B, P: C

In this dispute P attempts to show A justified. Both B and D defeat A, which means that O has two choices in response to A. O chooses to respond with B in the second move. Then C is the only argument defeating B, so that P has no choice than to respond with C in the third move. There are no arguments against C, so that O cannot move and loses the dispute.

However, this outcome is not inevitable for O; her loss was merely caused by her weak play. A dispute in which O follows an optimal strategy is

P: A, O: D

[1] When below we say that move m defeats move m' we mean that $s(m)$ defeats $s(m')$.

And P has no reply, so O wins. Concluding, in this example P has no winning strategy. The only reason why P wins the first dispute is that O chooses the wrong argument, viz. B, in response to A. In fact, O is in the position to win every game, provided it chooses the right moves. In other words, O possesses a winning strategy.

Example 3.2.3 To give an another example, consider two strategies for P as depicted in Figure 3.1. The tree on the left is based on an argumentation theory AT_1 with $Args = \{A, B, C, D, E, F, G\}$ and *defeat* as shown by the arrows. Here P has a winning strategy, since in all disputes O eventually runs out of moves; so argument A is provable on the basis of AT_1. The tree on the right is based on an extension of AT_1 into AT_2 by adding H, I and J to $Args$ and adding new defeat relations corresponding to the new arrows (the extension is shown inside the dotted box). This is not a winning strategy for P, since one dispute ends with a move by O; so (assuming P has no better strategy) A is not provable on the basis of AT_2.

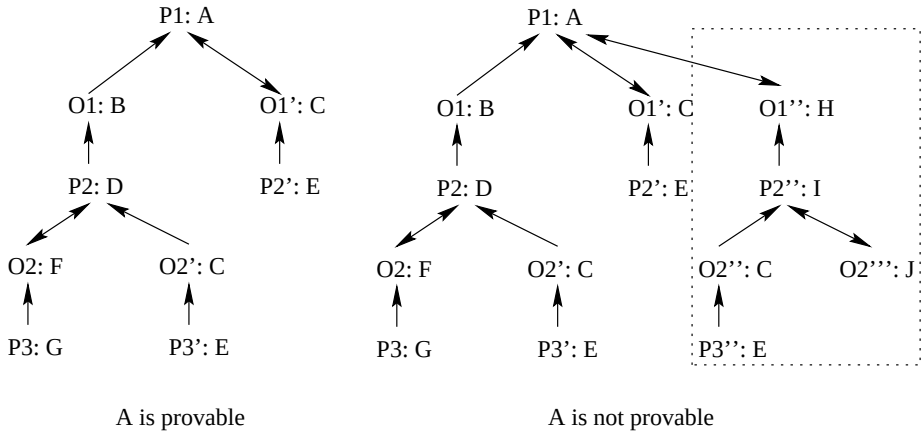


Figure 3.1: Two trees of proof-theoretical disputes.

Some words are in order on the non-repetition requirement of Definition 3.2.1 (condition 1a). This requirement does not change provability of any argument, since O will have a reply the second time iff she had a reply the first time. However, it avoids infinite disputes if $Args$ is finite, which is especially convenient for computational purposes. The same holds for the condition that P's arguments are strictly defeating; allowing them to be simply defeating does not change provability, but it avoids certain infinite disputes.

As for the relation between grounded semantics and its proof theory, the following proposition holds.

Proposition 3.2.4 [Soundness and completeness of the G-game.] An argument is in the grounded extension of an AT iff it is defeasibly provable on the basis of AT in the G-game.

Proof: (Sketch). We give a sketch of the proof for finitary AT's. Without this restriction the proof is more complicated. The restriction makes sense for computational purposes, since saying that an AT is finitary is equivalent to saying that each strategy based on AT has at most a finite number of branches.

$\Leftarrow$ (soundness):
Assume that P has a winning strategy W for A. Clearly, all of W's leaves A_n are in F^1, since they have no defeaters. But then in every branch of W, A_{n-2} is acceptable with respect to F^1 and so is in F^2. This can be repeated until the root of W is reached. □

$\Rightarrow$ (completeness):
Suppose A is in the grounded extension of AT. Then, since AT is finitary, there is a least number i such that $A \in F^i$. Then P has the following winning strategy if he begins a dispute with A. For each argument B defeating A moved by O, P can choose one argument C from F^{i-1} that strictly defeats B. This can be repeated for each argument defeating C, and so on, until P can choose an argument from F^1, which has no defeaters, so O has no legal reply. □

Note that completeness here does not imply semi-decidability (a logic is semi-decidable iff there exists an algorithm that can produce any provable formula): if the logic for constructing individual arguments is not decidable, then the search for counterarguments is in general not even semi-decidable, since this search is essentially a consistency check.

This completes the discussion of the dialectical proof theory for grounded semantics. We now turn to a dialectical proof theory for credulous reasoning, in particular for preferred semantics.

3.3 Dialectics for preferred semantics

In this section we present the so-called P- game[2], which serves as a credulous proof theory for preferred semantics, and was developed by Vreeswijk and Prakken (2000). For notational convenience we now denote defeat relations with $\leftarrow$. Throughout this section we will use the following example.

Example 3.3.1 The pair $\mathcal{A} = \langle X, \leftarrow \rangle$ with arguments

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

and $\leftarrow$ as indicated in Figure 3.2 is an (abstract) example of an argumentation theory. It accommodates a number of interesting cases, and will therefore be used as a running example throughout this chapter.

3.3.1 The basic ideas illustrated

Example 3.3.1 gives us some useful clues as to which features the argument game for preferred semantics should have. We are interested in credulous reasoning, so in testing membership of *some* extension. The argument game is based on the following idea. By definition, a preferred extension is a $\subseteq$-maximal admissible set. It is known that each admissible set is contained in a maximal admissible set (see the proof of Proposition 2.3.14), so the procedure comes down to trying to construct an admissible set 'around' the argument in question. If this succeeds, we know that the admissible set and hence the argument in question is contained in a preferred extension.

[2]The P in 'P-game should not be confused with the P denoting proponent.

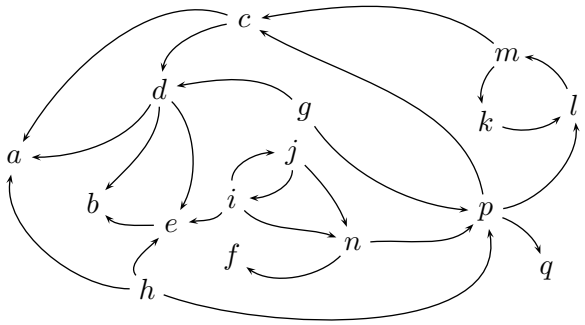
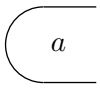


Figure 3.2: Defeat relations in the running example.

Suppose now we wish to investigate whether a is preferred, i.e., belongs to a preferred extension. We know that it suffices to show that the argument in question is admissible. The idea is to start with $S = \{a\}$ and, if a has defeaters, to find other arguments in order to complete S into an admissible set.

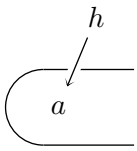
Example 3.3.2 (Straight failure). Consider the argument system of Figure 3.2, and suppose that P's task is to show that a is preferred.

The first action of P is simply putting forward a:



If a cannot be defeated, then $S = \{a\}$ is admissible, and P succeeds. However, since $a \leftarrow h$,

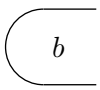
O forwards h:



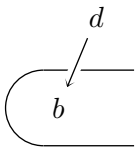
Now it is up to P to defend a by finding arguments against h. There are no such arguments, so that P fails to construct an admissible set 'around' a. So a is not admissible, hence not preferred.

Example 3.3.3 (Straight success). Suppose that P wants to show that b is admissible.

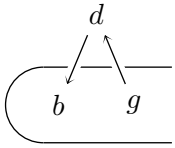
The first action of P is putting forward b:



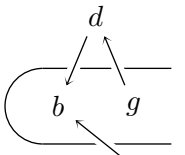
O defeats b with d:



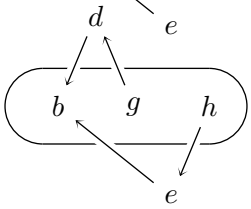
P defends this attack with g:



Since O 's attack on b with d has failed, O returns to b and defeats it again, this time with e:



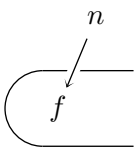
P defends b again, this time with h. Since O is unable to find other arguments against b, g or h, P may now close S:



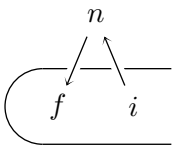
Example 3.3.4 (Even loop success). Suppose that P wants to show that f is admissible.

The first action of P is putting forward f:

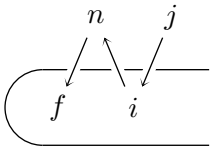
O defeats f with n:



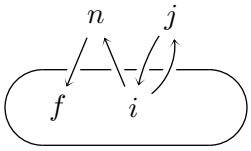
P defends this attack with i:



O defeats i with j:



P defends i with i itself (so that i is self-defending). O is unable to put forward other arguments that defeat f or i so that P closes S:

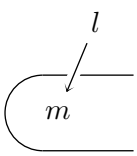


This example shows that P must be allowed to repeat his arguments, while O must be forbidden to repeat O 's arguments (at least in the same 'line of dispute'; see further below).

