CONTENTS

HENRY PRAKKEN & GERARD VREESWIJK

LOGICS FOR DEFEASIBLE ARGUMENTATION

1 INTRODUCTION

Logic is the science that deals with the formal principles and criteria of validity of patterns of inference. This chapter surveys logics for a particular group of patterns of inference, namely those where arguments for and against a certain claim are produced and evaluated, to test the tenability of the claim. Such reasoning processes are usually analysed under the common term 'defeasible argumentation'. We shall illustrate this form of reasoning with a dispute between two persons, A and B. They disagree on whether it is morally acceptable for a newspaper to publish a certain piece of information concerning a politician's private life.[1] 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 moral principle 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(4) Information concerning a person's private life may not be published if that person does not agree with publication.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, page 2). 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 his 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 the moral principle 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. 2, page 3). A agrees with (5–7) and even accepts (8) as a moral principle, but A does not give up his initial claim. (It is assumed that A and B are both male.) Instead he tries to defend it by arguing that he has the stronger argument: he does so by arguing that in this case

[1] Adapted from [Sartor, 1994].

(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 principle (4) has priority over principle (8).

Thus it can be derived that the principle used in A's first argument is stronger than the principle used by B (Fig. 3, page 4), 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.

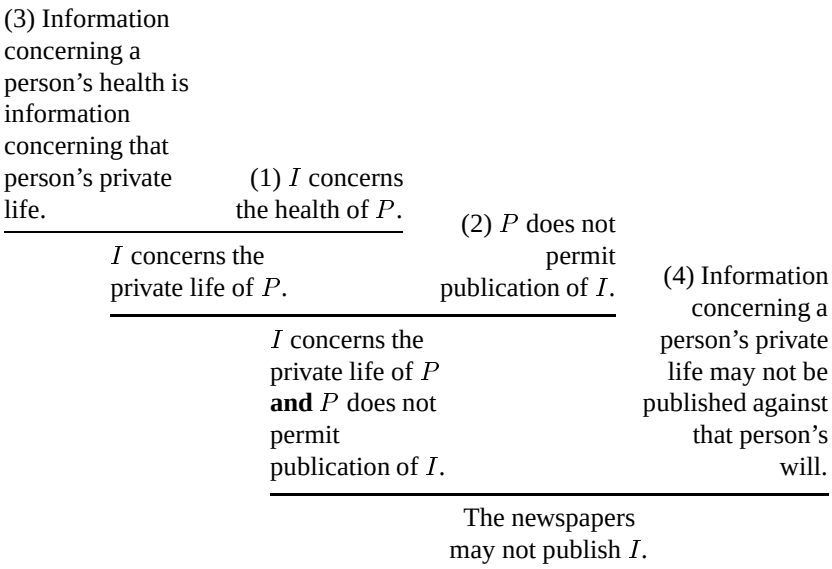


Figure 1. A's argument.

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 moral principles, 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 moral claim as warranted.

This chapter is about logical systems that formalise this kind of reasoning. We

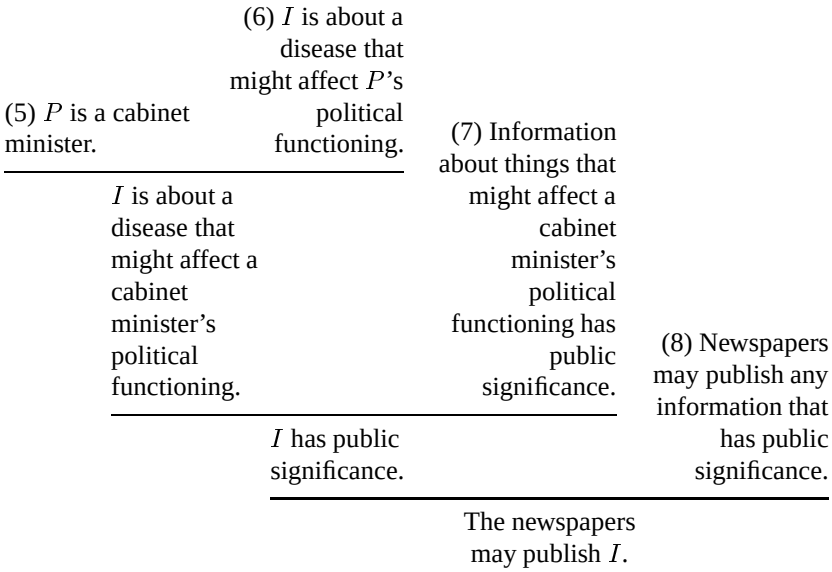


Figure 2. B's argument.

shall call them 'logics for defeasible argumentation', or 'argumentation systems'. As the example shows, these systems lack one feature of 'standard', deductive logic (say, first-order predicate logic, FOL). The notion of 'warrant' that we used in explaining the example is clearly not the same as first-order logical consequence, which has the property of monotonicity: in FOL any conclusion that can be drawn from a given set of premises, remains valid if we add new premises to this set. So 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

(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 principle (4) has priority over principle (8).

Principle (4) has priority over principle (8).

Figure 3. A's priority argument.

not warranted on the basis of (1–8).

Such nonmonotonic consequence notions have been studied over the last twenty years in an area of artificial intelligence called 'nonmonotonic reasoning' (recently the term 'defeasible reasoning' has also become popular), and logics for defeasible argumentation are largely a result of this development. Some might say that the lack of the property of monotonicity disqualifies these notions from being notions of logical consequence: isn't the very idea of calling an inference 'logical' that it is (given the premises) beyond any doubt? We are not so sure. Our view on logic is that it studies criteria of warrant, that is, criteria that determine the degree according to which it is reasonable to accept logical conclusions, even though some of these conclusions are established non-deductively: sometimes it is reasonable to accept a conclusion of an argument even though this argument is not strong enough to establish its conclusion with absolute certainty.

Several ways to formalise nonmonotonic, or defeasible reasoning have been studied. This chapter is not meant to survey all of them but only discusses the argument-based approach, which defines notions like argument, counterargument, attack and defeat, and defines consequence notions in terms of the interaction of arguments for and against certain conclusions. This approach was initiated by the philosopher John Pollock [1987], based on his earlier work in epistemology, e.g. [1974], and the computer scientist Ronald Loui [1987]. As we shall see, argumentation systems are able to incorporate the traditional, monotonic notions of logical consequence as a special case, for instance, in their definition of what an argument is.

The field of defeasible argumentation is relatively young, and researchers disagree on many issues, while the formal meta-theory is still in its early stages. Yet we think that the field has sufficiently matured to devote a handbook survey to it[2]. We aim to show that there are also many similarities and connections between the various systems, and that many differences are variations on a few basic notions, or are caused by different focus or different levels of abstraction. Moreover, we shall show that some recent developments pave the way for a more elaborate meta-theory of defeasible argumentation.

[2]For a survey of this topic from a computer science perspective, see [Chesñevar *et al.*, 1999].

Although when discussing individual systems we aim to be as formal as possible, when comparing them we shall mostly use conceptual or quasi-formal terms. We shall also report on some formal results on this comparison, but it is not our aim to present new technical results; this we regard as a task for further research in the field.

The structure of this chapter is as follows. In Section 2 we give an overview of the main approaches in nonmonotonic reasoning, and argue why the study of this kind of reasoning is relevant not only for artificial intelligence but also for philosophy. In Section 3 we give a brief conceptual sketch of logics for defeasible argumentation, and we argue that it is not obvious that they need a model-theoretic semantics. In Section 4 we become formal, studying how semantic consequence notions for argumentation systems can be defined given a set of arguments ordered by a defeat relation. This discussion is still abstract, leaving the structure of arguments and the origin of the defeat relation largely unspecified. In Section 5 we become more concrete, in discussing particular logics for defeasible argumentation. Then in Section 6 we discuss one way in which argumentation systems can be formulated, viz. in the form of rules for dispute. We end this chapter in Section 7 with some concluding remarks, and with a list of the main open issues in the field.

2 NONMONOTONIC LOGICS: OVERVIEW AND PHILOSOPHICAL RELEVANCE

Before discussing argumentation systems, we place them in the context of the study of nonmonotonic reasoning, and discuss why this study deserves a place in philosophical logic.

2.1 Research in nonmonotonic reasoning

Although this chapter is not about nonmonotonic logics in general, it is still useful to give a brief impression of this field, to put systems for defeasible argumentation in context. Several styles of nonmonotonic logics exist. Most of them take as the basic ‘nonstandard’ unit the notion of a default, or defeasible conditional or rule: this is a conditional that can be qualified with phrases like ‘typically’, ‘normally’ or ‘unless shown otherwise’ (the two principles in our example may be regarded as defaults). Defaults do not guarantee that their consequent holds whenever their antecedent holds; instead they allow us in such cases to defeasibly derive their consequent, i.e., if nothing is known about exceptional circumstances. Most nonmonotonic logics aim to formalise this phenomenon of ‘default reasoning’, but they do so in different ways.

Firstly, they differ in whether the above qualifications are regarded as extra conditions in the antecedent of a default, as aspects of the use of a default, or as inherent in the meaning of a defeasible conditional operator. In addition, within

each of these views on defaults, nonmonotonic logics differ in the technical means by which they formalise it. Let us briefly review the main approaches. (More detailed overviews can be found in e.g. [Brewka, 1991] and [Gabbay *et al.*, 1994].)

Preferential entailment

Preferential entailment, e.g. [Shoham, 1988], is a model-theoretic approach based on standard first-order logic, which weakens the standard notion of entailment. The idea is that instead of checking *all* models of the premises to see if the conclusion holds, only some of the models are checked, viz. those in which as few exceptions to the defaults hold as possible. This technique is usually combined with the 'extra condition' view on defaults, by adding a special 'normality condition' to their antecedent, as in

(1) $\forall x.\mathtt{Birds}(x) \wedge \neg \mathtt{ab}_1(x) \supset \mathtt{Canfly}(x)$

Informally, this reads as 'Birds can fly, unless they are abnormal with respect to flying'. Let us now also assume that Tweety is a bird:

(2) $\mathtt{Bird}(Tweety)$

We want to infer from (1) and (2) that $\mathtt{Canfly}(Tweety)$, since there is no reason to believe that $\mathtt{ab}_1(Tweety)$. This inference is formalised by only looking at those models of (1,2) where the extension of the $\mathtt{ab}_i$ predicates are minimal (with respect to set inclusion). Thus, since on the basis of (1) and (2) nothing is known about whether Tweety is an abnormal bird, there are both FOL-models of these premises where $\mathtt{ab}_1(Tweety)$ is satisfied and FOL-models where this is not satisfied. The idea is then that we can disregard the models satisfying $\mathtt{ab}_1(Tweety)$, and only look at the models satisfying $\neg\ \mathtt{ab}_1(Tweety)$; clearly in all those models $\mathtt{Canfly}(Tweety)$ holds.

The defeasibility of this inference can be shown by adding $\mathtt{ab}_1(Tweety)$ to the premises. Then all models of the premises satisfy $\mathtt{ab}_1(Tweety)$, and the preferred models are now those in which the extension of $\mathtt{ab}_1$ is $\{Tweety\}$. Some of those models satisfy $\mathtt{Canfly}(Tweety)$ but others satisfy $\neg\mathtt{Canfly}(Tweety)$, so we cannot any more draw the conclusion $\mathtt{Canfly}(Tweety)$.

A variant of this approach is Poole's [1988] 'abductive framework for default reasoning'. Poole also represents defaults with normality conditions, but he does not define a new semantics. Instead, he recommends a new way of using first-order logic, viz. for constructing 'extensions' of a theory. Essentially, extensions can be formed by adding as many normality statements to a theory as is consistently possible. The standard first-order models of a theory extension correspond to the preferred models of the original theory.