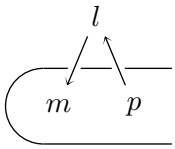
Example 3.3.5 (Odd loop failure). Suppose that P wants to show that m is admissible.

The first action of P is putting forward m:

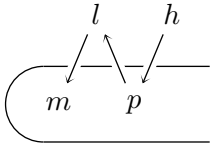
O defeats m with l:



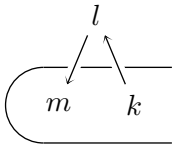
P defends this attack with p:



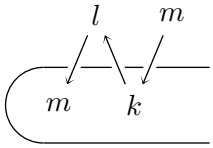
O defeats p with h:



P backtracks and removes p from S. He then tries to defend l with k instead:

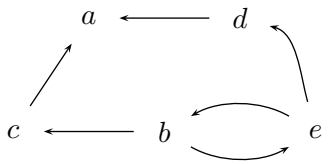


O defeats k with m (and, as a bonus, introduces an inconsistency in S):



P has no other arguments in response to l and m, so that he is unable to close S into an admissible set. So m is not contained in an admissible set. Note that we cannot allow P to reply to m with l, since otherwise the set that P is constructing 'around' m is not conflict-free, hence not admissible. So we must forbid P to repeat O 's moves. On the other hand, this example also shows that O should be allowed to repeat P 's moves, since such a repetition reveals a conflict in P 's position.

Example 3.3.6 (The need for backtracking). The next feature of our argument game is not illustrated by Figure 3.2 so we need a new example. Consider an argument system with five arguments a, b, c, d and e and defeat relations as shown in the graph.



This example shows that we must allow O to backtrack. Suppose P starts with a, O defeats a with d, and P defends a with e. If O now defeats e with b, P can defend e by repeating e itself. However, O can backtrack to a, this time defeating it with c, after which P can only defend a with b which repeats O, and in Example 3.3.5 we concluded that P must be forbidden to do so. So by backtracking O can reveal that P's position is not conflict-free.

Repetition

Let us summarise our observations about repetition of moves.

i. It makes sense for P to repeat himself (if possible), because O might fail to find or produce a new defeater of P's repeated argument. If so, then P's repetition closes a cycle of even length, of which P's arguments are admissible.

ii. It makes sense for O to repeat P (if possible), because thus she shows that P's collection of arguments is not conflict-free.

iii. P must not repeat O, because doing so would introduce a conflict into P's own collection of arguments.

iv. O must not repeat herself, because P has already shown to have adequate defense for O's previous arguments.

3.3.2 The P-game defined

We now turn to the formal definition of the argument game for preferred semantics. Let us fix some terminology.

- A *dispute line* is a dispute without backtracking moves.
- An *eo ipso* (meaning: "you said it yourself") is a move that uses a previous argument of the other player.

Definition 3.3.7 [A proof theory for preferred semantics.] A dispute satisfies the *P-game* protocol iff satisfies the following conditions.

1. Moves are legal iff in addition to Definition 3.1.1 they satisfy the following conditions.
   (a) A move by P responds to the previous move by O.
   (b) A move by O responds to some earlier move by P.
   (c) A move defeats the argument to which it responds.
   (d) P does not repeat O's moves.
   (e) O does not repeat O's moves in the same dispute line.
   (f) No two responses to the same move have the same content.
2. O wins a dispute iff she does an *eo ipso* or makes P run out of legal moves; otherwise P wins.

A dispute satisfying the rules of the P-game is called a P-dispute.

Note that an infinite dispute is won by P.

Since the P-game allows O to backtrack, during a P-dispute a tree of dispute lines is constructed. (By contrast, a G-dispute consists of only one dispute line, since in a G-dispute each argument replies to the immediately preceding move in the dispute.) Accordingly, there are two ways to display a P-dispute: as a *linear* structure, in the order in which the arguments are moved, and as a *tree* structure, where the edges indicate to which argument an argument replies. The reader should not confuse the tree form of a single dispute with the tree form of a strategy: in the latter tree (cf. Definition 3.1.2) an edge between two arguments indicates that the child argument is moved immediately after the parent argument; in other words, each branch of a strategy tree is a complete dispute, possibly with backtracking moves, but displayed in linear form.

Proposition 3.3.8 [Soundness and completeness of the P-game.] An argument is in some preferred extension of an AT iff it is defeasibly provable on the basis of AT in the P-game

Proof: (Below we say that an argument a is *defended* in a dispute iff the dispute begins with a and is won by P.) By definition of preferred extensions it suffices to show that an argument is admissible iff it can be defended in every dispute.

First suppose that a can be defended in every dispute. This includes disputes in which O has opposed optimally. Let us consider such a dispute. Let A be the arguments that P used to defend a. (in particular $a \in A$.) If A is not conflict-free then $a_i \leftarrow a_j$ for some $a_i, a_j \in A$, and O would have done an *eo ipso*, which is not the case. If A is not admissible, then $a_i \leftarrow b$ for some $a_i \in A$ while $b \not\leftarrow A$. In that case, O would have used b as a winning argument, which is also not the case. Hence A is admissible.

Conversely, suppose that $a \in A$ with A admissible. Now P can win every dispute by starting with a, and replying with arguments from A only. (P can do this, because all arguments in A are acceptable wrt A.) As long as P picks his arguments from A, O cannot win by *eo ipso*, because A is conflict-free. So a can be defended in dispute.□

Finally, a drawback of the P-game is that in some cases proofs have to be infinite. This is obvious when an argument has an infinite number of defeaters, but even otherwise some proofs are infinite, as in the case of Example 2.2.5. Nevertheless, it is easy to verify that with a finite set of arguments all proofs are finite.

3.4 Exercises

EXERCISE 3.4.1 Consider an argumentation theory with the arguments $\{A - G\}$ and the following defeat relations: A and B defeat each other, E and G defeat each other, C defeats B, D defeats A, E defeats D, and F defeats D.

1. Draw the defeat graph.
2. Determine all strategies for P and O in a game for A according to grounded semantics. Indicate which of these strategies are winning.

EXERCISE 3.4.2

1. Change Definition 3.2.1 to the effect that the non-repetition rule is dropped, and P's arguments are allowed to be simply defeating. Give a dispute that is finite under the original definition but infinite under the new definition.
2. Answer the same question for the case that only the non-repetition rule is dropped.
3. Give a dispute that is infinite under the original definition.

EXERCISE 3.4.3

1. Investigate for the following arguments in Exercise 2.6.3 whether they can be proven justified with respect to grounded semantics. For each provable argument, give a winning strategy for P. For each argument that is not provable, show why P's strategies fail.
   (a) In (a): investigate A, B and D.
   (b) In (b): investigate C and E.
   (c) In (c): investigate A, B and C.

(d) In (d): investigate C.

2. Answer the same question about defeat graph (e) of Exercise 2.6.12, for the arguments C and D.

3. For each argument under 1 that is provable, compare the structure of P's winning strategy with the construction of the grounded extension that you found in Exercise 2.6.3. How are they related?

EXERCISE 3.4.4 Verify that a proof in the P-game of A_1 in Example 2.2.5 has to be infinite.

EXERCISE 3.4.5 Show with an example that the P-game is incorrect as a proof theory for stable semantics.

EXERCISE 3.4.6

1. Investigate for the following arguments in Exercise 2.6.12 whether they can be proven to be in some preferred extension. For each provable argument, give a winning strategy for P. For each argument that is not provable, show why P's strategies fail.

   (a) All arguments in (b);

   (b) All arguments in (c);

   (c) Argument c in (d).

2. Answer the same question for argument c in Figure 3.2.

Chapter 4

A framework for argumentation with structured arguments

As explained above, Dung's (1995) abstract framework was an important advance in the formal study of argumentation. However, its fully abstract nature makes it less suitable for directly representing specific argumentation problems. It is best used as a tool for analysing particular argumentation formalisms and for developing a metatheory of such systems. When actual applications of argumentation-based inference have to be modelled, Dung's framework should be refined with accounts of the structure of arguments and the nature of the defeat relation. However, here too abstraction is still possible and worthwhile. This chapter instantiates Dung's abstract approach by assuming an unspecified logical language and by defining arguments as inference trees formed by applying two kinds of inference rules, deductive (or 'strict') and defeasible rules'. As explained in Section 1.3, the notion of an argument as an inference tree naturally leads to three ways of attacking an argument: attacking a premise, attacking a conclusion and attacking an inference. To resolve such conflicts, preferences may be used, which leads to three corresponding kinds of defeat: undermining, rebutting and undercutting defeat. To characterise them, some minimal assumptions on the logical object language must be made, namely that certain well-formed formulas are a contrary or contradictory of certain other well-formed formulas. Apart from this the framework is still abstract: it applies to any set of inference rules, as long as it is divided into strict and defeasible ones, and to any logical language with a contrary relation defined over it.

The account offered in this chapter further develops work undertaken in the European ASPIC project (Amgoud *et al.*, 2006; Caminada and Amgoud, 2007) and is more fully reported in (Prakken, 2010). It is based on work of John Pollock (1987; 1994) and Gerard Vreeswijk (1993; 1997) on the structure of arguments, work of Pollock (1974; 1987) on notions of defeat and work of Prakken and Sartor (1997) and others on argumentation with prioritised rules. The proofs of the formal results stated in this chapter can be found in (Prakken, 2010).

4.1 Argumentation systems with structured arguments

In this section the arguments of Dung's argumentation frameworks will be given structure and its defeat relation will be defined in terms of the structure of arguments plus external preference information. The resulting framework unifies two ways to capture the

defeasibility of reasoning. Some, e.g. Bondarenko *et al.* (1997), locate the defeasibility of arguments in the uncertainty of their premises, so that arguments can only be attacked on their premises. Others, e.g. Pollock (1994); Vreeswijk (1997), instead locate the defeasibility of arguments in the riskiness of their inference rules: in these logics inference rules are of two kinds, being either deductive or defeasible, and arguments can only be attacked on their applications of defeasible inference rules. Vreeswijk (1993, Ch. 8) called these two approaches *plausible* and *defeasible* reasoning: he described plausible reasoning as sound (i.e, deductive) reasoning on an uncertain basis, and defeasible reasoning as unsound (but still rational) reasoning on a solid basis. In his chapter 8, Vreeswijk attempted to combine both forms of reasoning in a single formalism, but since then most formal accounts of argumentation have modelled either only plausible or only defeasible reasoning. The present framework again combines the two forms of reasoning but this time within the abstract setting of Dung (1995).

4.1.1 Basic definitions

The basic notion of the present framework is that of an argumentation system, which extends the familiar notion of a proof system with a distinction between strict and defeasible inference rules and a preference ordering on the defeasible inference rules.

Definition 4.1.1 [Argumentation system] An *argumentation system* is a tuple $AS = (\mathcal{L}, ^-, \mathcal{R}, \leq)$ where

- $\mathcal{L}$ is a logical language,
- $^-$ is a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$,
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules such that $\mathcal{R}_s \cap \mathcal{R}_d = \varnothing$,
- $\leq$ is a partial preorder on $\mathcal{R}_d$.

($2^{\mathcal{L}}$ denotes the powerset of $\mathcal{L}$, that is, the set of all its subsets.) Amgoud *et al.* (2006) and Caminada and Amgoud (2007) assume that arguments are expressed in a logical language that is left unspecified except that it is closed under classical negation. In this chapter this assumption will be generalised in two ways. Firstly, non-symmetric conflict relations between formulas will be allowed, such as the contrariness relation of Bondarenko *et al.* (1997) (which can, for instance, be used to express negation as failure as in logic programming or a consistency check as in default logic). Secondly, in addition to classical negation, other symmetric conflict relations will be allowed, so that, for instance, formulas like 'bachelor' and 'married' can, if desired, be declared contradictory without having to reason with an axiom $\neg$(bachelor $\wedge$ married).

Definition 4.1.2 [Logical language] Let $\mathcal{L}$, a set, be a logical language and $^-$ a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$. If $\varphi \in \overline{\psi}$ then if $\psi \notin \overline{\varphi}$ then φ is called a *contrary* of ψ, otherwise φ and ψ are called *contradictory*. The latter case is denoted by $\varphi = -\psi$ (i.e., $\varphi \in \overline{\psi}$ and $\psi \in \overline{\varphi}$).

Unless specified otherwise, in examples below the contrariness function will for simplicity be assumed to conform to classical negation. That is, if φ does not start with a negation then $\neg\varphi \in \overline{\varphi}$ while if φ is of the form $\neg\psi$ then $\psi \in \overline{\varphi}$.

Now that the notion of negation has been generalised, the same must be done with the notion of consistency.

Definition 4.1.3 [consistent set] Let $\mathcal{P} \subseteq \mathcal{L}$. $\mathcal{P}$ is *consistent* iff $\nexists\, \psi, \varphi \in \mathcal{P}$ such that $\psi \in \overline{\varphi}$, otherwise it is *inconsistent*.

Note that this is a weak form of consistency, determined by whether a set contains contrary or contradictory formulas. Caminada and Amgoud (2007) call this *direct consistency* and they call consistency of the closure of a set under strict inference *indirect consistency*.

Arguments are built by applying inference rules to one or more elements of $\mathcal{L}$. Inference rules are either *strict* or *defeasible*.

Definition 4.1.4 [Strict and defeasible rules] Let $\varphi_1, \ldots, \varphi_n, \varphi$ be elements of $\mathcal{L}$.

- A *strict rule* is of the form $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$, informally meaning that if $\varphi_1, \ldots, \varphi_n$ hold, then *without exception* it holds that φ.

- A *defeasible rule* is of the form $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$, informally meaning that if $\varphi_1, \ldots, \varphi_n$ hold, then it *presumably* holds that φ.

$\varphi_1, \ldots, \varphi_n$ are called the *antecedents* of the rule and φ its *consequent*.

As is usual in logic, inference rules will often be specified by schemes in which a rule's antecedents and consequent are metavariables ranging over $\mathcal{L}$.

If an argumentation system is to include standard propositional and/or first-order logic, then the strict rules could be assumed to contain all valid propositional or first-order inferences. Two examples of such inference rules are (in scheme form):

$\varphi, \psi \rightarrow \varphi \wedge \psi$ (for any propositional formulas φ and ψ)
$\forall x Px \rightarrow Pa$ (for any predicate P and constant a).

Possible defeasible inference rules will be discussed in more detail in Section 4.2. As explained there, the main choice to be made is whether these rules are domain-specific (as in e.g. default logic) or whether they express general patterns of reasoning.

Arguments are constructed from a knowledge base, which is assumed to contain three kinds of formulas.

Definition 4.1.5 [Knowledge bases] A *knowledge base* in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}, \leq)$ is a pair $(\mathcal{K}, \leq')$ where $\mathcal{K} \subseteq \mathcal{L}$ and $\leq'$ is a partial preorder on $\mathcal{K} \setminus K_n$. Here $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a$ where these subsets of $\mathcal{K}$ are disjoint and

- $\mathcal{K}_n$ is a set of (necessary) *axioms*. Intuitively, arguments cannot be attacked on their axiom premises.

- $\mathcal{K}_p$ is a set of *ordinary premises*. Intuitively, arguments can be attacked on their ordinary premises, and whether this results in defeat must be determined by comparing the attacker and the attacked premise (in a way specified below).

- $\mathcal{K}_a$ is a set of *assumptions*. Intuitively, arguments can be attacked on their assumptions, where these attacks always succeed.

4.1.2 Arguments

Next the arguments that can be constructed from a knowledge base in an argumentation system are defined. Arguments can be constructed step-by-step by chaining inference rules into trees. Arguments thus contain subarguments, which are the structures that support intermediate conclusions (plus the argument itself and its premises as limiting cases). In what follows, for a given argument, the function `Prem` returns all the formulas of $\mathcal{K}$ (called *premises*) used to build the argument, `Conc` returns its conclusion, `Sub` returns all its sub-arguments, `DefRules` returns all the defeasible rules of the argument and, finally, `TopRule` returns the last inference rule used in the argument.

Definition 4.1.6 [Argument] An *argument* A on the basis of a knowledge base $(\mathcal{K}, \leq')$ in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}, \leq)$ is:

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

2. $A_1, \ldots A_n \rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there exists a strict rule $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow \psi$ in $\mathcal{R}_s$.
   $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$,
   $\texttt{Conc}(A) = \psi$,
   $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$.
   $\texttt{DefRules}(A) = \texttt{DefRules}(A_1) \cup \ldots \cup \texttt{DefRules}(A_n)$,
   $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots \texttt{Conc}(A_n) \rightarrow \psi$

3. $A_1, \ldots A_n \Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there exists a defeasible rule $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \Rightarrow \psi$ in $\mathcal{R}_d$.
   $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$,
   $\texttt{Conc}(A) = \psi$,
   $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$,
   $\texttt{DefRules}(A) = \texttt{DefRules}(A_1) \cup \ldots \cup \texttt{DefRules}(A_n) \cup \{\texttt{Conc}(A_1), \ldots \texttt{Conc}(A_n) \Rightarrow \psi\}$,
   $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots \texttt{Conc}(A_n) \Rightarrow \psi$.

Example 4.1.7 Consider a knowledge base in an argumentation system with

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

An argument for w is displayed in traditional proof-tree format in Figure 4.1, where a single line stands for a strict inference and a double line for a defeasible inference. The type of a premise is indicated with a superscript. Formally the argument and its subarguments are written as follows:

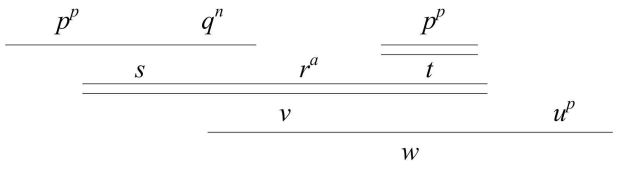


Figure 4.1: An argument

A_1: p $\quad$ A_5: $A_1 \Rightarrow t$
A_2: q $\quad$ A_6: $A_1, A_2 \rightarrow s$
A_3: r $\quad$ A_7: $A_5, A_3, A_6 \Rightarrow v$
A_4: u $\quad$ A_8: $A_7, A_4 \rightarrow w$

We have that

$\texttt{Prem}(A_8) = \{p, q, r, u\}$
$\texttt{Conc}(A_8) = w$
$\texttt{Sub}(A_8) = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\}$
$\texttt{DefRules}(A_8) = \{p \Rightarrow t;\ s, r, t \Rightarrow v\}$
$\texttt{TopRule}(A_8) = v, u \rightarrow w$

The distinction between two kinds of inference rules and three kinds of premises motivates a distinction into four kinds of arguments.