Intensional semantics for defaults

There are also intensional approaches to the semantics of defaults, e.g. [Delgrande, 1988, Asher & Morreau, 1990]. The idea is to interpret defaults in a possible-

worlds semantics, and to evaluate their truth in a model by focusing on a subset of the set of possible worlds within a model. This is similar to the focusing on certain models of a theory in preferential entailment. On the other hand, intensional semantics capture the defeasibility of defaults not with extra normality conditions, but in the meaning of the conditional operator. This development draws its inspiration from the similarity semantics for counterfactuals in conditional logics, e.g. [Lewis, 1973]. In these logics a counterfactual conditional is interpreted as follows: $\varphi \Rightarrow \psi$ is true just in case ψ is true in a subset of the possible worlds in which φ is true, viz. in the possible worlds which resemble the actual world as much as possible, given that in them φ holds. Now with respect to defeasible conditionals the idea is to define in a similar way a possible-worlds semantics for defeasible conditionals. A defeasible conditional $\varphi \Rightarrow \psi$ is roughly interpreted as 'in all most normal worlds in which φ holds, ψ holds as well'. Obviously, if read in this way, then modus ponens is not valid for such conditionals, since even if φ holds in the actual world, the actual world need not be a normal world. This is different for counterfactual conditionals, where the actual world is always among the worlds most similar to itself. This difference makes that intensional defeasible logics need a component that is absent in counterfactual logics, and which is similar to the selection of the 'most normal' models in preferential entailment: in order to derive default conclusions from defeasible conditionals, the actual world is assumed to be as normal as possible given the premises. It is this assumption that makes the resulting conclusions defeasible: it validates modus ponens for those defaults for which there is no evidence of exceptions.

Consistency and non-provability statements

Yet another approach is to somehow make the expression possible of consistency or non-provability statements. This is, for instance, the idea behind Reiter's [1980] *default logic*, which extends first-order logic with constructs that technically play the role of inference rules, but that express domain-specific generalisations instead of logical inference principles. In default logic, the Tweety default can be written as follows.

$$\mathtt{Bird}(x) : \mathtt{Canfly}(x) / \mathtt{Canfly}(x)$$

The middle part of this 'default' can be used to express consistency statements. Informally the default reads as 'If it is provable that Tweety is a bird, and it is not provable that Tweety cannot fly, then we may infer that Tweety can fly'. To see how this works, assume that in addition to this default we have a first-order theory

$$W = \{\mathtt{Bird}(Tweety), \forall x.\mathtt{Penguin}(x) \supset \neg\mathtt{Canfly}(x)\}$$

Then (informally) since $\mathtt{Canfly}(Tweety)$ is consistent with what is known, we can apply the default to $Tweety$ and defeasibly derive $\mathtt{Canfly}(Tweety)$ from W. That this inference is indeed defeasible becomes apparent if $\mathtt{Penguin}(Tweety)$

is also added to W: then $\neg$Canfly($Tweety$) is classically entailed by what is known and the consistency check for applying the default fails, for which reason Canfly($Tweety$) cannot be derived from $W \cup \{$Penguin(Tweety)$\}$.

This example seems straightforward but the formal definition of default-logical consequence is tricky: in this approach, what is provable is determined by what is not provable, so the problem is how to avoid a circular definition. In default logic (as in related logics) this is solved by giving the definition a fixed-point appearance; see below in Section 5.4. Similar equilibrium-like definitions for argumentation systems will be discussed throughout this chapter.

Inconsistency handling

It has also been proposed to formalise defeasible reasoning as strategies for dealing with inconsistent information, e.g. by Brewka [1989]. In this approach defaults are formalised with ordinary material implications and without normality conditions, and their defeasible nature is captured in how they are used by the consistency handling strategies. In particular, in case of inconsistency, alternative consistent subsets (subtheories) of the premises give rise to alternative default conclusions, after which a choice can be made for the subtheory containing the exceptional rule.

In our birds example this works out as follows.

(1) bird $\rightarrow$ canfly
(2) penguin $\rightarrow \neg$ canfly
(3) bird
(4) penguin

The set $\{(1), (3)\}$ is a subtheory supporting the conclusion canfly, while $\{(2), (4)\}$ is a subtheory supporting the opposite. The exceptional nature of (2) over (1) can be captured by preferring the latter subtheory.

Systems for defeasible argumentation

Argumentation systems are yet another way to formalise nonmonotonic reasoning, viz. as the construction and comparison of arguments for and against certain conclusions. In these systems the basic notion is not that of a defeasible conditional but that of a defeasible argument. The idea is that the construction of arguments is monotonic, i.e., an argument stays an argument if more premises are added. Nonmonotonicity, or defeasibility, is not explained in terms of the interpretation of a defeasible conditional, but in terms of the interactions between conflicting arguments: in argumentation systems nonmonotonicity arises from the fact that new premises may give rise to stronger counterarguments, which defeat the original argument. So in case of Tweety 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.

Argumentation systems can be combined with each of the above-discussed views on defaults. The 'normality condition' view can be formalised by regarding an argument as a standard derivation from a set of premises augmented with normality statements. Thus a counterargument is an attack on such a normality statement. A variant of this method can be applied to the use of consistency and nonprovability expressions. The 'pragmatic' view on defaults (as in inconsistency handling) can be formalised by regarding arguments as a standard derivation from a consistent subset of the premises. Here a counterargument attacks a premise of an argument. Finally, the 'semantic' view on defaults could be formalised by allowing the construction of arguments with inference rules (such as modus ponens) that are invalid in the semantics. In that case a counterargument attacks the use of such an inference rule.

It is important to note, however, that argumentation systems have wider scope than just reasoning with defaults. Firstly, argumentation systems can be applied to any form of reasoning with contradictory information, whether the contradictions have to do with rules 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.

Most argumentation systems have been developed in artificial intelligence research on nonmonotonic reasoning, although Pollock's work, which was the first logical formalisation of defeasible argumentation, was initially applied to the philosophy of knowledge and justification (epistemology). The first artificial intelligence paper on argumentation systems was [Loui, 1987]. One domain in which argumentation systems have become popular is legal reasoning [Loui *et al.*, 1993, Prakken, 1993, Sartor, 1994, Gordon, 1995, Loui & Norman, 1995, Prakken & Sartor, 1996, Freeman & Farley, 1996, Prakken & Sartor, 1997a, Prakken, 1997, Gordon & Karacapilidis, 1997]. This is not surprising, since legal reasoning often takes place in an adversarial context, where notions like argument, counterargument, rebuttal and defeat are very common. However, argumentation systems have also been applied to such domains as medical reasoning [Das *et al.*, 1996], negotiation [Parsons *et al.*, 1998] and risk assessment in oil exploration [Clark, 1990].

2.2 Nonmonotonic reasoning: artificial intelligence or logic?

Usually, nonmonotonic logics are studied as a branch of artificial intelligence. However, it is more than justified to regard these logics as also part of philosophical logic. In fact, several issues in nonmonotonic logic have come up earlier in philosophy. For instance, in the context of moral reasoning, Ross [1930] has studied the notion of *prima facie* obligations. According to Ross an act is *prima facie*

obligatory if it has a characteristic that makes the act (by virtue of an underlying moral principle) *tend* to be a 'duty proper'. Fulfilling a promise is a *prima facie* duty because it is the fulfillment of a promise, i.e., because of the moral principle that one should do what one has promised to do. But the act may also have other characteristics which make the act tend to be forbidden. For instance, if John has promised a friend to visit him for a cup of tea, and then John's mother suddenly falls ill, then he also has a *prima facie* duty to do his mother's shopping, based, say, on the principle that we ought to help our parents when they need it. To find out what one's duty proper is, one should 'consider all things', i.e., compare all *prima facie* duties that can be based on any aspect of the factual circumstances and find which one is 'more incumbent' than any conflicting one. If we qualify the all-things-considered clause as 'consider all things that you know', then the reasoning involved is clearly nonmonotonic: if we are first only told that John has promised his friend to visit him, then we conclude that Johns' duty proper is to visit his friend. But if we next also hear that John's mother has become ill, we conclude instead that John's duty proper is to help his mother.

The term 'defeasibility' was first introduced not in logic but in legal philosophy, viz. by Hart [1949] (see the historical discussion in [Loui, 1995]). Hart observed that legal concepts are defeasible in the sense that the conditions for when a fact situation classifies as an instance of a legal concept (such as 'contract'), are only ordinarily, or presumptively, sufficient. If a party in a law suit succeeds in proving these conditions, this does not have the effect that the case is settled; instead, legal procedure is such that the burden of proof shifts to the opponent, whose turn it then is to prove additional facts which, despite the facts proven by the proponent, nevertheless prevent the claim from being granted (for instance, insanity of one of the contracting parties). Hart's discussion of this phenomenon stays within legal-procedural terms, but it is obvious that it provides a challenge for standard logic: an explanation is needed of how proving new facts without rejecting what was proven by the other party can reverse the outcome of a case.

Toulmin [1958], who criticised the logicians of his days for neglecting many features of ordinary reasoning, was aware of the implications of this phenomenon for logic. In his well-known pictorial scheme for arguments he leaves room for rebuttals of an argument. He also urges logicians to take the *procedural* aspect (in the legal sense) of argumentation seriously. In particular, Toulmin argues that (outside mathematics) an argument is valid if it can stand against criticism in a properly conducted dispute, and the task of logicians is to find criteria for when a dispute has been conducted properly.

The notion of burden of proof, and its role in dialectical inquiry, has also been studied by Rescher [1977], in the context of epistemology. Among other things, Rescher claims that a dialectical model of scientific reasoning can explain the rational force of inductive arguments: they must be accepted if they cannot be successfully challenged in a properly conducted scientific dispute. Rescher thereby assumes that the standards for *constructing* inductive arguments are somehow given by generally accepted practices of scientific reasoning; he only focuses on the di-

alectical interaction between conflicting inductive arguments.

Another philosopher who has studied defeasible reasoning is John Pollock. Although his work, to be presented below, is also well-known in the field of artificial intelligence, it was initially a contribution to epistemology, with, like Rescher, much attention for induction as a form of defeasible reasoning.

As this overview shows, a logical study of nonmonotonic, or defeasible reasoning fully deserves a place in philosophical logic. Let us now turn to the discussion of logics for defeasible argumentation.

3 SYSTEMS FOR DEFEASIBLE ARGUMENTATION: A CONCEPTUAL SKETCH

In this section we give a conceptual sketch of the general ideas behind logics for defeasible argumentation. These systems contain the following five elements (although sometimes implicitly): an underlying logical language, definitions of an argument, of conflicts between arguments and of defeat among arguments and, finally, a definition of the status of arguments, which can be used to define a notion of defeasible logical consequence.

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 premises cannot invalidate arguments as arguments but only give rise to 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. The notion of an argument corresponds to a proof (or the existence of 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. One argumentation system, viz. Dung [1995], leaves the internal structure of an argument completely unspecified. Dung treats the notion of an argument as a primitive, and exclusively focuses on the ways arguments interact. Thus Dung's framework is of the most abstract kind.

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 defeasible argumentation.

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. The first type is when arguments have contradictory conclusions, as in 'Tweety flies, because it is a bird' and 'Tweety does not fly because it is a penguin'

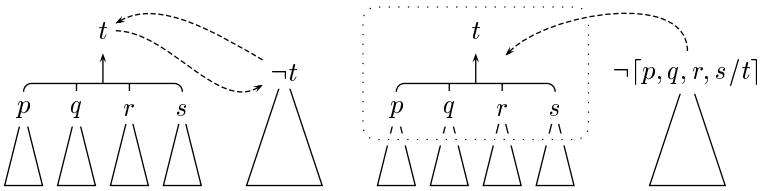


Figure 4. Rebutting attack (left) vs. undercutting attack (right).

(cf. the left part of Fig. 4). Clearly, this form of attack, which is often called *rebutting* an argument, is symmetric. The other two types of conflict are not symmetric. One is where one argument makes a non-provability assumption (as in default logic) and another argument proves what was assumed unprovable by the first. For example, an argument 'Tweety flies because it is a bird, and it is not provable that Tweety is a penguin', is attacked by any argument with conclusion 'Tweety is a penguin'. We shall call this *assumption attack*. The final type of conflict (first discussed by Pollock [1970]) is when one argument challenges, not a proposition, but a rule of inference of another argument (cf. the right part of Fig. 4). After Pollock, this is usually called *undercutting* an inference. Obviously, a rule of inference can only be undercut if it is not deductive. Non-deductive rules of inference occur in argumentation systems that allow inductive, abductive or analogical arguments. To consider an example, the inductive 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'. In order to formalise this type of conflict, the rule of inference that is to be undercut (in Fig. 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: $\neg \lceil p, q, r, s/t \rceil$.[3]

Note that all these senses 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 5 for indirect rebutting.