Definition 4.1.8 [Argument properties] An argument A is

- *strict* if $\texttt{DefRules}(A) = \varnothing$;
- *defeasible* if $\texttt{DefRules}(A) \neq \varnothing$;
- *firm* if $\texttt{Prem}(A) \subseteq \mathcal{K}_n$;
- *plausible* if $\texttt{Prem}(A) \not\subseteq \mathcal{K}_n$.

We write $S \vdash \varphi$ if there exists a strict argument for φ with all premises taken from S, and $S \mathrel{|\!\sim} \varphi$ if there exists a defeasible argument for φ with all premises taken from S.

Example 4.1.9 In Example 4.1.7 the argument A_2 is strict and firm, while A_1, A_3, A_4 and A_6 are strict and plausible and A_5, A_7 and A_8 are defeasible and plausible. Furthermore, we have that $\mathcal{K} \vdash p$, $\mathcal{K} \vdash q$, $\mathcal{K} \vdash r$, $\mathcal{K} \vdash u$, $\mathcal{K} \vdash s$ and $\mathcal{K} \mathrel{|\!\sim} t$, $\mathcal{K} \mathrel{|\!\sim} v$, $\mathcal{K} \mathrel{|\!\sim} w$.

4.1.3 Argument orderings

Now that the notion of an argument has been defined, orderings on arguments can be considered. Below $\preceq$ is a partial preorder such that $A \preceq B$ means that B is at least as 'good' as A. As usual $A \prec B$ means $A \preceq B$ and $B \not\preceq A$.

The present framework allows for any partial preorder on arguments that satisfies two basic assumptions.

Definition 4.1.10 [Admissible argument orderings] Let $\mathcal{A}$ be a set of arguments. Then a partial preorder $\preceq$ on A is an *admissible argument ordering* iff

1. if A is firm and strict and B is defeasible or plausible, then $B \prec A$;
2. if $A = A_1, \ldots A_n \rightarrow \psi$ then for all $1 \leq i \leq n, A \preceq A_i$ and for some $1 \leq i \leq n, A_i \preceq A$.

The first condition says that strict-and-firm arguments are stronger than all other arguments, while the second condition says that a strict inference cannot make an argument weaker or stronger.

The notion of an argument ordering is used in the notion of an argument theory, which is the more concrete counterpart of the notion of an abstract argumentation theory of Definition 4.1.30.

Definition 4.1.11 [Argumentation theories] An *argumentation theory* is a triple $AT = (AS, KB, \preceq)$ where AS is an argumentation system, KB is a knowledge base in AS and $\preceq$ is an admissible ordering of the set of all arguments that can be constructed from KB in AS (below called the set of arguments on the basis of AT).

If there is no danger for confusion the argumentation system will below be left implicit.

We next define two argument orderings, called the weakest-link and last-link orderings. Both orderings are defined as a function from the orderings $\leq$ on $\mathcal{R}_d$ and $\leq'$ on $\mathcal{K}/K_n$. To make this work, a way is needed to 'lift' a partial preorder $\leq_e$ of elements of two sets to an ordering of the two sets. The following way results in a strict partial order $\prec_s$ on sets:

- $S_1 \prec_s S_2$ iff there exists an $e_1 \in S_1$ such that for all $e_2 \in S_2$ it holds that $e_1 <_e e_2$.

In words, S_2 is strictly preferred over S_1 if there exists an element of S_1 that is strictly inferior to all elements of S_2.

Now the **last-link principle** prefers an argument A over another argument B if the last defeasible rules used in B are less preferred than the last defeasible rules in A or, in case both arguments are strict, if the premises of B are less preferred than the premises of A. The concept of 'last defeasible rules' is defined as follows.

Definition 4.1.12 [Last defeasible rules] Let A be an argument.

- $\texttt{LastDefRules}(A) = \varnothing$ iff $\texttt{DefRules}(A) = \varnothing$.
- If $A = A_1, \ldots, A_n \Rightarrow \phi$, then $\texttt{LastDefRules}(A) = \{\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \Rightarrow \phi\}$, otherwise $\texttt{LastDefRules}(A) = \texttt{LastDefRules}(A_1) \cup \ldots \cup \texttt{LastDefRules}(A_n)$.

A simple example with more than one last defeasible rule is with $\mathcal{K} = \{p; q\}$ and $\mathcal{R}_d = \{p \Rightarrow r;\ q \Rightarrow s\}$. Then for the argument A for $r \wedge s$ we have that $\texttt{LastDefRules}(A) = \{p \Rightarrow r;\ q \Rightarrow s\}$.

The above definition is now used to compare pairs of arguments as follows:

Definition 4.1.13 [Last link principle] Let A and B be two arguments. Then $A \prec B$ iff either

1. condition (1) of Definition 4.1.10 holds; or
2. $\texttt{LastDefRules}(A) \prec_s \texttt{LastDefRules}(B)$; or

3. $\texttt{LastDefRules}(A)$ and $\texttt{LastDefRules}(B)$ are empty and $\texttt{Prem}(A) \prec_s \texttt{Prem}(B)$.

Consider the following example on whether people misbehaving in a university library may be denied access to the library.[1]

Example 4.1.14 Let $\mathcal{K}_p = \{Snores;\ Professor\}$, $\mathcal{R}_d =$

$\{Snores \Rightarrow_{r_1} Misbehaves;$
$Misbehaves \Rightarrow_{r_2} AccessDenied;$
$Professor \Rightarrow_{r_3} \neg AccessDenied\}$.

Assume that $Snores <' Professor$ and $r_1 < r_2,\ r_1 < r_3,\ r_3 < r_2$, and consider the following arguments.

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

Let us apply the ordering to the arguments A_3 and B_2. The rule sets to be compared are $\texttt{LastDefRules}(A_3) = \{r_2\}$ and $\texttt{LastDefRules}(B_2) = \{r_3\}$. Since $r_3 < r_2$ we have that $B_2 \prec_s A_3$.

The **weakest-link principle** considers not the last but all uncertain elements in an argument. It prefers an argument A over an argument B if A is preferred to B on both their premises and their defeasible rules.

Definition 4.1.15 [Weakest link principle] Let A and B be two arguments. Then $A \prec B$ iff either condition (1) of Definition 4.1.10 holds; or

1. $\texttt{Prem}(A) \prec_s \texttt{Prem}(B)$; and
2. If $\texttt{DefRules}(B) \neq \varnothing$ then $\texttt{DefRules}(A) \prec_s \texttt{DefRules}(B)$.

Example 4.1.16 Consider again Example 4.1.14. With the weakest-link principle the ordering between A_3 and B_2 is different. The rule sets to be compared are now $\texttt{DefRules}(A_3) = \{r_1, r_2\}$ and $\texttt{DefRules}(B_2) = \{r_3\}$. Since $r_1 < r_3$ we have that $\texttt{DefRules}(A_3) \prec_s \texttt{DefRules}(B_2)$. Moreover, since $Snores <' Professor$ we also have that $\texttt{Prem}(A_3) \prec_s \texttt{Prem}(B_2)$.

4.1.4 Attack and defeat

In this section the notion of defeat will be defined, in terms of two more basic components: non-evaluative syntactic notions of attack and the preference relation on arguments. In short, the idea is that defeat is determined by attack plus preference (except in some cases, where attack automatically leads to defeat). As explained in Chapter 1, when arguments are inference trees, then three syntactic forms of attack are possible, namely, attack on a premise, on a conclusion and on an inference. These notions will now be formally defined and then combined with a preference ordering on arguments to yield three kinds of defeat.

[1] In all examples below, sets that are not specified are assumed to be empty.

Attack

First the ways in which arguments can be attacked are defined. Recall that these are just syntactic categories and do not reflect any preference between arguments. The first way of attack corresponds to the case where one argument uses a defeasible rule of which another argument says that it does not apply to the case at hand. Its definition assumes that inference rules can be named in the object language; the precise nature of this naming convention will be left implicit, unless indicated otherwise in examples.

Definition 4.1.17 [Undercutting attack] Argument A *undercuts* argument B (on B') iff $\mathtt{Conc}(A) \in \overline{B'}$ for some $B' \in \mathtt{Sub}(B)$ of the form $B''_1, \ldots, B''_n \Rightarrow \psi$.

Example 4.1.18 In Example 4.1.7 argument A_8 can be undercut in two ways: by an argument with conclusion $\overline{A_5}$, which undercuts A_8 on A_5, and by an argument with conclusion $\overline{A_7}$, which undercuts A_8 on A_7. To illustrate the construction of such arguments, let us name $p \Rightarrow t$ with d_1 and $s,\ r,\ t \Rightarrow w$ with d_2, and extend $\mathcal{K}_p$ with x and R_s with $p,\ x \rightarrow \neg d_1$ and $\mathcal{R}_d$ with $q,\ x \Rightarrow \neg d_2$. Then A_5 is undercut by

A_9: $\quad A_1, x \rightarrow \neg d_1$

while A_7 is undercut by

A_{10}: $\quad A_2, x \Rightarrow \neg d_2$

Undercutting attackers only say that there is some exceptional situation in which a defeasible inference rule cannot be applied, without drawing the opposite conclusion. Rebutting attacks do the latter: they provide a contrary or contradictory conclusion for a defeasible (sub-)conclusion of the attacked argument.

Definition 4.1.19 [rebutting attack] Argument A *rebuts* argument B (on B') iff $\mathtt{Conc}(A) \in \overline{\varphi}$ for some $B' \in \mathtt{Sub}(B)$ of the form $B''_1, \ldots, B''_n \Rightarrow \varphi$. In such a case A *contrary-rebuts* B iff $\mathtt{Conc}(A)$ is a contrary of φ.

Example 4.1.20 In Example 4.1.7 argument A_8 can be rebutted on A_5 with an argument for $\bar{t}$ and on A_7 with an argument for $\bar{v}$. Moreover, if $\bar{t} = -t$ and the rebutter has a defeasible top rule, then A_5 in turn rebuts the argument for $\bar{t}$. However, A_8 itself does not rebut that argument, except in the special case where $w \in \bar{\bar{t}}$. This shows that for three reasons rebutting attack is not symmetric: the rebutting argument can have a strict top rule, rebutting can be contrary-rebutting, and rebutting can be launched on a subargument. However, the present example also shows that in the latter case, if the rebutting attack is not of the contrary-rebutting kind and the rebutter has a defeasible top rule, the directly rebutted subargument in turn rebuts its attacker.

This can be illustrated by extending $\mathcal{K}_p$ again with x and R_d with $p, x \Rightarrow \neg t$. Then the argument

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

rebuts and is rebutted by A_5.

The final way of attack is an attack on a (non-axiom) premise.

Definition 4.1.21 [undermining attack] Argument A *undermines* B iff $\mathtt{Conc}(A) \in \overline{\varphi}$ for some $\varphi \in \mathtt{Prem}(B) \setminus \mathcal{K}_n$. In such a case argument A *contrary-undermines* B iff $\mathtt{Conc}(A)$ is a contrary of φ or if $\varphi \in \mathcal{K}_a$.

Example 4.1.22 In Example 4.1.7 argument A_8 can be undermined with an argument that has conclusion $\overline{p}$, $\overline{r}$ or $\overline{u}$. Note that if the attacker has, say, conclusion $\overline{p}$, has a defeasible top rule and does not contrary-undermine A_8 then p as an argument in turn rebuts the attacker. For example, if $\mathcal{K}_p$ is again extended with x and R_d is extended with $x \Rightarrow \neg p$ then

A_{12}: $\quad x \Rightarrow \neg p$

undermines A_5, A_6, A_7 and A_8 while A_1 in turn rebuts A_{12}.

The following example (based on Example 4 of Caminada and Amgoud 2007) illustrates the interplay between strict and defeasible rules in rebutting attack.

Example 4.1.23 Let $\mathcal{R}_d = \{r_1, r_2\}$ where

$r_1 = \quad WearsRing \Rightarrow Married$
$r_2 = \quad PartyAnimal \Rightarrow Bachelor$

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

$r_3 = \quad Married \rightarrow \neg Bachelor$
$r_4 = \quad Bachelor \rightarrow \neg Married$

and let $\mathcal{K}_p = \{WearsRing, PartyAnimal\}$. Consider the following two arguments.

A_1:	$WearsRing$	B_1:	$PartyAnimal$
A_2:	$A_1 \Rightarrow Married$	B_2:	$B_1 \Rightarrow Bachelor$
A_3:	$A_2 \rightarrow \neg Bachelor$	B_3:	$B_2 \rightarrow \neg Married$

A_3 rebuts B_3 on its subargument B_2 while B_3 rebuts A_3 on its subargument A_2. Note that A_2 does not rebut B_3, since B_3 applies a strict rule; likewise for B_2 and A_3.

Defeat

Now that we know how arguments can be attacked, the argument ordering can be used to define which attacks result in defeat. For undercutting attack no preferences will be needed to make it result in defeat, since otherwise a weaker undercutter and its stronger target might be in the same extension. This would be strange since then the extension contains an argument that applies an inference rule of which another argument in the same extension says that it should not be applied. The same holds for the other two ways of attack as far as they involve contraries (i.e., non-symmetric conflict relations between formulas). The reason for this is that otherwise if a rebutting or undermining attacker is weaker than its target, both may be in the same extension. For the remaining forms of attack the argument ordering will be used to determine whether they result in defeat.

Definition 4.1.24 [Successful rebuttal] Argument A *successfully rebuts* argument B if A rebuts B on B' and either A contrary-rebuts B' or $A \not\prec B'$.

This definition determines whether rebutting attack is successful by comparing the conflicting arguments at the points where they conflict.

Example 4.1.25 Consider again Example 4.1.23. The conflict between A_3 and B_3 is resolved by comparing A_3 with B_2 and comparing B_3 with A_2. Let us apply the last-link ordering. If $r_1 < r_2$ then $B_2 \prec A_3$ so A_3 successfully rebuts B_2 and B_3 while

B_3 does not successfully rebut A_2 or A_3. If, by contrast, $r_1 \not\prec r_2$ and $r_2 \not\prec r_1$ then $A_2 \not\prec B_3$ and $B_2 \not\prec A_3$ so both A_3 and B_3 successfully rebut each other (while A_3 still successfully rebuts B_2 and not vice versa, and likewise for B_3 and A_2).

Example 4.1.23 also illustrates why Definitions 4.1.19 and 4.1.24 should not allow that a defeasible argument with a strict top rule can be (successfully) rebutted on its final conclusion. The reason is that otherwise if all defeasible rules in the example are of equal preference, the set $\{A_1, A_2, B_1, B_2\}$ is admissible, which violates the rationality postulate of indirect consistency (see Section 4.3 below).

Definition 4.1.26 [Successful undermining] Argument A *successfully undermines* B if A undermines B and either A contrary-undermines B or $A \not\prec B$.

This definition exploits that an argument premise is also defined to be a subargument.

In Example 4.1.7 any argument for $\overline{r}$ successfully undermines A_8 since it contrary-undermines it since $r \in \mathcal{K}_a$. The same holds for any argument for a contrary of p or u while for arguments for contradictories of p or u this depends on the argument ordering (which may in turn depend on the ordering $\leq'$ on $\mathcal{K}$; see e.g. Definitions 4.1.13 and 4.1.15 above).

The three defeat relations can now be combined in an overall definition of 'defeat':

Definition 4.1.27 [Defeat] Argument A *defeats* argument B iff A undercuts or successfully rebuts or successfully undermines B. Argument A *strictly defeats* argument B if A defeats B and B does not defeat A.

Example 4.1.28 To further continue Example 4.1.23, if $r_1 < r_2$ we have that A_3 strictly defeats B_3 while if both $r_1 \not< r_2$ and $r_2 \not< r_1$ then A_3 and B_3 defeat each other. Note also that if A_3 is deleted from the example, then if $B_3 \prec A_2$, no argument in the example is defeated. This may at first sight seem counterintuitive but this is due to the fact that the example violates closure of R_s under transposition (cf. Section 4.3 below).

Example 4.1.29 Consider finally again Example 4.1.14. Both A_3 and B_2 rebut each other. With the last-link argument ordering we saw that $B_2 \prec_s A_3$, so A_3 successfully rebuts B_2 while B_2 does not successfully rebut A_3. Hence A_3 strictly defeats B_2. However, with the weakest-link ordering we instead had that $A_3 \prec_s B_2$, so then B_2 strictly defeats A_3.

4.1.5 Linking structured and abstract argumentation

We are now ready to link our structured argumentation theories to Dung-style abstract argumentation theories.

Definition 4.1.30 [Abstract argumentation theory corresponding to an AT] An *abstract argumentation theory AAT_{AT} corresponding to an argumentation theory AT* is a pair $\langle$*Args*, *Defeat*$\rangle$ such that:

- *Args* is the set of arguments on the basis of AT as defined by Definition 4.1.6,
- *Defeat* is the relation on *Args* given by Definition 4.1.27.

It is now also possible to define a consequence notion for well-formed formulas. Several definitions are possible. The following definition directly uses the notions of justified, defensible and overruled arguments from Chapter 2: (here an S-justified (S-defensible, S-overruled) argument is an argument that is justified (defensible, overruled) according to semantics S):

Definition 4.1.31 [The status of conclusions] For any semantics S and for any argumentation theory AT and formula $\varphi \in \mathcal{L}_{AT}$:

1. φ is *S-justified* in AT if and only if there exists an S-justified argument on the basis of AT with conclusion φ;
2. φ is *S-defensible* in AT if and only if φ is not S-justified in AT and there exists an S-defensible argument on the basis of AT with conclusion φ;
3. φ is *S-overruled* in AT if and only if it is not S-justified or S-defensible in AT and there exists an S-overruled argument on the basis of AT with conclusion φ.

Note that the first condition is equivalent to

1. φ is *S-justified* in AT if and only if there exists an argument with conclusion φ that is contained in all S-extensions of AT.

Thus this definition does not allow that different extensions contain different arguments for a skeptical conclusion and therefore does not capture floating conclusions (see Section 2.2). The following alternative definition does capture floating conclusions.

Definition 4.1.32 [Justified conclusions (possibly floating)]

1. φ is *S-f-justified* in AT if and only if all S-extensions of AT contain an argument with conclusion φ.