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' [Prakken & Sartor, 1997b], 'attack' [Dung, 1995, Bondarenko *et al.*, 1997] and 'interference' [Loui, 1998]. Other systems do not explicitly name this notion but leave it implicit in the definitions. In this chapter we shall use 'defeat' for the weak notion and 'strict defeat' for the strong, asymmetric notion. Note that the several forms of

[3]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.

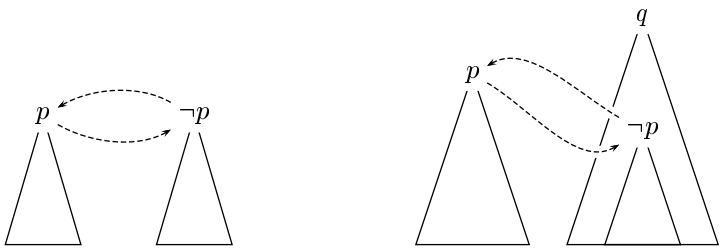


Figure 5. Direct attack (left) vs. indirect attack (right).

attack, rebutting vs. assumption vs. undercutting and direct vs. indirect, have their counterparts for defeat.

Argumentation systems vary in their grounds for the evaluation of arguments. In artificial intelligence the specificity principle, which prefers arguments based on the most specific defaults, is by many regarded as very important, but several researchers, e.g. Vreeswijk [1989], Pollock [1995] and Prakken & Sartor [1996], have argued that specificity is not a general principle of common-sense reasoning but just one of the many standards that might or might not be used. Moreover, some have claimed that general, domain-independent principles of defeat do not exist or are very weak, and that information from the semantics of the domain will be the most important way of deciding among competing arguments [Konolige, 1988, Vreeswijk, 1989]. For these reasons several argumentation systems are parametrised by user-provided criteria. Some, e.g. Prakken & Sartor, even argue that the evaluation criteria are part of the domain theory, and are debatable, just as the rest of the domain theory is, and that argumentation systems should therefore allow for defeasible arguments on these criteria. (Our example in the introduction contains such an argument, 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 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 A defeats argument B, but that A is itself defeated by a third argument C; in that case C 'reinstates' B (see Figure 6)[4]. Suppose, for instance, that the argument A that Tweety flies because it is 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

[4]While in figures 4 and 5 the arrows stood for attack relations, from now on they will depict defeat relations.

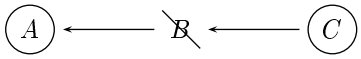


Figure 6. Argument C reinstates argument A.

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 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 (by Vreeswijk [1997] called the 'compositionality principle'). 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. (Cf. Fig. 5 on page 13.) 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, this chapter 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 premises 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.

In fact, this point is not universally accepted, and therefore we devote a separate subsection to it.

Semantics: model-theoretic or not?

A much-discussed issue is whether logics for nonmonotonic reasoning should have a model-theoretic semantics or not. In the early days of this field it was usual to

criticise several systems (such as default logic) for the lack of a model-theoretic semantics. However, when such semantics were provided, this was not always felt to be a major step forward, unlike when, for instance, possible-worlds semantics for modal logic was introduced. In addition, several researchers argued that non-monotonic reasoning needs a different kind of semantics than a model theory, viz. an argumentation-theoretic semantics. It is here not the place to decide the discussion. Instead we confine ourselves to presenting some main arguments for this view that have been put forward.

Traditionally, model theory has been used in logic to define the meaning of logical languages. Formulas of such languages were regarded as telling us something about reality (however defined). Model-theoretic semantics defines the meaning of logical symbols by defining how the world looks like if an expression with these symbols is true, and it defines logical consequence, entailment, by looking at what else must be true if the premises are true. For defaults this means that their semantics should be in terms of how the world normally, or typically looks like when defaults are true; logical consequence should, in this approach, be determined by looking at the most normal worlds, models or situations that satisfy the premises.

However, others, e.g. Pollock [1991, p. 40], Vreeswijk [1993a, pp. 88–9] and Loui [1998], have argued that the meaning of defaults should not be found in a correspondence with reality, but in their role in dialectical inquiry. That a relation between premises and conclusion is defeasible means that a certain burden of proof is induced. In this approach, the central notions of defeasible reasoning are notions like attack, rebuttal and defeat among arguments, and these notions are not 'propositional', for which reason their meaning is not naturally captured in terms of correspondence between a proposition and the world. This approach instead defines 'argumentation-theoretic' semantics for such notions. The basic idea of such a semantics is to capture sets of arguments that are as large as possible, and adequately defend themselves against attacks on their members.

It should be noted that this approach does not deny the usefulness of model theory but only wants to define its proper place. Model theory should not be applied for things for which it is not suitable, but should be reserved for the initial components of an argumentation system, the notions of a logical language and a consequence relation defining what an argument is.

It should also be noted, however, that some have proposed argumentation systems as proof theories for model-theoretic semantics of preferential entailment (in particular Geffner & Pearl [1992]). In our opinion, one criterion for success of such model-theoretic semantics of argumentation systems is whether *natural* criteria for model preference can be defined. For certain restricted cases this seems possible, but whether this approach is extendable to more general argumentation systems, for instance, those allowing inductive, analogical or abductive arguments, remains to be investigated.

4 GENERAL FEATURES OF ARGUMENT-BASED SEMANTICS

Let us now, before looking at some systems in detail, become more formal about some of the notions that these systems have in common. We shall focus in particular on the semantics of argumentation systems, i.e., on the conditions that sets of justified arguments should satisfy. In line with the discussion at the end of Section 3, we can say that argumentation systems 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. Let us concentrate on the task of defining the notion of a justified argument. Which properties should such a definition have?

Let us assume as background a set of arguments, with a binary relation of 'defeat' defined over it. 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. For the moment we leave the internal structure of an argument unspecified, as well as the precise definition of defeat.[5] Then a simple definition of the status of an argument is the following.

DEFINITION 1. Arguments are either justified or not justified.

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

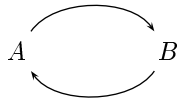
$A =$ 'Tweety flies because it is a bird'
$B =$ 'Tweety does not fly because it is a penguin'
$C =$ 'The observation that Tweety is a penguin is unreliable'

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 1 is circular or ambiguous. Especially when arguments of equal strength interfere with each other, it is not clear which argument should remain undefeated.

[5] This style of discussion is inspired by Dung [1995]; see further Subsection 5.1 below.

EXAMPLE 3. (Even cycle.) Consider the arguments A and B such that A defeats B and B defeats A.



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 1 is satisfied. However, it is obvious that by a completely 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 1: 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 1 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 subsections discuss the details of both approaches.

Figure 7. A self-defeating argument.

First, however, another problem with Definition 1 must be explained, having to do with self-defeating arguments.

EXAMPLE 4. (Self-defeat.) Consider an argument L, such that L defeats L. Suppose L is not justified. Then all arguments defeating L are not justified, so by clause 1 of Definition 1 L is justified. Contradiction. Suppose now L is justified. Then L is defeated by a justified argument, so by clause 2 of Definition 1 L is not justified. Contradiction.

Thus, Definition 1 implies that there are no self-defeating arguments. Yet the notion of self-defeating arguments seems intuitively plausible, as is illustrated by

the following example.

EXAMPLE 5. (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 lier. Is Hendrik a truth-teller or a lier?

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):

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

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 sub-arguments, and thus is a self-defeating argument.

In conclusion, it seems that Definition 1 needs another revision, to leave room for the existence of self-defeating arguments. Below we shall not discuss this in general terms since, perhaps surprisingly, in the literature it is hard to find generally applicable solutions to this problem. Instead we shall discuss for each particular system how it deals with self-defeat.

4.1 The unique-status-assignment approach

The idea to enforce unique status assignments basically comes in two variants. The first defines status assignments in terms of some fixed-point operator, and the second involves a recursive definition of a justified argument, by introducing the notion of a subargument of an argument. We first discuss the fixed-point approach.

Fixed-point definitions

This approach, followed by e.g. Pollock [1987, 1992], Simari & Loui [1992] and Prakken & Sartor [1997b], can best be explained with the notion of 'reinstatement' (see above, Section 3). The key observation is 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. This idea is captured by Dung's [1995] notion of *acceptability*.

DEFINITION 6. An argument A is *acceptable* with respect to a set S of arguments iff each argument defeating A is defeated by an argument in S.

The arguments in S can be seen as the arguments capable of reinstating A in case A is defeated[6].

However, the notion of acceptability is not sufficient. Consider in Example 3 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 a fixed-point operator must be used. Consider the following operator from [Dung, 1995], which for each set of arguments returns the set of all arguments that are acceptable to it.

DEFINITION 7. (Dung's [1995] grounded semantics.) Let *Args* be a set of arguments ordered by a binary relation of defeat, and let $S \subseteq$ *Args*. Then the operator F is defined as follows:

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

Dung proves that the operator F has a least fixed point. (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 3 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 $F(\emptyset) = \emptyset$.

These observations allow the following definition of a justified argument.

DEFINITION 8. An argument is *justified* iff it is a member of the least fixed point of F.

[6] As remarked above, Dung uses the term 'attack' instead of 'defeat'.

It is possible to reformulate Definition 7 in various ways, which are either equivalent to, or approximations of the least fixed point of F. To start with, Dung shows that it can be approximated from below, and when each argument has at most finitely many defeaters even be obtained, by iterative application of F to the empty set.

PROPOSITION 9. *Consider the following sequence of arguments.*

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

The following observations hold [Dung, 1995].

1. *All arguments in* $\cup_{i=0}^{\infty}(F^i)$ *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* $\cup_{i=0}^{\infty}(F^i)$

In the iterative construction 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. We have that

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

Dung [1995] also shows that F is equivalent to double application of a simpler operator G, i.e. $F = G \circ G$. The operator G returns for each set of arguments all arguments that are not defeated by any argument in that set.

DEFINITION 10. Let *Args* be a set of arguments ordered by a binary relation of defeat. Then the operator G is defined as follows:

- $G(S) = \{A \in Args | A \text{ is not defeated by any argument in } S\}$

The G operator is in turn very similar to the one used by Pollock [1987, 1992]. To see this, we reformulate G in Pollock's style, by considering the sequence obtained by iterative application of G to the empty set, and defining an argument A to be justified if and only if at some point (or "level") m in the sequence A remains in G_n for all $n \geq m$.

DEFINITION 11. (Levels in justification.)

- All arguments are *in at level* 0.

- An argument is *in at level* $n+1$ iff it is not defeated by any argument in at level n.
- An argument is *justified* iff there is an m such that for every $n \geq m$, the argument is in at level n.

As shown by Dung [1995], this definition stands to Definition 10 as the construction of Proposition 9 stands to Definition 7. Dung also remarks that Definition 11 is equivalent to Pollock's [1987, 1992] definition, but as we shall see below, this is not completely accurate.

In Example 2, Definition 11 works out as follows.

level	in
0	A, B, C
1	C
2	A, C
3	A, C
.	...

C is in at all levels, while A becomes in at 2 and stays in at all subsequent levels.

And in Example 3 both A and B are in at all even levels and out at all odd levels.

level	in
0	A, B
1	
2	A, B
3	
4	A, B
.	...

The following example, with an infinite chain of defeat relations, gives another illustration of Definitions 7 and 11.

EXAMPLE 12. (Infinite defeat chain.) 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.

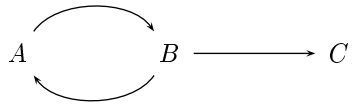
$$A_1 \longleftarrow A_2 \longleftarrow A_3 \longleftarrow A_4 \longleftarrow A_5 \longleftarrow \cdots$$

The least fixed point of this chain is empty, since no argument is undefeated. Consequently, $F(\emptyset) = \emptyset$. Note that this example has two other fixed points, which also satisfy Definition 1, viz. the set of all A_i where i is odd, and the set of all A_i where i is even.

Defensible arguments

A final peculiarity of the definitions is that they allow a distinction between two types of arguments that are not justified. Consider first again Example 2 and observe that, although B defeats A, A is still justified since it is reinstated by C. Consider next the following extension of Example 3.

EXAMPLE 13. (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 Definitions 8 and 11, neither of the three arguments are justified. For A and B this is since their relation is the same as in Example 3, 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, 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. Makinson & Schlechta [1991] call arguments like B 'zombie arguments'[7]: 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. Following Prakken & Sartor [1997b], we shall call this intermediate status 'defensible'. In the unique-status-assignment approach it can be defined as follows.