4.2 Domain-specific vs. general inference rules

The framework defined in this chapter can be used in two ways, depending on whether the inference rules are domain-specific or not. The inference rules of argumentation systems are not part of the logical language $\mathcal{L}$ but are metalevel constructs. The usual practice in standard logic is that inference rules express general patterns of reasoning, such as modus ponens, universal instantiation and so on. By contrast, in nonmonotonic logic often domain-specific inference rules are used, as in default logic. The difference between both approaches is illustrated with the following example. Consider the information that all Frisians are Dutch, that the Dutch are usually tall and that Wiebe is Frisian. With domain-specific inference rules this can in a propositional language be represented as follows:

$$\begin{array}{lcl} \mathcal{R}_s & = & \{Frisian \rightarrow Dutch\} \\ \mathcal{R}_d & = & \{Dutch \Rightarrow Tall\} \\ \mathcal{K}_p & = & \{Frisian\} \end{array}$$

The argument that Wiebe is tall then has the form as displayed on the left in Figure 4.2.

With general inference rules the two rules must instead be represented in the object language $\mathcal{L}$. The first one can be represented with the material implication but for the

second one a connective for defeasible conditionals must be added to $\mathcal{L}$ and a defeasible modus-ponens inference rule must be added for this connective. For example:

$$\begin{array}{lcl} \mathcal{R}_s & = & \{\varphi, \varphi \supset \psi \rightarrow \psi \text{ (for all } \varphi, \psi \in \mathcal{L}), \ldots\} \\ \mathcal{R}_d & = & \{\varphi, \varphi \rightsquigarrow \psi \Rightarrow \psi \text{ (for all } \varphi, \psi \in \mathcal{L}), \ldots\} \\ \mathcal{K}_p & = & \{\mathit{Frisian} \supset \mathit{Dutch}, \mathit{Dutch} \rightsquigarrow \mathit{Tall}, \mathit{Frisian}\} \end{array}$$

Then the argument that Wiebe is tall has the form as displayed on the right in Figure 4.2.

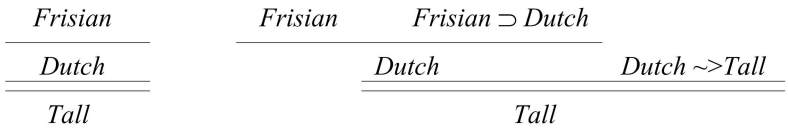


Figure 4.2: Domain-specific vs. general inference rules

If domain-specific defeasible rules are defined over a first-order language, then the same notational naming convention is often used as for defaults in default logic. A rule with free variables is used as a scheme for all its ground instances, that is, for all its instances in which the variable x is replaced by a ground term from $\mathcal{L}$. Moreover, the scheme is often given a name $d(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are all free variables that occur in the scheme. Such a name allows the formulation of undercutters to a rule. Consider, for example:

$d(x)$: $\texttt{Bird}(x) \Rightarrow \texttt{Flies}(x)$

Then schemes for undercutters can be written as follows:

$u(x)$: $\texttt{Penguin}(x) \Rightarrow \neg d(x)$

To see how this naming convention can be used, consider the following knowledge base:

$$\begin{array}{ll} K_n = & \{\forall x(\texttt{Penguin}(x) \supset \texttt{Bird}(x))\} \\ K_p = & \{\texttt{Penguin}(\mathit{Tweety}), \texttt{Bird}(\mathit{Polly})\} \end{array}$$

Then two arguments can be constructed for the conclusions that Tweety and Polly can fly (the strict rules are assumed to be all valid first-order inferences):

$$\begin{array}{ll} A_1\text{:} & \texttt{Penguin}(\mathit{Tweety}) \\ A_2\text{:} & \forall x(\texttt{Penguin}(x) \supset \texttt{Bird}(x)) \\ A_3\text{:} & A_1, A_2 \rightarrow \texttt{Bird}(\mathit{Tweety}) \\ A_4\text{:} & A_3 \Rightarrow \texttt{Flies}(\mathit{Tweety}) \end{array} \qquad \begin{array}{ll} B_1\text{:} & \texttt{Bird}(\mathit{Polly}) \\ B_2\text{:} & B_1 \Rightarrow \texttt{Flies}(\mathit{Polly}) \end{array}$$

However, only for Tweety can an undercutter be constructed:

$$\begin{array}{ll} C_1\text{:} & \texttt{Penguin}(\mathit{Tweety}) \\ C_2\text{:} & C_1 \Rightarrow \neg d(\mathit{Tweety}) \end{array}$$

The point is that $d(x)$ is not a rule name but a rule name scheme, and only for its instance $d_1(\mathit{Tweety})$ can an undercutter be constructed. If, by contrast, the birds-fly rule had been named with d, then applying the undercutter for Tweety would also block the default for Polly, which is clearly undesirable.

Both Vreeswijk (1993; 1997) and Pollock (1987)−(1994) intended their inference rules to express general patterns of reasoning, which is much more in line with the role of inference rules in standard logic. Indeed, an important part of John Pollock's work was the study of general patterns of (epistemic) defeasible reasoning, which he called prima facie reasons. He formalised prima facie reasons for reasoning patterns involving perception, memory, induction, temporal persistence and the statistical syllogism, as well as undercutters for these reasons. In the present framework such prima facie reasons can be expressed as defeasible inference schemes.

For example, the principles of perception and memory could be written as follows:

$d_p(x, \varphi)$: $\quad \texttt{Sees}(x, \varphi) \Rightarrow \varphi$
$d_m(x, \varphi)$: $\quad \texttt{Recalls}(x, \varphi) \Rightarrow \varphi$

(Note that these schemes assume a naming convention for formulas in a first-order language, since φ is a term in the antecedent while it is a well-formed formula in the consequent.) Then undercutters for d_p state circumstances in which perceptions are unreliable, while undercutters of d_m state conditions under which memories may be flawed. For example, a well-known cause of false memories of events is that the memory is distorted by, for instance, hearing, reading, or watching a TV programme about the remembered event. A general undercutter for distorted memories could be

$u_m(x, \varphi)$: $\quad \texttt{DistortedMemory}(x, \varphi) \Rightarrow \neg d_m(x, \varphi)$

combined with information such as

$\forall x, \varphi(\texttt{ReadsAbout}(x, \varphi) \rightsquigarrow \texttt{DistortedMemory}(x, \varphi))$

Pollock also formulated a prima facie reason for temporal persistence. Using reification and the `Holds` predicate it can be written as follows:[2]

$d_t(\varphi, t_1, t_2)$: $\quad \texttt{Holds}(\varphi, t_1), t_1 < t_2 \Rightarrow \texttt{Holds}(\varphi, t_2)$

Pollock defines the following undercutter for this scheme.

$u_t(\varphi, t_1, t_2, t_3)$: $\quad \neg\texttt{Holds}(\varphi, t_3), t_1 < t_3 < t_2 \Rightarrow \neg d_t(\varphi, t_1, t_2)$

Pollock's prima facie reasons are in fact a subspecies of argument schemes. The notion of an argument scheme was developed in philosophy and is currently is an important topic in the computational study of argumentation (cf. Walton *et al.* 2008). Argument schemes are stereotypical non-deductive patterns of reasoning, consisting of a set of premises and a conclusion that is presumed to follow from them. Uses of argument schemes are evaluated in terms of critical questions specific to the scheme. An example of an epistemic argument scheme is the scheme from expert opinion (Walton *et al.*, 2008, p. 310):

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

P is true

This scheme has six critical questions:

[2] Pollock actually restricts this principle to certain kinds of propositions, to avoid some counterintuitive consequences.

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?

A natural way to formalise reasoning with argument schemes is to regard them as defeasible inference rules and to regard critical questions as pointers to counterarguments. More precisely, the three kinds of attack on arguments correspond to three kinds of critical questions of argument schemes. Some critical questions challenge an argument's premise and therefore point to undermining attacks, others point to undercutting attacks, while again other questions point to rebutting attacks. In the scheme from expert opinion questions (2) and (3) point to underminers (of, respectively, the first and second premise), questions (4), (1) and (6) point to undercutters (the exceptions that the expert is biased or incredible for other reasons and that he makes scientifically unfounded statements) while question (5) points to rebutting applications of the expert opinion scheme.

4.3 Rationality postulates

Dung's semantics can be seen as rationality constraints on evaluating arguments in abstract argumentation frameworks. The refinement of his abstract approach with structured arguments naturally leads to the question whether this additional structure gives rise to additional rationality constraints. Caminada and Amgoud (2007) gave a positive answer to this question by proposing a number of 'rationality postulates' for what they called 'rule-based argumentation'. These postulates formulate constraints on any extension of an argumentation framework corresponding to an argumentation theory:

- **Closure under subarguments:** for every argument in an extension also all its subarguments are in the extension.
- **Closure under strict rules:** the set of conclusions of all arguments in an extension is closed under strict-rule application.
- **Direct consistency:** the set of conclusions of all arguments in an extension is consistent.
- **Indirect consistency:** the closure of the set of conclusions of all arguments in an extension under strict-rule application is consistent.

Before it can be studied to what extent the present framework satisfies these rationality postulates, first some some technicalities concerning strict inference rules must be discussed. To start with, Caminada and Amgoud define the notions of a transposition of a strict rule and closure of sets of strict rules under transposition.

Definition 4.3.1 [Transposition] A strict rule s is a *transposition* of $\varphi_1, \ldots, \varphi_n \rightarrow \psi$ iff $s = \varphi_1, \ldots, \varphi_{i-1}, -\psi, \varphi_{i+1}, \ldots, \varphi_n \rightarrow -\varphi_i$ for some $1 \leq i \leq n$.