DEFINITION 14. (Overruled and defensible arguments.)

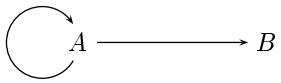
- An argument is *overruled* iff it is not justified, and defeated by a justified argument.
- An argument is *defensible* iff it is not justified and not overruled.

Self-defeating arguments

Finally, we must come back to the problem of self-defeating arguments. How does Definition 7 deal with them? Consider the following extension of Example 4.

EXAMPLE 15. Consider two arguments A and B such that A defeats A and A defeats B.

[7] Actually, they talk about 'zombie paths', since their article is about inheritance systems.



Intuitively, we want that B is justified, since the only argument defeating it is self-defeating. However, we have that $F(\emptyset) = \emptyset$, so neither A nor B are justified. Moreover, they are both defensible, since they are not defeated by any justified argument.

How can Definitions 7 and 11 be modified to obtain the intuitive result that A is overruled and B is justified? Here is where Pollock's deviation from the latter definition becomes relevant. His version is as follows.

DEFINITION 16. (Pollock, [1992])

- An argument is *in at level* 0 iff it is not self-defeating.
- An argument is *in at level* $n+1$ iff it is in at level 0 and it is not defeated by any argument in at level n.
- An argument is *justified* iff there is an m such that for every $n \geq m$, the argument is in at level n.

The additions *iff it is not self-defeating* in the first condition and *iff it is in at level* 0 in the second make the difference: they render all self-defeating arguments out at every level, and incapable of preventing other arguments from being out.

Another solution is provided by Prakken & Sartor [1997b] and Vreeswijk [1997], who distinguish a special 'empty' argument, which is not defeated by any other argument and which by definition defeats any self-defeating argument. Other solutions are possible, but we shall not pursue them here.

Recursive definitions

Sometimes a second approach to the enforcement of unique status assignments is employed, e.g. by Prakken [1993] and Nute [1994]. The idea is to make explicit that arguments are usually constructed step-by-step, proceeding from intermediate to final conclusions (as in Example 13, where A has an intermediate conclusion 'Dixon is a pacifist' and a final conclusion 'Dixon has no gun'). This approach results in an explicitly recursive definition of justified arguments, reflecting the basic intuition that an argument cannot be justified if not all its subarguments are justified. At first sight, this recursive style is very natural, particularly for implementing the definition in a computer program. However, the approach is not so straightforward as it seems, as the following discussion aims to show.

To formalise the recursive approach, we must make a first assumption on the structure of arguments, viz. that they have subarguments (which are 'proper' iff they are not identical to the entire argument). Justified arguments are then defined as follows. (We already add how self-defeating arguments can be dealt with, so

that our discussion can be confined to the issue of avoiding multiple status assignments. Note that the explicit notion of a subargument makes it possible to regard an argument as self-defeating if it defeats one of its subarguments, as in Example 5.)

DEFINITION 17. (Recursively justified arguments.) An argument A is *justified* iff

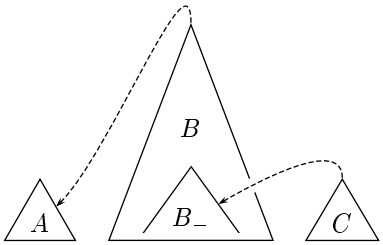
1. A is not self-defeating; and
2. All proper subarguments of A are justified; and
3. All arguments defeating A are self-defeating, or have at least one proper subargument that is not justified.

How does this definition avoid multiple status assignments in Example 3? The 'trick' is that for an argument to be justified, clause (2) requires that it have no (non self-defeating) defeaters of which all proper subarguments are justified. This is different in Definition 1, which leaves room for such defeaters, and instead requires that these themselves are not justified; thus this definition implies in Example 3 that A is justified if and only if B is not justified, inducing two status assignments. With Definition 17, on the other hand, A is prevented from being justified by the existence of a (non-selfdefeating) defeater with justified subarguments, viz. B (and likewise for B).

The reader might wonder whether this solution is not too drastic, since it would seem to give up the property of reinstatement. For instance, when applied to Example 2, Definition 17 says that argument A is not justified, since it is defeated by B, which is not self-defeating. That B is in turn defeated by C is irrelevant, even though C is justified.

However, here it is important that Definition 17 allows us to distinguish between two kinds of reinstatement. Intuitively, the reason why C defeats B in Example 2, is that it defeats B's proper subargument that Tweety is a penguin. And if the subarguments in the example are made explicit as follows, Definition 17 yields the intuitive result. (As for notation, for any pair of arguments X and X^-, the latter is a proper subargument of the first.)

EXAMPLE 18. Consider four arguments A, B, B^- and C such that B defeats A and C defeats B^-.



According to Definition 17, A and C are justified and B and B^- are not justified. Note that B is not justified by Clause 2. So C reinstates A not by directly defeating B but by defeating B's subargument B^-.

The crucial difference between the Examples 2 and 3 is that in the latter example the defeat relation is of a different kind, in that A and B are in conflict on their final conclusions (respectively that Nixon is, or is not a pacifist). The only way to reinstate, say, the argument A that Nixon was a pacifist is by finding a defeater of B's proper subargument that Nixon was a republican (while making the subargument relations explicit).

So the only case in which Definition 17 does not capture reinstatement is when all relevant defeat relations concern the final conclusions of the arguments involved. This might even be regarded as a virtue of the definition, as is illustrated by the following modification of Example 2 (taken from [Nute, 1994]).

EXAMPLE 19. Consider three arguments A, B and C such that B defeats A and C defeats B. Read the arguments as follows.

$A =$ 'Tweety flies because it is a bird'
$B =$ 'Tweety does not fly because it is a penguin'
$C =$ 'Tweety might fly because it is a genetically altered penguin'

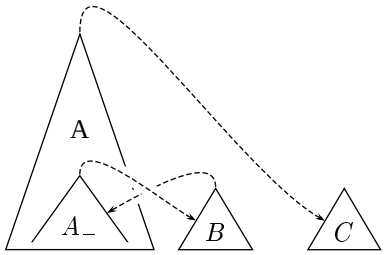
Note that, unlike in Example 2, these three arguments are in conflict on the same issue, viz. on whether Tweety can fly. According to Definitions 7 and 11 both A and C are justified; in particular, A is justified since it is reinstated by C. However, according to Definition 17 only C is justified, since A has a non-self-defeating defeater, viz. B. The latter outcome might be regarded as the intuitively correct one, since we still accept that Tweety is a penguin, which blocks the 'birds fly' default, and C allows us at most to conclude that Tweety *might* fly.

So does this example show that Definitions 7 and 11 must be modified? We think not, since it is possible to represent the arguments in such a way that these definitions give the intuitive outcome. However, this solution requires a particular logical language, for which reason its discussion must be postponed (see Section 5.2, p. 48).

Nevertheless, we can at least conclude that while the indirect form of reinstatement (by defeating a subargument) clearly seems a basic principle of argumentation, Example 19 shows that with direct reinstatement this is not so clear.

Unfortunately, Definition 17 is not yet fully adequate, as can be shown with the following extension of Example 3. It is a version of Example 13 with the subarguments made explicit.

EXAMPLE 20. (Zombie arguments 2.) Consider the arguments A^-, A, B and C such that A^- and B defeat each other and A defeats C.



A concrete example is

A^-	=	'Dixon is a pacifist because he is a quaker'
B	=	'Dixon is no pacifist because he is a republican'
A	=	'Dixon has no gun because he is a pacifist'
C	=	'Dixon has a gun because he lives in Chicago'

According to Definition 17, C is justified since its only defeater, A, has a proper subargument that is not justified, viz. A^-. Yet, as we explained above with Example 13, intuitively A should retain its capacity to prevent C from being justified, since the defeater of its subargument is not justified.

There is an obvious way to repair Definition 17: it must be made explicitly 'three-valued' by changing the phrase 'not justified' in Clause 3 into 'overruled',[8] where the latter term is defined as follows.

DEFINITION 21. (Defensible and overruled arguments 2.)

- An argument is *overruled* iff it is not justified and either it is self-defeating, or it or one of its proper subarguments is defeated by a justified argument.
- An argument is *defensible* iff it is not justified and not overruled.

This results in the following definition of justified arguments.

DEFINITION 22. (Recursively justified arguments—revised.) An argument A is *justified* iff

1. A is not self-defeating; and
2. All proper subarguments of A are justified; and
3. All arguments defeating A are self-defeating, or have at least one proper subargument that is overruled.

In Example 20 this has the following result. Note first that none of the arguments are self-defeating. Then to determine whether C is justified, we must determine the status of A. A defeats C, so C is only justified if A is overruled. Since A is

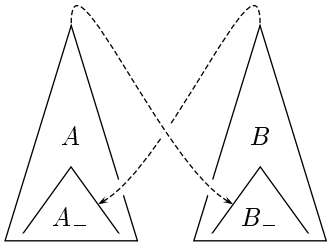
[8] Makinson & Schlechta [1991] criticise this possibility and recommend the approach with multiple status assignments.

not defeated, A can only be overruled if its proper subargument A^- is overruled. No proper subargument of A^- is defeated, but A^- is defeated by B. So if B is justified, A^- is overruled. Is B justified? No, since it is defeated by A^-, and A^- is not self-defeating and has no overruled proper subarguments. But then A is not overruled, which means that C is not justified. In fact, all arguments in the example are defensible, as can be easily verified.

Comparing fixed-point and recursive definitions

Comparing the fixed-point and recursive definitions, we have seen that in the main example where their outcomes differ (Example 19), the intuitions seem to favour the outcome of the recursive definitions (but see below, p. 48). We have also seen that the recursive definition, if made 'three-valued', can deal with zombie arguments just as well as the fixed-point definitions. So must we favour the recursive form? The answer is negative, since it also has a problem: Definitions 17 and 22 do not always enforce a unique status assignment. Consider the following example.

EXAMPLE 23. (Crossover defeat.)[9] Consider four arguments A^-, A, B^-, B such that A defeats B^- while B defeats A^-.



Definition 17 allows for two status assignments, viz. one in which only A^- and A are justified, and one in which only B^- and B are justified. In addition, Definition 22 also allows for the status assignment which makes all arguments defensible. Clearly, the latter status assignment is the intuitively intended one. However, without fixed-point constructions it seems hard to enforce it as the unique one.

Note, finally, that in our discussion of the non-recursive approach we implicitly assumed that when a proper subargument of an argument is defeated, thereby the argument itself is also defeated (see e.g. Example 2). In fact, any particular argumentation system that has no explicitly recursive definition of justified arguments should satisfy this assumption. By contrast, systems that have a recursive definition, can leave defeat of an argument independent from defeat of its proper subarguments. Furthermore, if a system has no recursive definition of justified arguments, but still distinguishes arguments and subarguments for other reasons

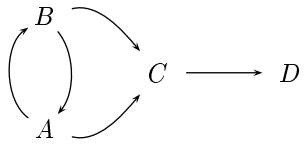
[9] The name 'crossover' is taken from Hunter [1993].

(as e.g. [Simari & Loui, 1992] and [Prakken & Sartor, 1997b]), then a proof is required that Clause 2 of Definition 17 holds. Further illustration of this point must be postponed to the discussion of concrete systems in Section 5.

Unique status assignments: evaluation

Evaluating the unique-status-assignment approach, we have seen that it can be formalised in an elegant way if fixed-point definitions are used, while the, perhaps more natural attempt with a recursive definition has some problems. However, regardless of its precise formalisation, this approach has inherent problems with certain types of examples, such as the following.

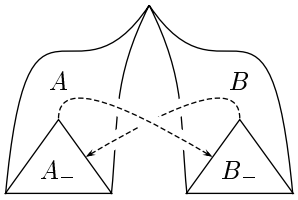
EXAMPLE 24. (Floating arguments.) Consider the arguments A, B, C and D such that A defeats B, B defeats A, A defeats C, B defeats C and C defeats D.



Since no argument is undefeated, Definition 8 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 again make an assumption on the structure of arguments, viz. that they have a conclusion.

EXAMPLE 25. (Floating conclusions.)[10] 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

[10]The term 'floating conclusions' was coined by Makinson & Schlechta [1991].

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

4.2 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. Note that both these assignments will satisfy Definition 1. In this approach, an argument is 'genuinely' justified iff it receives this status in all status assignments. To prevent terminological confusion, we now slightly reformulate the notion of a status assignment.

DEFINITION 26. A *status assignment* to a set X of arguments ordered by a binary defeat relation is an assignment to each argument of either the status 'in' or the status 'out' (but not both), satisfying the following conditions:

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

Note that the conditions (1) and (2) are just the conditions of Definition 1. In Example 3 there are precisely two possible 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? Since both of them are 'in' in one status assignment but 'out' in the other, we must conclude that neither of them is justified. This is captured by redefining the notion of a justified argument as follows:

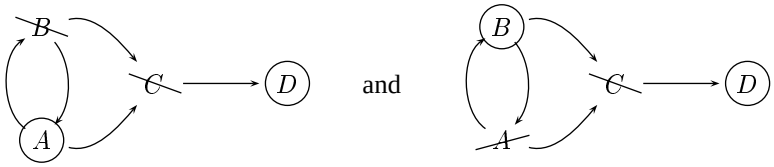
DEFINITION 27. Given a set X of arguments and a relation of defeat on X, an argument is *justified* iff it is 'in' in all status assignments to X.

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 extension, but others are at least in some extensions. The first category can be called the *overruled*, and the latter category the *defensible* arguments.

DEFINITION 28. Given a set X of arguments and a relation of defeat on X

- An argument is *overruled* iff it is 'out' in all status assignments to X;
- An argument is *defensible* iff it is 'in' in some and 'out' in some status assignments to X.

It is easy to see that the unique-assignment and multiple-assignments approaches are not equivalent. Consider again Example 24. Argument A and B form an even 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 two status assignments:



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

Multiple status assignments also make it possible to capture floating conclusions. This can be done by defining the status of formulas as follows.

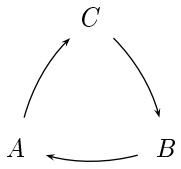
DEFINITION 29. (The status of conclusions.)

- φ is a *justified conclusion* iff every status assignment assigns 'in' to an argument with conclusion φ;
- φ is a *defensible conclusion* iff φ is not justified, and a conclusion of a defensible argument.
- φ is an *overruled conclusion* iff φ is not justified or defensible, and a conclusion of an overruled argument.

Changing the first clause into 'φ *is a* justified conclusion *iff* φ *is the conclusion of a justified argument*' would express a stronger notion, not recognising floating conclusions as justified.

There is reason to distinguish several variants of the multiple-status-assignments approach. Consider the following example, with an 'odd loop' of defeat relations.

EXAMPLE 30. (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 27 has some problems, since this example has no 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 30, viz. the case where B and C are identical to A. This means that sets of arguments containing a self-defeating argument have no status assignment.

To deal with the problem of odd defeat cycles, several alternatives to Definition 26 have been studied in the literature. They will be discussed in Section 5, in particular in 5.1 and 5.2.

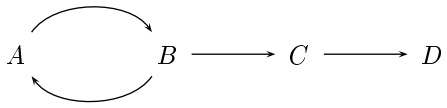
4.3 Comparing the two approaches

How do the unique- and multiple-assignment approaches compare to each other? It is sometimes said that their difference reflects a difference between a 'sceptical' and 'credulous' attitude towards drawing defeasible conclusions: when faced with an unresolvable conflict between two arguments, a sceptic 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 sceptical 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 this interpretation of the two approaches is incorrect. 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 evaluated given these assignments. And this evaluation is captured by the qualifications 'justified' and 'defensible', which thus capture the distinction between 'sceptical' 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 'sceptical' vs. 'credulous' reasoning.

Although both approaches can capture the notion of a defensible argument, they do so with one important difference. The multiple-assignments approach is more convenient for identifying *sets* of arguments that are compatible with each other. The reason is that while with unique assignments the defensible arguments are defensible on an individual basis, with multiple assignments they are defensible because they belong to a set of arguments that are 'in' and thus can be defended simultaneously. Even if two defensible arguments do not defeat each other, they might be incompatible in the sense that no status assignment makes them both 'in', as in the following example.

EXAMPLE 31. A and B defeat each other, B defeats C, C defeats D.



This example has two status assignments, viz. $\{A, C\}$ and $\{B, D\}$. Accordingly, all four arguments are defensible. Note that, although A and D do not defeat each other, A is in iff D is out. So A and D are in some sense incompatible. In the unique-assignment approach this notion of incompatibility seems harder to capture.

As we have seen, the unique-assignment approach has no inherent difficulty to recognise 'zombie arguments'; this problem only occurs if this approach uses a recursive two-valued definition of the status of arguments.

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 Brygt Rykkje likes ice skating in Example 25 is justified is different from the sense in which, for instance, the conclusion that Tweety flies in Example 2 is justified: only in the second case is the conclusion supported by a justified argument. And the status of D in Example 24 is not quite the same as the status of, for instance, A in Example 2. 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. In the concluding section we come back to this point, and generalise it to other differences between the various systems.

4.4 General properties of consequence notions

We conclude this section with a much-discussed issue, viz. whether any nonmonotonic consequence notion, although lacking the property of monotonicity, should still satisfy other criteria. Many argue that this is the case, and much research has been devoted to formulating such criteria and designing systems that satisfy them; see e.g. [Gabbay, 1985, Makinson, 1989, Kraus *et al.*, 1990]. We, however, do not follow this approach, since we think that it is hard to find any criterion that should really hold for any argumentation system, or nonmonotonic consequence notion, for that matter. We shall illustrate this with the condition that is perhaps most often defended, called *cumulativity*. In terms of argumentation systems this principle says that if a formula φ is justified on the basis of a set of premises T, then any formula ψ is justified on the basis of T if and only if ψ is also justified on the basis of $T \cup \{\varphi\}$. We shall in particular give counterexamples to the 'if' part of the biconditional, which is often called *cautious monotony*. This condition in fact says that adding justified conclusions to the premises cannot make other justified conclusions unjustified.

At first sight, this principle would seem uncontroversial. However, we shall now (quasi-formally) discuss reasonably behaving argumentation systems, with plausible criteria for defeat, and show by example that they do not satisfy cautious monotony and are therefore not cumulative. These examples illustrate two points. First they illustrate Makinson & Schlechta's [1991] remark that systems that do not satisfy cumulativity assign facts a special status. Second, since the examples are quite natural, they illustrate that argumentation systems *should* assign facts a special status and therefore should not be cumulative.

Below, the $\longrightarrow$ symbols stand for unspecified reasoning steps in an argument, and the formulas stand for the conclusion drawn in such a step.

EXAMPLE 32. Consider two (schematic) arguments

$A: \; p \longrightarrow q \longrightarrow r \longrightarrow \neg q \longrightarrow s$
$B: \longrightarrow \neg s$

Suppose we have a system in which self-defeating arguments have no capacity to prevent other arguments from being justified. Assume also that A is self-defeating, since a subconclusion, $\neg q$, is based on a subargument for a conclusion q. Assume, finally, that the system makes A's subargument for r justified (since it has no non-selfdefeating counterarguments). Then B is justified. However, if r is now added to the 'facts', the following argument can be constructed:

$A': \; r \longrightarrow \neg q \longrightarrow s$

This argument is not self-defeating, and therefore it might have the capacity to prevent B from being justified.

EXAMPLE 33. Consider next the following arguments.

A is a two-step argument $p \longrightarrow q \dashrightarrow r$
B is a three-step argument $s \longrightarrow t \longrightarrow u \dashrightarrow \neg r$

And assume that conflicting arguments are compared on their length (the shorter, the better). Then A strictly defeats B, so A is justified. Assume, however, also that B's subargument

$$s \longrightarrow t \longrightarrow u$$

is justified, since it has no counterarguments, and assume that u is added to the facts. Then we have a new argument for $\neg r$, viz.

$$B' : \ u \longrightarrow \neg r$$

which is shorter than A and therefore strictly defeats A.

Yet another type of example uses numerical assessments of arguments.

EXAMPLE 34. Consider the arguments

$A : \ p \longrightarrow q \longrightarrow r$
$B : \ s \longrightarrow \neg r$

Assume that in A the strength of the derivation of q from p is 0.7 and that the strength of the derivation of r from q is 0.85, while in B the strength of the derivation of $\neg r$ from s is 0.8. Consider now an argumentation system where arguments are compared with respect to their weakest links. Then B strictly defeats A, since B's weakest link is 0.8 while A's weakest link is 0.7. However, assume once more that A's subargument for q is justified because it has no counterargument, and then assume that q is added as a fact. Then a new argument

$$A' : \ q \longrightarrow r$$

can be constructed, with as weakest link 0.85, so that it strictly defeats B.

The point of these examples is that reasonable argumentation systems with plausible criteria for defeat are conceivable which do not satisfy cumulativity, so that cumulativity cannot be required as a minimum requirement for justified belief. Vreeswijk [1993a, pp. 82–8] has shown that other properties of nonmonotonic consequence relations also turn out to be counterintuitive in a number of realistic logical scenario's.

5 SOME ARGUMENTATION SYSTEMS

Let us, after our general discussions, now turn to individual argumentation systems and frameworks. We shall present them according to the conceptual sketch of Section 3, and also evaluate them in the light of Section 4.

5.1 *The abstract approach of Bondarenko, Dung, Kowalski and Toni*

Introductory remarks

We first discuss an abstract approach to nonmonotonic logic developed in several articles by Bondarenko, Dung, Toni and Kowalski (below called the 'BDKT approach'). Historically, this work came after the development by others of a number of argumentation systems (to be discussed below). The major innovation of the BDKT approach is that it provides a framework and vocabulary for investigating the general features of these other systems, and also of nonmonotonic logics that are not argument-based.

The latest and most comprehensive account of the BDKT approach is Bondarenko *et al.* [1997]. In this account, the basic notion is that of a set of "assumptions". In their approach the premises come in two kinds: 'ordinary' premises, comprising a *theory*, and *assumptions*, which are formulas (of whatever form) that are designated (on whatever ground) as having default status. Inspired by Poole [1988], Bondarenko *et al.* [1997] regard nonmonotonic reasoning as adding sets of assumptions to theories formulated in an underlying monotonic logic, provided that the contrary of the assumptions cannot be shown. What in their view makes the theory argumentation-theoretic is that this provision is formalised in terms of sets of assumptions attacking each other. In other words, according to Bondarenko *et al.* [1997] an argument is a set of assumptions. This approach has especially proven successful in capturing existing nonmonotonic logics.

Another version of the BDKT approach, presented by Dung [1995], completely abstracts from both the internal structure of an argument and the origin of the set of arguments; all that is assumed is the existence of a set of arguments, ordered by a binary relation of 'defeat'.[11] This more abstract point of view seems more in line with the aims of this chapter, and therefore we shall below mainly discuss Dung's version of the BDKT approach. As remarked above, it inspired much of our discussion in Section 4. The assumption-based version of Bondarenko *et al.* [1997] will be briefly outlined at the end of this subsection.

Basic notions

As just remarked, Dung's [1995] primitive notion is a set of arguments ordered by a binary relation of defeat. Dung then defines various notions of so-called argument extensions, which are intended to capture various types of defeasible consequence. These notions are declarative, just declaring sets of arguments as having a certain status. Finally, Dung shows that many existing nonmonotonic logics can be reformulated as instances of the abstract framework.

Dung's basic formal notions are as follows.

[11] BDKT use the term 'attack', but to maintain uniformity we shall use 'defeat'.

DEFINITION 35. An *argumentation framework* (AF) is a pair ($Args$, defeat), where $Args$ is a set of arguments, and defeat a binary relation on $Args$.

- An AF is *finitary* iff each argument in $Args$ is defeated by at most a finite number of arguments in $Args$.
- A set of arguments is *conflict-free* iff no argument in the set is defeated by an argument in the set.

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 discussions below of 'partial computation'). Unless stated otherwise, we shall below implicitly assume an arbitrary but fixed AF.

Dung interprets *defeat*, like us, in the weak sense of 'conflicting and not being weaker'. Thus in Dung's approach two arguments can defeat each other. Dung does not explicitly use the stronger (and asymmetric) notion of strict defeat, but we shall sometimes use it below.

A central notion of Dung's framework is acceptability, already defined above in Definition 6. We repeat it here. It captures how an argument that cannot defend itself, can be protected from attacks by a set of arguments.

DEFINITION 36. 6 An argument A is *acceptable* with respect to a set S of arguments iff each argument defeating A is defeated by an argument in S.