Definition 4.3.2 [Transposition operator] Let $\mathcal{R}_s$ be a set of strict rules. $Cl_{tp}(\mathcal{R}_s)$ is the smallest set such that:

- $\mathcal{R}_s \subseteq Cl_{tp}(\mathcal{R}_s)$, and
- If $s \in Cl_{tp}(\mathcal{R}_s)$ and t is a transposition of s then $t \in Cl_{tp}(\mathcal{R}_s)$.

We say that $\mathcal{R}_s$ is *closed under transposition* iff $Cl_{tp}(\mathcal{R}_s) = \mathcal{R}_s$.

Now the subclass of argumentation systems closed under transposition can be defined.

Definition 4.3.3 [Closure under transposition] An argumentation system $(\mathcal{L}, ^-, \mathcal{R}, \leq)$ is *closed under transposition* if $\mathcal{R}_s = Cl_{tp}(\mathcal{R}_s)$. An argumentation theory is closed under transposition if its argumentation system is.

Caminada and Amgoud (2007) also define the closure of a set of formulas under application of strict rules.

Definition 4.3.4 [Closure of a set of formulas] Let $\mathcal{P} \subseteq \mathcal{L}$. The *closure* of $\mathcal{P}$ under the set $\mathcal{R}_s$ of strict rules, denoted $Cl_{\mathcal{R}s}(\mathcal{P})$, is the smallest set such that:

- $\mathcal{P} \subseteq Cl_{\mathcal{R}s}(\mathcal{P})$.
- if $\varphi_1, \ldots, \varphi_n \rightarrow \psi \ \in \ \mathcal{R}_s$ and $\varphi_1, \ldots, \varphi_n \in Cl_{\mathcal{R}s}(\mathcal{P})$ then $\psi \in Cl_{\mathcal{R}s}(\mathcal{P})$.

If $\mathcal{P} = Cl_{\mathcal{R}s}(\mathcal{P})$, then $\mathcal{P}$ is said to be *closed*.

Below it will also be relevant whether the notion $\vdash$ induced by an argumentation system satisfies contraposition.

Definition 4.3.5 [Closure under contraposition] An argumentation system is *closed under contraposition* if for all $S \subseteq \mathcal{L}$, all $s \in S$ and all φ it holds that if $S \vdash \varphi$ then $S \setminus \{s\} \cup \{-\varphi\} \vdash -s$. An argumentation theory is closed under contraposition if its argumentation system is.

Now the first two postulates, closure under subarguments and under strict rules, hold unconditionally for the present framework.

Proposition 4.3.6 Let $<\mathcal{A}, \mathit{defeat}>$ be an abstract argumentation theory as defined in Definition 4.1.30 and E any of its grounded, preferred or stable extensions. Then for all $A \in E$: if $A' \in \mathtt{Sub}(A)$ then $A' \in E$.

Proposition 4.3.7 Let $<\mathcal{A}, \mathit{defeat}>$ be an abstract argumentation theory corresponding to an argumentation theory, and E any of its grounded, preferred or stable extensions. Then $\{\mathtt{Conc}(A) | A \in E\} = Cl_{\mathcal{R}s}(\{\mathtt{Conc}(A) | A \in E\})$.

The two consistency postulates do not hold in general. However, when the weakest- or last-link argument ordering are used, then they do hold for systems that are closed under transposition or contraposition, of which the strict closure of the axioms $\mathcal{K}_n$ is consistent, and that are 'well-formed' in that they respect the intended use of assumptions and contraries:

Definition 4.3.8 An argumentation theory is *well-formed* if:

1. no consequent of a defeasible rule is a contrary of the consequent of a strict rule;

2. if $\varphi \in \mathcal{K}_a$ and φ is a contrary of ψ, then $\psi \notin \mathcal{K}_n \cup K_p$ and ψ is not the conclusion of a rule in $\mathcal{R}$.

Condition (2) in effect says that assumptions can only be contraries of other assumptions. An example of a AT that is not well-formed is

$\mathcal{R}_s = \{p \rightarrow q\}, \mathcal{R}_d = \{r \Rightarrow s, t \Rightarrow u\}$
$\mathcal{K}_p = \{p, r\}, \mathcal{K}_a = \{v\}$

and such that s is a contrary of q and v is a contrary of u. Then condition (1) of Definition 4.3.8 is violated since we have arguments A: $p \rightarrow q$ and B: $r \Rightarrow s$. Moreover, condition (2) is violated since $v \in \mathcal{K}_a$ and $t \Rightarrow u \in R_d$.

Theorem 4.3.9 Let $<\mathcal{A}$, *defeat*$>$ be an abstract argumentation theory corresponding to a well-formed argumentation theory that is closed under contraposition or transposition and has a weakest- or last-link argument ordering and a consistent $Cl_{\mathcal{R}s}(\mathcal{K}_n)$, and let E be any of its grounded, preferred or stable extensions. Then the set $\{\texttt{Conc}(A) \mid A \in E\}$ is consistent.

Theorem 4.3.10 Let $<\mathcal{A}$, *defeat*$>$ be an abstract argumentation theory corresponding to a well-formed argumentation theory that is closed under contraposition or transposition and has a weakest- or last-link argument ordering and a consistent $Cl_{\mathcal{R}s}(\mathcal{K}_n)$, and let E be any of its grounded, preferred or stable extensions. Then the set $Cl_{\mathcal{R}s}(\{\texttt{Conc}(A) \mid A \in E\})$ is consistent.

Corollary 4.3.11 If the conditions of Theorem 4.3.10 are satisfied, then for any grounded, preferred or stable extension E the set $\{\varphi \mid \varphi$ is a premise of an argument in $E\}$ is consistent.

4.4 Self-defeat

In Chapter 2, Section 2.2 we said that a proper analysis of self-defeating arguments must make the structure of arguments explicit. Now that we have done so, we can explain why this is needed. In the present framework two types of self-defeating arguments are possible: *serial self-defeat* occurs when an argument defeats one if its earlier steps, while *parallel self-defeat* occurs when the contradictory conclusions of two or more arguments are taken as the premises for $\bot$. It turns out that parallel self-defeating can cause problems if argumentation systems are not carefully defined, particularly if they include standard propositional logic.

The following example why serial self-defeat does not cause problems.

Example 4.4.1 Consider the following version of the argument scheme from witness testimony plus undercutter in case the witness is incredible:

$d_w(x, \varphi)$: $\texttt{Says}(x, \varphi) \Rightarrow \varphi$
$u_w(x, \varphi)$: $\texttt{Incredible}(x) \rightarrow \neg d_w(x, \varphi)$

Now suppose that $\mathcal{K}_p$ contains $\texttt{Says}(John,$ "$\texttt{Incredible}(John)$"$)$. Then we have

A_1: $\texttt{Says}(John,$ "$\texttt{Incredible}(John)$"**)**
A_2: $A_1 \Rightarrow \texttt{Incredible}(John)$
A_3: $A_2 \rightarrow \neg d_w(John,$ "$\texttt{Incredible}(John)$"$)$

Argument A_3 is self-defeating since it undercuts itself on A_2. In both preferred and grounded semantics there is a unique extension $E = \{A_1\}$. Arguably this is the desired outcome, since suppose witness John also says something completely unrelated, say, 'the suspect stabbed the victim with a knife' if the self-defeating argument A_3 were overruled, the argument that can be constructed for 'the suspect stabbed the victim with a knife' would be justified since all its defeaters are overruled, while yet it is based on a statement of a witness who says of himself that he is incredible.

The following abstract example illustrates the problems that can be caused by parallel self-defeat.

Example 4.4.2 Let $\mathcal{R}_d = \{p \Rightarrow q;\ r \Rightarrow \neg q;\ t \Rightarrow s\}$ and $\mathcal{K} = \{p, r, t\}$ while $\mathcal{R}_s$ contains all propositionally valid inferences. Then:

A_1: p	A_2: $A_1 \Rightarrow q$
B_1: r	B_2: $B_1 \Rightarrow \neg q$
C_1: $A_2, B_2 \rightarrow \bot$	C_2: $C_1 \rightarrow \neg s$
D_1: t	D_2: $D_1 \Rightarrow s$

Here a problem arises since s can be any formula, so any defeasible argument unrelated to A_2 or B_2, such as D_2, can, depending on the argument ordering, be rebutted by C_2. Clearly, this is extremely harmful, since the existence of just a single case of mutual rebutting defeat, which is very common, could trivialise the system. In fact, of the semantics defined by Dung (1995) this is only a problem for grounded semantics. Since all preferred/stable extensions contain either A_2 or B_2, argument C_2 is not in any of these extensions so D_2 is in these extensions. However, if neither of A_2 and B_2 strictly defeats the other, then neither of them is in the grounded extension so that extension does not defend D_2 against C_2 and therefore does not contain D_2.

According to Martin Caminada (personal communication) this problem can only be solved by making parallel self-defeat impossible. Since these problems only arise in particular argumentation systems and with particular semantics, no general solution will be pursued here; instead such solutions should be provided in instantiations of the framework.

In conclusion, there are good reasons to believe that the two types of self-defeating arguments should be treated differently: while arguments based on parallel self-defeat should always be overruled, arguments with serial self-defeat should retain their force to prevent other arguments from being justified or defensible.