As remarked above, 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, which in terms of Dung has an AF (called 'TT' for 'Tweety Triangle') with $Args = \{A, B, C\}$ and *defeat* = $\{(B, A), (C, B)\}$ (B strictly defeats A and C strictly defeats B). A is acceptable with respect to $\{C\}$, $\{A, C\}$, $\{B, C\}$ and $\{A, B, C\}$, but not with respect to $\emptyset$ and $\{B\}$.

Another central notion is that of an admissible set.

DEFINITION 37. A conflict-free set of arguments S is *admissible* iff 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 the sets $\emptyset$, $\{C\}$ and $\{A, C\}$ are admissible but all other subsets of $\{A, B, C\}$ are not admissible.

Argument extensions

In terms of the notions of acceptability and admissibility several notions of 'argument extensions' can be defined, which are what we above called 'status assignments'. The following notion of a stable extension is equivalent to Definition 26 above.

DEFINITION 38. A conflict-free set S is a *stable extension* iff every argument that is not in S, is defeated by some argument in S.

In Example 2, TT has only one stable extension, viz. $\{A, C\}$. Consider next an AF called ND (the Nixon Diamond), corresponding to Example 3, with $Args = \{A, B\}$, and *defeat* $= \{(A, B), (B, A)\}$. ND has two stable extensions, $\{A\}$ and $\{B\}$.

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 AF's have stable extensions, as Example 30 has shown.

To give such examples also a credulous semantics, Dung defines the notion of a preferred extension.

DEFINITION 39. A conflict-free set is a *preferred extension* iff it is a maximal (with respect to set inclusion) admissible set.

Let us go back to Definition 26 of a status assignment and define a *partial status assignment* in the same way as a status assignment, but without the condition that it assigns a status to all arguments. Then it is easy to verify that preferred extensions correspond to maximal partial status assignments.

Dung shows that every AF has a preferred extension. Moreover, he shows that stable extensions are preferred extensions, so in the Nixon Diamond and the Tweety Triangle the two semantics coincide. However, not all preferred extensions are stable: in Example 30 the empty set is a (unique) preferred extension, which is not stable. Preferred semantics leaves all arguments in an odd defeat cycle out of the extension, so none of them is defeated by an argument in the extension.

Preferred and stable semantics are an instance of the multiple-status-assignments approach of Section 4.2: in cases of an irresolvable conflict as in the Nixon diamond, two incompatible extensions are obtained. Dung also explores the unique-status-assignment approach, with his notion of a *grounded* extension, already presented above as Definition 7. To build a bridge between the various semantics, Dung also defines 'complete semantics'.

DEFINITION 40. An admissible set of arguments is a *complete extension* iff each argument that is acceptable with respect to S belongs to S.

This definition implies that a set of arguments is a complete extension iff it is a fixed point of the operator F defined in Definition 7. According to Dung, a complete extension captures the beliefs of a rational person who believes everything s/he can defend.

Self-defeating arguments

How do Dung's various semantics deal with self-defeating arguments? It turns out that all semantics have some problems. For stable semantics they are the most seri-

ous, since an AF with a self-defeating argument has no stable extensions. For preferred semantics this problem does not arise, since preferred extensions are guaranteed to exist. However, this semantics still has a problem, since self-defeating arguments can prevent other arguments from being justified. This can be illustrated with Example 15 (an AF with 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. Yet intuitively it seems that instead $\{B\}$ should be the only preferred extension, since B's only defeater is self-defeating. It is easy to see that the same holds for complete semantics. In Section 4.1 we already saw that this example causes the same problems for grounded semantics, but that for finitary AF's Pollock [1987] provides a solution. Both Dung [1995] and Bondarenko *et al.* [1997] recognise the problem of self-defeating arguments, and suggest that solutions in the context of logic programming of Kakas *et al.* [1994] could be generalised to deal with it. Dung also acknowledges Pollock's [1995] approach, to be discussed in Subsection 5.2.

Formal results

Both Dung [1995] and Bondarenko *et al.* [1997] establish a number of results on the existence of extensions and the relation between the various semantics. We now summarise some of them.

1. Every stable extension is preferred, but not vice versa.

2. Every preferred extension is a complete extension, but not vice versa.

3. The grounded extension is the least (with respect to set inclusion) complete extension.

4. The grounded extension is contained in the intersection of all preferred extensions (Example 24 is a counterexample against 'equal to'.)

5. If an AF contains no infinite chains $A_1, \ldots, A_n, \ldots$ such that each A_{i+1} defeats A_i then AF has exactly one complete extension, which is grounded, preferred and stable. (Note that the even loop of Example 3 and the odd loop of Example 30 form such an infinite chain.)

6. Every AF has at least one preferred extension.

7. Every AF has exactly one grounded extension.

Finally, Dung [1995] and Bondarenko *et al.* [1997] identify several conditions under which preferred and stable semantics coincide.

Assumption-based formulation of the framework

As mentioned above, Bondarenko *et al.* [1997] have developed a different version of the BDKT approach. This version is less abstract than the one of Dung [1995], in that it embodies a particular view on the structure of arguments. Arguments are seen as sets of assumptions that can be added to a theory in order to (monotonically) derive conclusions that cannot be derived from the theory alone. Accordingly, Bondarenko *et al.* [1997] define a more concrete version of Dung's [1995] argumentation frameworks as follows:

DEFINITION 41. Let $\mathcal{L}$ be a formal language and $\vdash$ a monotonic logic defined over $\mathcal{L}$. An *assumption-based framework* with respect to $(\mathcal{L}, \vdash)$ is a tuple $\langle T, Ab, \overline{} \rangle$ where

- $T, Ab \subseteq \mathcal{L}$
- $\overline{}$ is a mapping from Ab into $\mathcal{L}$, where $\overline{\alpha}$ denotes the *contrary* of α.

The notion of *defeat* is now defined for sets of assumptions (below we leave the assumption-based framework implicit).

DEFINITION 42. A set of assumptions A *defeats* an assumption α iff $T \cup A \vdash \overline{\alpha}$; and A defeats a set of assumptions Δ iff A defeats some assumption $\alpha \in \Delta$.

The notions of argument extensions are then defined in terms of sets of assumptions. For instance,

DEFINITION 43. A set of assumptions Δ is *stable* iff

- Δ is closed, i.e., $\Delta = \{\alpha \in Ab | T \cup \Delta \vdash \alpha\}$
- Δ does not defeat itself
- Δ defeats each assumption $\alpha \notin \Delta$

A *stable extension* is a set $Th(T \cup \Delta)$ for some stable set Δ of assumptions.

As remarked above, Bondarenko *et al.*'s [1997] main aim is to reformulate existing nonmonotonic logics in their general framework. Accordingly, what an assumption is, and what its contrary is, is determined by the choice of nonmonotonic logic to be reformulated. For instance, in applications of preferential entailment where abnormality predicates $\mathtt{ab}_i$ are to be minimised (see Section 2.1), the assumptions will include expressions of the form $\neg\mathtt{ab}_i(c)$, where $\overline{\neg\mathtt{ab}_i(c)} = \mathtt{ab}_i(c)$. And in default logic (see also Section 2.1), an assumption is of the form $M\varphi$ for any 'middle part' φ of a default, where $\overline{M\varphi} = \neg\varphi$; moreover, all defaults $\varphi{:}\psi/\chi$ are added to the rules defining $\vdash$ as monotonic inference rules $\varphi, M\psi/\chi$.

Procedure

The developers of the BDKT approach have also studied procedural forms for the various semantics. Dung *et al.* [1996, 1997] propose two abstract proof procedures

for computing admissibility (Definition 37), where the second proof procedure is a computationally more efficient refinement of the first. Both procedures are based upon a proof procedure originally intended for computing stable semantics in logic programming. And they are both formulated as logic programs that are derived from a formal specification. The derivation guarantees the correctness of the proof procedures. Further, Dung *et al.* [1997] show that both proof procedures are complete. Here, the first procedure is discussed.

It is defined in the form of a meta-level logic program, of which the top-level clause defines admissibility. This concept is captured in a predicate adm:

(1) $adm(\Delta_0, \Delta) \longleftrightarrow [\Delta_0 \subseteq \Delta$ and Δ is admissible$]$

Δ and Δ_0 are sets of assumptions, where 'Δ is admissible' is a low-level concept that is defined with the help of auxiliary clauses. In this manner, (1) provides a specification for the proof procedure. Similarly, a top-level predicate $defends$ is defined

$$defends(D, \Delta) \longleftrightarrow [D \text{ defeats } \Delta' - \Delta, \text{ for every } \Delta' \text{ that defeats } \Delta]$$

The proof procedure that Dung *et al.* propose can be understood in procedural terms as repeatedly adding defences to the initially given set of assumptions Δ_0 until no further defences need to be added. More precisely,

> given a current set of assumptions Δ, initialised as Δ_0, the proof procedure repeatedly
>
> 1. finds a set of assumptions D such that $defends(D, \Delta)$;
> 2. replaces Δ by $\Delta \cup D$
>
> until $D = \Delta$, in which case it returns Δ.

Step (1) is non-deterministic, since there might be more than one set of assumptions D defending the current Δ. The proof procedure potentially needs to explore a search tree of alternatives to find a branch which terminates with a self-defending set. The logic-programming formulation of the proof procedure is:

$$\begin{array}{lcl} adm(\Delta, \Delta) & \leftarrow & defends(\Delta, \Delta) \\ adm(\Delta, \Delta') & \leftarrow & defends(D, \Delta),\ adm(\Delta \cup D, \Delta') \end{array}$$

The procedural characterisation of the proof procedure is obtained by applying SLD resolution to the above clauses with a left-to-right selection rule, with an initial query of the form $\leftarrow adm(\Delta_0, \Delta)$ with Δ_0 as input and Δ as output.

The procedure is proved correct with respect to the admissibility semantics, but it is shown to be incorrect for stable semantics in general. According to Dung *et al.*, this is due to the above-mentioned 'epistemic aggressiveness' of stable semantics, viz. the fact that a stable extension defeats every argument not belonging to it. Dung *et al.* remark that, besides being counterintuitive, this property is also

computationally very expensive, because it necessitates a search through the entire space of arguments to determine, for every argument, whether or not it is defeated. Subsequent evaluation by Dung *et al.* of the proof procedure has suggested that it is the semantics, rather than the proof procedure, which was at fault, and that preferred semantics provides an improvement. This insight is also formulated by Dung [1995].

Finally, it should be noted that recently, Kakas & Toni [1999] have developed proof procedures in dialectical style (see Section 6 below) for the various semantics of Bondarenko *et al.* [1997] and for Kakas *et al.* [1994]'s acceptability semantics.

Evaluation

As remarked above, the abstract BDKT 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 and Bondarenko *et al.* extensively show how many 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, which was shown by Dung [1995] to have a stable semantics, can very easily be given an alternative semantics in which extensions are guaranteed to exist, 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. Because of these features, the BDKT framework is also very useful as guidance in the development of new systems, as, for instance, Prakken & Sartor have used it in developing the system of Subsection 5.7 below.

On the other hand, the level of abstractness of the BDKT approach (especially in Dung's version) 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. Moreover, it seems that at some points the BDKT approach needs to be refined or extended. We already mentioned the treatment of self-defeating arguments, and Prakken & Sartor [1997b] have extended the BDKT framework to let it cope with reasoning about priorities (see Subsection 5.7 below).

5.2 Pollock

John Pollock was one of the initiators of the argument-based approach to the formalisation of defeasible reasoning. Originally he developed his theory as a contribution to philosophy, in particular epistemology. Later he turned to artificial intelligence, developing a computer program called OSCAR, which implements his theory. Since the program falls outside the scope of this handbook, we shall only discuss the logical aspects of Pollock's system; for the architecture of the

computer program the reader is referred to e.g. Pollock [1995]. The latter also discusses other topics, such as practical reasoning, planning and reasoning about action.

Reasons, arguments, conflict and defeat

In Pollock's system, the underlying logical language is standard first-order logic, but the notion of an argument has some nonstandard features. What still conforms to accounts of deductive logic is that arguments are sequences of propositions linked by inference rules (or better, by instantiated inference schemes). However, Pollocks's formalism begins to deviate when we look at the kinds of inference schemes that can be used to build arguments. Let us first concentrate on linear arguments; these are formed by combining so-called *reasons*. Technically, reasons connect a set of propositions with a proposition. Reasons come in two kinds, conclusive and *prima facie* reasons.

Conclusive reasons still adhere to the common standard, since they are reasons that logically entail their conclusions. In other words, a conclusive reason is any valid first-order inference scheme (which means that Pollock's system includes first-order logic). Thus, examples of conclusive reasons are

$\{p, q\}$ is a conclusive reason for $p \wedge q$
$\{\forall x P x\}$ is a conclusive reason for Pa

Prima facie reasons, by contrast have no counterpart in deductive logic; they only create a presumption in favour of their conclusion, which can be defeated by other reasons, depending on the strengths of the conflicting reasons. Based on his work in epistemology, Pollock distinguishes several kinds of *prima facie* reasons: for instance, principles of perception, such as[12]

$\lceil x$ appears to me as $Y \rceil$ is a *prima facie* reason for believing $\lceil x$ is $Y \rceil$.

(For the objectification-operator $\lceil \cdot \rceil$ see page 12 and page 44.)

Another source of *prima facie* reasons is the statistical syllogism, which says that:

If $(r > 0.5)$ then $\lceil x$ is an F and prob$(G/F) = r \rceil$ is a *prima facie* reason of strength r for believing $\lceil x$ is a $G \rceil$.

Here prob(G/F) stands for the conditional probability of G given F.

Prima facie reasons can also be based on principles of induction, for example,

$\lceil X$ is a set of m F's and n members of X have the property G $(n/m > 0.5) \rceil$ is a *prima facie* reason of strength n/m for believing $\lceil$all F's have the property $G \rceil$.

[12] When a reason for a proposition is a singleton set, we drop the brackets.

Actually, Pollock adds to these definitions the condition that F is *projectible* with respect to G. This condition, introduced by Goodman, 1954, is meant to prevent certain 'unfounded' probabilistic or inductive inferences. For instance, the first observed person from Lanikai, who is a genius, does not permit the prediction that the next observed Lanikaian will be a genius. That is, the predicate 'intelligence' is not projectible with respect to 'birthplace'. Projectibility is of major concern in probabilistic reasoning.

To give a simple example of a linear argument, assume the following set of 'input' facts INPUT $= \{A(a), \text{prob}(B/A) = 0.8, \text{prob}(C/B) = 0.7\}$. The following argument uses reasons based on the statistical syllogism, and the first of the above-displayed conclusive reasons.

1.	$\langle A(a), \infty\rangle$	($A(a)$ is in INPUT)
2.	$\langle \text{prob}(B/A) = 0.8, \infty\rangle$	(prob$(B/A) = 0.8$ is in INPUT)
3.	$\langle A(a) \wedge \text{prob}(B/A) = 0.8, \infty\rangle$	(1,2 and $\{p, q\}$ is a conclusive reason for $p \wedge q$)
4.	$\langle B(a), 0.8\rangle$	(3 and the statistical syllogism)
5.	$\langle \text{prob}(C/B) = 0.7, \infty\rangle$	(prob$(C/B) = 0.7$ is in INPUT)
6.	$\langle B(a) \wedge \text{prob}(C/B) = 0.7, 0.8\rangle$	(4,5 and $\{p, q\}$ is a conclusive reason for $p \wedge q$)
7.	$\langle C(a), 0.7\rangle$	(6 and the statistical syllogism)

So each line of a linear argument is a pair, consisting of a proposition and a numerical value that indicates the strength, or degree of justification of the proposition. The strength ∞ at lines 1,2 and 5 indicates that the conclusions of these lines are put forward as absolute facts, originating from the epistemic base 'INPUT'. At line 4, the *weakest link* principle is applied, with the result that the strength of the argument line is the minimum of the strength of the reason for $B(a)$ (0.8) and the argument line 3 from which $C(a)$ is derived with this reason (∞). At lines 6 and 7 the weakest link principle is applied again.

Besides linear arguments, Pollock also studies *suppositional* arguments. In suppositional reasoning, we 'suppose' something that we have not inferred from the input, draw conclusions from the supposition, and then 'discharge' the supposition to obtain a related conclusion that no longer depends on the supposition. In Pollock's system, suppositional arguments can be constructed with inference rules familiar from natural deduction. Accordingly, the propositions in an argument have sets of propositions attached to them, which are the *suppositions* under which the proposition can be derived from earlier elements in the sequence.

The following definition (based on [Pollock, 1995]) summarises this informal account of argument formation.

DEFINITION 44. In OSCAR, an *argument* based on INPUT is a finite sequence $\sigma_1, \ldots, \sigma_n$, where each σ_i is a line of argument. A *line of argument* σ_i is a triple $\langle X_i, p_i, \nu_i\rangle$, where X_i, a set of propositions, is the set of *suppositions* at line i, p_i is a proposition, and ν_i is the *degree of justification* of σ at line i. A line of argument

is obtained from earlier lines of argument according to one of the following rules of argument formation.

Input. If p is in INPUT and σ is an argument, then for any X it holds that $\sigma, \langle X, p, \infty \rangle$ is an argument.

Reason. If σ is an argument, $\langle X_1, p_1, \eta_1 \rangle, \ldots, \langle X_n, p_n, \eta_n \rangle$ are members of σ, and $\{p_1, \ldots, p_n\}$ is a reason of strength ν for q, and for each i, $X_i \subset X$, then $\sigma, \langle X, q, \min\{\eta_1, \ldots \eta_n, \nu\} \rangle$ is an argument.

Supposition. If σ is an argument, X a set of propositions and $p \in X$, then $\sigma, \langle X, p, \infty \rangle$ is also an argument.

Conditionalisation. If σ is an argument and some line of σ is $\langle X \cup \{p\}, q, \nu \rangle$, then $\sigma, \langle X, (p \rightarrow q), \nu \rangle$ is also an argument.

Dilemma. If σ is an argument and some line of σ is $\langle X, p \vee q, \nu \rangle$, and some line of σ is $\langle X \cup \{p\}, r, \mu \rangle$, and some line of σ is $\langle X \cup \{q\}, r, \xi \rangle$, then $\sigma, \langle X, r, \min\{\nu, \mu, \xi\} \rangle$ is also an argument.

Pollock [1995] notes that other inference rules could be added as well.

It is the use of *prima facie* reasons that makes arguments defeasible, since these reasons can be defeated by other reasons. This can take place in two ways: by *rebutting* defeaters, which are at least as strong reasons with the opposite conclusion, and by *undercutting* defeaters, which are at least as strong reasons of which the conclusion denies the connection that the undercut reason states between its premises and its conclusion. A typical example of rebutting defeat is when an argument using the reason 'Birds fly' is defeated by an argument using the reason 'Penguins don't fly'. Pollock's favourite example of an undercutting defeater is when an object looks red because it is illuminated by a red light: knowing this undercuts the reason for believing that this object is red, but it does not give a reason for believing that the object is not red.

Before we can explain how Pollock formally defines the relation of defeat among arguments, some extra notation must be introduced. In the definition of defeat among arguments, Pollock uses a, what may be called, *objectification* operator, $\lceil \cdot \rceil$. (This operator was also used in Fig. 4 on page 12 and in the *prima facie* reasons on page 42.) With this operator, expressions in the meta-language are transformed into expressions in the object language. For example, the meta-level rule

$$\{p, q\} \text{ is a conclusive reason for } p$$

may be transformed into the object-level expression

$$\lceil \{p, q\} \text{ is a conclusive reason for } p \rceil .$$

If the object language is rich enough, then the latter expression is present in the object language, in the form $(p \wedge q) \supset q$. Evidently, a large fraction of the meta-expressions cannot be conveyed to the object language, because the object language lacks sufficient expressibility. This is the case, for example, if corresponding connectives are missing in the object language.

Pollock formally defines the relation of defeat among arguments as follows.

Defeat among arguments. An argument σ defeats another argument η if and only if:

1. η's last line is $\langle X, q, \alpha\rangle$ and is obtained by the argument formation rule *Reason* from some earlier lines $\langle X_1, p_1, \alpha_1\rangle, \ldots, \langle X_n, p_n, \alpha_n\rangle$ where $\{p_1, \ldots, p_n\}$ is a *prima facie* reason for q; and
2. σ's last line is $\langle Y, r, \beta\rangle$ where $Y \subseteq X$ and either:
   (a) r is $\lceil\neg q\rceil$ and $\beta \geq \alpha$; or
   (b) r is $\lceil\neg((p_1 \wedge \ldots \wedge p_n) \gg q)\rceil$ and $\beta \geq \alpha$.

(1) determines the weak spot of η, while (2) determines whether that weak spot is (2a) a conclusion (in this case q), or (2b) a reason (in this case $(p_1 \& \ldots \& p_n) \gg q$). For Pollock, (2a) is a case of *rebutting defeat*, and 2b is a case of *undercutting defeat*: if σ undercuts the last reason of η, it blocks the derivation of q, without supporting $\neg q$ as alternative conclusion. The formula $\lceil\neg((p_1 \wedge \ldots \wedge p_n) \gg q)\rceil$ stands for the translation of the negation of '$\{p_1, \ldots, p_n\}$ is a *prima facie* reason for q' into the object language. Similarly for $\lceil\neg q\rceil$.

Pollock leaves the notion of conflicting arguments implicit in this definition of defeat. Note also that a defeater of an argument always defeats the last step of an argument; Pollock treats 'subargument defeat' by a recursive definition of a justified argument, i.e., in the manner explained above in Section 4.1.

Suppositional reasoning

As noted above, the argument formation rules *supposition*, *conditionalisation* and *dilemma* can be used to form suppositional arguments. OSCAR is one of the very few nonmonotonic logics that allow for suppositional reasoning. Pollock finds it necessary to introduce suppositional reasoning because, in his opinion, this type of reasoning is ubiquitous not only in deductive, but also in defeasible reasoning. Pollock mentions, among other things, the reasoning form 'reasoning by cases', which is notoriously hard for many nonmonotonic logics. An example is 'presumably, birds fly, presumably, bats fly, Tweety is a bird or a bat, so, presumably, Tweety flies'. In Pollock's system, this argument can be formalised as follows.

Consider the following reasons.

(1) $\texttt{Bird}(x)$ is a *prima facie* reason of strength ν for $\texttt{Flies}(x)$
(2) $\texttt{Bat}(x)$ is a *prima facie* reason of strength μ for $\texttt{Flies}(x)$

And consider INPUT = $\{\texttt{Bird}(t) \vee \texttt{Bat}(t)\}$. The conclusion $\texttt{Flies}(t)$ can be defeasibly derived as follows.

1.	$\langle \emptyset, \texttt{Bird}(t) \vee \texttt{Bat}(t), \infty \rangle$	($\texttt{Bird}(t) \vee \texttt{Bat}(t)$ is in INPUT)
2.	$\langle \{\texttt{Bird}(t)\}, \texttt{Bird}(t), \infty \rangle$	(Supposition)
3.	$\langle \{\texttt{Bird}(t)\}, \texttt{Flies}(t), \nu \rangle$	(2 and *prima facie* reason (1))
4.	$\langle \{\texttt{Bat}(t)\}, \texttt{Bat}(t), \infty \rangle$	(Supposition)
5.	$\langle \{\texttt{Bat}(t)\}, \texttt{Flies}(t), \mu \rangle$	(4 and *prima facie* reason (2))
6.	$\langle \emptyset, \texttt{Flies}(t), \min\{\nu, \mu\} \rangle$	(3,5 and Dilemma)

At line 1, the proposition $\texttt{Bird}(t) \vee \texttt{Bat}(t)$ is put forward as an absolute fact. At line (2), the proposition $\texttt{Bird}(t)$ is temporarily supposed to be true. From this assumption, at the following line the conclusion $\texttt{Flies}(t)$ is defeasibly derived with the first *prima facie* reason. Line (4) is an alternative continuation of line 1. At line (4), $\texttt{Bat}(t)$ is supposed to be true, and at line (5) it is used to again defeasibly derive $\texttt{Flies}(t)$, this time from the second *prima facie* reason. Finally, at line (6) the Dilemma rule is applied to (3) and (5), discharging the assumptions in the alternative suppositional arguments, and concluding to $\texttt{Flies}(t)$ under no assumption.

According to Pollock, another virtue of his system is that it validates the defeasible derivation of a material implication from a *prima facie* reason. Consider again the 'birds fly' reason (1), and assume that INPUT is empty.