4.5 Exercises

EXERCISE 4.5.1 Consider an argumentation system in which $\mathcal{R}_s$ contains all valid propositional and first-order inferences and with as knowledge base

$\mathcal{K}_n = \{\forall x(Px \supset Qx)\}$
$\mathcal{K}_p = \{Pa, \forall x(Qx \supset Rx)\}, \mathcal{K}_a = \varnothing$

1. Construct an argument A for Ra.
2. Identify $\texttt{Prem}(A)$, $\texttt{Conc}(A)$, $\texttt{Sub}(A)$, $\texttt{DefRules}(A)$ and $\texttt{TopRule}(A)$.
3. What is in terms of Definition 4.1.8 the type of this argument?

EXERCISE 4.5.2 Consider the following argumentation theory with:

$\mathcal{R}_s$ consists of all valid inferences of propositional logic;
$R_d = \{$
$\quad p, q \Rightarrow r,$
$\quad r \vee s \Rightarrow t,$
$\quad u \Rightarrow v,$
$\quad w \Rightarrow \neg u\}$
$\mathcal{K}_n = \{\neg(q \wedge v)\}$
$\mathcal{K}_p = \{p, u, w\}$
$\mathcal{K}_a = \{q\}$

Verify the status of t according to grounded semantics, assuming the weakest-link ordering on arguments.

EXERCISE 4.5.3 Consider the following argumentation theory with:

$\mathcal{R}_s = \{p, q \rightarrow r,\ t \rightarrow \neg d_1\},$
$R_d = \{$
$\quad d_1\colon p \Rightarrow q,$
$\quad d_2\colon s \Rightarrow t,$
$\quad d_3\colon u \Rightarrow v,$
$\quad d_4\colon v \Rightarrow \neg t\}$
$\mathcal{K}_p = \{p, s, u\}$

Where $u <' s$, $d_2 < d_4$ and $d_3 < d_2$.

1. Verify the status of r according to preferred semantics, assuming the weakest-link ordering on arguments.

2. Answer the same question assuming the last-link ordering on arguments.

EXERCISE 4.5.4 Consider the following argumentation theory with:

$\mathcal{R}_s$ consists of all valid propositional inferences,
$\quad R_d = \{$
$\quad d_1\colon p \Rightarrow q,$
$\quad d_2\colon p, q \Rightarrow r,$
$\quad d_3\colon s \Rightarrow t\}$
$\mathcal{K}_p = \{p, s, (q \wedge r) \supset \neg t\}$

Where $d_3 < d_1$ and $d_2 < d_3$. Verify the status of t according to preferred semantics, assuming the last-link ordering on arguments.

EXERCISE 4.5.5 Consider the following default theory $\Delta = (W, D)$:

$$W = \{p\}$$

$$D = \left\{\frac{p : q}{r}, \frac{r : s}{t}, \frac{p : u}{\neg s}\right\}$$

1. Is t in some or all extensions of Δ?
2. Translate $\Delta = (W, D)$ into an argumentation theory AT with domain-specific defeasible inference rules (hint: use assumption-type premises).
3. What is the status of t in AT if $S =$ stable semantics?

EXERCISE 4.5.6 Consider Example 4.4.1.

1. Explain why $E = \{A_1\}$ is the only grounded and preferred extension.
2. Extend the example with the argument based on John's testimony about the suspect and verify its status in grounded and preferred semantics.

EXERCISE 4.5.7 Consider the following, equally strong defaults

1. Persons born in The Netherlands are typically Dutch.
2. Persons with a Norwegian name are typically Norwegian.
3. Persons who are Dutch or Norwegian typically like ice skating.

and the following facts:

4. Brigt Rykkje was born in the Netherlands
5. Brigt Rykkje has a Norwegian name.
6. Nobody is both Dutch and Norwegian.

1. Translate this information into an argumentation theory of which $\mathcal{R}_s$ consists of all valid propositional and first-order inferences and $\mathcal{R}_d$ consists of the defeasible inference scheme for $\rightsquigarrow$ from Section 4.2.
2. Assume that the argument ordering is determined by the last-link principle. We want to know whether Brigt Rykkje likes ice skating. Construct all arguments that are relevant for this proposition and determine whether the conclusion that Brigt Rykkje likes ice skating is justified in grounded semantics.
3. Answer the same question for preferred semantics.
4. Answer the same question for f-justification in preferred semantics.

EXERCISE 4.5.8 Formalise the example of Exercise 2.6.13 as an argumentation theory with domain-specific defaults in a way that satisfies your intuitions about this example.

EXERCISE 4.5.9 Let $\mathcal{R}_s = \{p \rightarrow q; p \rightarrow r; p, r \rightarrow s\}$ and let $^-$ correspond to classical negation.

1. Determine $Cl_{tp}(R_s)$.
2. Determine whether with $Cl_{tp}(R_s)$ it holds that $\{p\} \vdash s$.
3. Determine whether with $Cl_{tp}(R_s)$ it holds that $\{-s\} \vdash -p$.

EXERCISE 4.5.10 Let $\mathcal{R}_s = \{p \rightarrow q;\ \neg q \rightarrow r;\ r \rightarrow \neg p;\ \neg r \rightarrow q;\ p \rightarrow \neg r\}$ and let $^-$ correspond to classical negation.

1. Is $\mathcal{R}_s$ closed under transposition?
2. Does $\mathcal{R}_s$ satisfy contraposition?

EXERCISE 4.5.11 Consider the argumentation theory of Example 4.4.2.

1. Verify the status of argument D_2 for s in grounded semantics.
2. Verify the status of argument D_2 for s in preferred semantics.

Bibliography

Amgoud, L., Bodenstaff, L., Caminada, M., McBurney, P., Parsons, S., Prakken, H., van Veenen, J., and Vreeswijk, G. (2006), "Final review and report on formal argumentation system," Deliverable D2.6, ASPIC IST-FP6-002307.

Antoniou, G., *Nonmonotonic Reasoning*, Cambridge, MA: MIT Press (1997).

Bondarenko, A., Dung, P., Kowalski, R., and Toni, F. (1997), "An abstract, argumentation-theoretic approach to default reasoning," *Artificial Intelligence*, 93, 63–101.

Caminada, M. (2006), "On the issue of reinstatement in argumentation," in *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, no. 4160 in Springer Lecture Notes in AI, Berlin: Springer Verlag, pp. 111–123.

Caminada, M., and Amgoud, L. (2007), "On the evaluation of argumentation formalisms," *Artificial Intelligence*, 171, 286–310.

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

Loui, R. (1987), "Defeat among arguments: a system of defeasible inference," *Computational Intelligence*, 2, 100–106.

McCarthy, J. (1980), "Circumscription - a form of non-monotonic reasoning," *Artificial Intelligence*, 13, 27–39.

Pollock, J., *Knowledge and Justification*, Princeton: Princeton University Press (1974).

Pollock, J. (1987), "Defeasible reasoning," *Cognitive Science*, 11, 481–518.

Pollock, J. (1994), "Justification and Defeat," *Artificial Intelligence*, 67, 377–408.

Prakken, H. (2006), "Formal systems for persuasion dialogue," *The Knowledge Engineering Review*, 21, 163–188.

Prakken, H. (2010), "An abstract framework for argumentation with structured arguments," *Argument and Computation*, 1.

Prakken, H., "Reader Commonsense Reasoning," Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands (2010).

Prakken, H., and Sartor, G. (1997), "Argument-based extended logic programming with defeasible priorities," *Journal of Applied Non-classical Logics*, 7, 25–75.

Prakken, H., and Vreeswijk, G. (2002), "Logics for defeasible argumentation," in *Handbook of Philosophical Logic* (Vol. 4, Second ed.), eds. D. Gabbay and F. Günthner, Dordrecht/Boston/London: Kluwer Academic Publishers, pp. 219–318.

Rahwan, I., and Simari, G. (eds.) *Argumentation in Artificial Intelligence*, Berlin: Springer (2009).

Reiter, R. (1980), "A logic for default reasoning," *Artificial Intelligence*, 13, 81–132.

Rescher, N., *Dialectics: a Controversy-oriented Approach to the Theory of Knowledge*, Albany, N.Y.: State University of New York Press (1977).

Toulmin, S., *The Uses of Argument*, Cambridge: Cambridge University Press (1958).

Vreeswijk, G. (1993), "Defeasible dialectics: a controversy-oriented approach towards defeasible argumentation," *Journal of Logic and Computation*, 3, 317–334.

Vreeswijk, G., *Studies in Defeasible Argumentation*, Doctoral dissertation Free University Amsterdam (1993).

Vreeswijk, G. (1997), "Abstract argumentation systems," *Artificial Intelligence*, 90, 225–279.

Vreeswijk, G., and Prakken, H. (2000), "Credulous and sceptical argument games for preferred semantics," in *Proceedings of the 7th European Workshop on Logics in Artificial Intelligence (JELIA'2000)*, no. 1919 in Springer Lecture Notes in AI, Berlin: Springer Verlag, pp. 239–253.

Walton, D., *Argumentation Schemes for Presumptive Reasoning*, Mahwah, NJ: Lawrence Erlbaum Associates (1996).

Walton, D., Reed, C., and Macagno, F., *Argumentation Schemes*, Cambridge: Cambridge University Press (2008).