1.	$\langle \{\texttt{Bird}(t)\}, \texttt{Bird}(t), \infty \rangle$	(Supposition)
2.	$\langle \{\texttt{Bird}(t)\}, \texttt{Flies}(t), \nu \rangle$	(1 and *prima facie* reason (1))
3.	$\langle \emptyset, \texttt{Bird}(t) \supset \texttt{Flies}(t), \nu \rangle$	(2 and Conditionalisation)

Pollock regards the validity of these inferences as desirable. On the other hand, Vreeswijk has argued that suppositional defeasible reasoning, in the way Pollock proposes it, sometimes enables incorrect inferences. Vreeswijk's argument is based on the idea that the strength of a conclusion obtained by means of conditionalisation is incomparable to the reason strength of the implication occurring in that conclusion. For a discussion of this problem the reader is further referred to Vreeswijk [1993a, pp. 184–7].

Having seen how Pollock defines the notions of arguments, conflicting arguments, and defeat among arguments, we now turn to what was the main topic of Section 4 and the main concern of Dung [1995], defining the status of arguments.

The status of arguments

Over the years, Pollock has more than once changed his definition of the status of arguments. One change is that while earlier versions (e.g. Pollock, 1987) dealt with (successful) attack on a subargument in an implicit way via the definition of defeat, the latest version makes this part of the status definition, by explicitly requiring that all subarguments of an 'undefeated' argument are also undefeated

(cf. Section 4.1). Another change is in the form of the status definition. Earlier Pollock took the unique-status-assignment approach, in particular, the fixed-point variant of Definition 16 which, as shown by Dung [1995], (almost) corresponds to the grounded semantics of Definition 7. However, his most recent work is in terms of multiple status assignments, and very similar to the preferred semantics of Definition 39. Pollock's thus combines the recursive style of Definition 17 with the multiple-status-assignments approach. We present the most recent definition, of [Pollock, 1995]. To maintain uniformity in our terminology, we state it in terms of arguments instead of, as Pollock, in terms of an 'inference graph'. To maintain the link with inference graphs, we make the definition relative to a *closed* set of arguments, i.e., a set of arguments containing all subarguments of all its elements. With a (proper) *subargument* of an argument A we mean any argument that is a (proper) subsequence of A.

DEFINITION 45. An assignment of 'defeated' and 'undefeated' to a closed set S of arguments is a *partial defeat status assignment* iff it satisfies the following conditions.

1. All arguments in S with only lines obtained by the *input* argument formation rule are assigned 'undefeated';
2. $A \in S$ is assigned 'undefeated' iff:
   (a) All proper sub-arguments of A are assigned 'undefeated'; and
   (b) All arguments in S defeating A are assigned 'defeated'.
3. $A \in S$ is assigned 'defeated' iff:
   (a) One of A's proper sub-arguments is assigned 'defeated';
   or
   (b) A is defeated by an argument in S that is assigned 'undefeated'.

A *defeat status assignment* is a maximal (with respect to set inclusion) partial defeat status assignment.

Observe that the conditions (2a) and (3a) on the sub-arguments of A make the weakest link principle hold by definition.

The similarity of defeat status assignments to Dung's preferred extensions of Definition 39 shows itself as follows: the conditions (2b) and (3b) on the defeaters of A are the analogues of Dung's notion of acceptability, which make a defeat status assignment an admissible set; then the fact that a defeat status assignment is a maximal partial assignment induces the similarity with preferred extensions.

It is easy to verify that when two arguments defeat each other (Example 3), an input has more than one status assignment. Since Pollock wants to define a sceptical consequence notion, he therefore has to consider the intersection of all assignments. Pollock does so in a variant of Definitions 27 and 21.

DEFINITION 46. (The status of arguments.) Let S be a closed set of arguments based on INPUT. Then, relative to S, an argument is *undefeated* iff every status assignment to S assigns 'undefeated' to it; it is *defeated outright* iff no status assignment to S assigns 'undefeated' to it; otherwise it is *provisionally defeated*.

In our terms, 'undefeated' is 'justified', 'defeated outright' is 'overruled', and 'provisionally defeated' is 'defensible'.

Direct vs. indirect reinstatement

It is now the time to come back to the discussion in Section 4.1 on reinstatement. Example 19 showed that there is reason to invalidate the direct version of this principle, viz. when the conflicts are about the same issue. We remarked that the explicitly recursive Definition 17 of justified arguments indeed invalidates direct reinstatement while preserving its indirect version. However, we also promised to explain that both versions of reinstatement can be retained if Example 19 is represented in a particular way. In fact, Pollock (personal communication) would represent the example as follows:

(1) Being a bird is a prima facie reason for being able to fly
(2a) Being a penguin is an undercutting reason for (1)
(2b) Being a penguin is a defeasible reason for not being able to fly
(3) Being a genetically altered penguin is an undercutting reason for (2b)
(4) Tweety is a genetically altered penguin

It is easy to verify that Definitions 45 and 46, which validate both direct and indirect of reinstatement, yield the intuitive outcome, viz. that it is neither justified that Tweety can fly, nor that it cannot fly. A similar representation is possible in systems that allow for abnormality or exception clauses, e.g. in [Geffner & Pearl, 1992, Bondarenko *et al.*, 1997, Prakken & Sartor, 1997b].

Self-defeating arguments

Pollock has paid much attention to the problem of self-defeating arguments. In Pollock's system, an argument defeats itself iff one of its lines defeats another of its lines. Above in Section 4.1 we already discussed Pollock's treatment of self-defeating arguments within the unique-status-assignment approach. However, he later came to regard this treatment as incorrect, and he now thinks that it can only be solved in the multiple-assignment approach (personal communication).

Let us now see how Pollock's Definitions 45 and 46 deal with the problem. Two cases must be distinguished. Consider first two defeasible arguments A and B rebutting each other. Then A and B are 'parallel' subarguments of a deductive argument $A + B$ for any proposition. Then (if no other arguments interfere with A or B) there are two status assignments, one in which A is assigned 'undefeated' and B assigned 'defeated', and one the other way around. Now $A + B$ is in both of these assignments assigned 'defeated', since in both assignments one of its proper

subarguments is assigned 'defeated'. Thus the self-defeating argument $A+B$ turns out to be defeated outright, which seems intuitively plausible.

A different case is the following, with the following reasons

(1) p is a *prima facie* reason of strength 0.8 for q
(2) q is a *prima facie* reason of strength 0.8 for r
(3) r is a conclusive reason for $\lceil\neg(p \gg q)\rceil$

and with INPUT = $\{p\}$. The following (linear) argument can be constructed.

1.	$\langle p, \infty\rangle$	(p is in INPUT)
2.	$\langle q, 0.8\rangle$	(1 and *prima facie* reason (1))
3.	$\langle r, 0.8\rangle$	(2 and *prima facie* reason (2))
4.	$\langle\neg(p \gg q), 0.8\rangle$	(3 and conclusive reason (3))

Let us call this argument A, with proper subarguments A_1, A_2, A_3 and A_4, respectively. Observe first that, according to Pollock's definition of self-defeat, A_4 is self-defeating. Further, according to Pollock's earlier approach with Definition 16, A_4 is, as being self-defeating, overruled, or 'defeated', while A_1, A_2 and A_3 are justified, or 'undefeated'. Pollock now regards this outcome as incorrect: since A_4 is a deductive consequence of A_3, A_3 should also be 'defeated'.

This result is obtained with Definitions 45 and 46. Firstly, A_1 is clearly undefeated. Consider next A_2. This argument is undercut by A_4, so if A_4 is assigned 'undefeated', then A_2 must be assigned 'defeated'. But then A_4 must also be assigned 'defeated', since one of its proper subarguments is assigned 'defeated'. Contradiction. If, on the other hand, A_4 is assigned 'defeated', then A_2 and so A_3 must be assigned 'undefeated'. But then A_4 must be assigned 'undefeated'. Contradiction. In conclusion, no partial status assignment will assign a status to A_4 and, consequently, no status assignment will assign a status to A_2 or A_3 either. And since this implies that no status assignment assigns the status 'undefeated' to any of these arguments, they are by Definition 46 all defeated outright.

Two remarks about this outcome can be made. Firstly, it might be doubted whether A_2 should indeed be defeated outright, i.e., overruled. It is not self-defeating, its only defeater is self-defeating, and this defeater is not a deductive consequence of A_2's conclusion. Other systems, e.g. those of Vreeswijk (Section 5.5) and Prakken & Sartor (Section 5.7), regard A_2 as justified. In these systems Pollock's intuition about A_3 is formalised by regarding A_3 as self-defeating because its conclusion deductively, not just defeasibly, implies a conclusion incompatible with itself. This makes it possible to regard A_3 as overruled but A_2 as justified.

Furthermore, even if Pollock's outcome is accepted, the situation is not quite the same as with the previous example. Consider another defeasible argument B which rebuts and is rebutted by A_3. Then no assignment assigns a status to B either, for which reason B is also defeated outright. Yet this shows that the 'defeated outright' status of A_2 is not the same as the 'defeated outright' status

of an argument that has an undefeated defeater: apparently, A_2 is still capable of preventing other arguments from being undefeated. In fact, the same holds for arguments involved in an odd defeat cycle (as in Example 30).

In conclusion, Pollock's definitions leave room for a fourth status of arguments, which might be called 'seemingly defeated'. This status holds for arguments that according to Definition 46 are defeated outright but still have the power to prevent other arguments from being ultimately undefeated. The four statuses can be partially ordered as follows: 'undefeated' is better than 'provisionally defeated' and than 'seemingly defeated', which both in turn are better than 'defeated outright'. This observation applies not only to Pollock's definition, but to all approaches

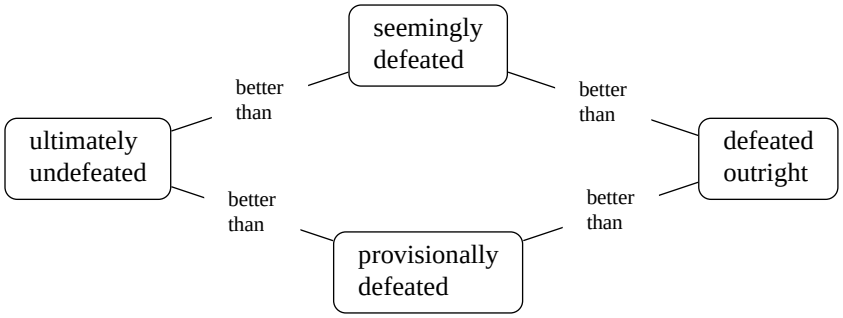
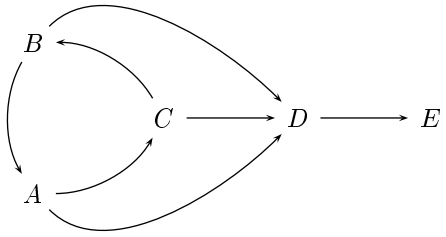


Figure 8. Partial ordering of defeat statuses.

based on partial status assignments, like Dung [1995] preferred semantics.

However, this is not yet all: even if the notion of seeming defeat is made explicit, there still is an issue concerning floating arguments (cf. Example 24). To see this, consider the following extension of Example 30 (formulated in terms of [Dung, 1995]).

EXAMPLE 47. Let A, B and C be three arguments, represented in a triangle, such that A defeats C, B defeats A, and C defeats B. Furthermore, let D and E be arguments such that all of A, B and C defeat D, and D defeats E.



The difference between Example 24 and this example is that the even defeat loop between two arguments is replaced by an odd defeat loop between three arguments. One view on the new example is that this difference is inessential and that, for the same reasons as why in Example 24 the argument D is justified, here the argument E is ultimately undefeated: although E is strictly defeated by D, it is reinstated by all of A, B and C, since all these arguments strictly defeat D. On this account Definitions 45 and 46 are flawed since they render all five arguments defeated outright (and in our terms seemingly defeated). However, an alternative view is that odd defeat loops are of an essentially different kind than even defeat loops, so that our analysis of Example 24 does not apply here and that the outcome in Pollock's system reflects a flaw in the available input information rather than in the system.

Ideal and resource-bounded reasoning

We shall now see that Definition 46 is not yet all that Pollock has to say on the status of arguments. In the previous section we saw that the BDKT approach leaves the origin of the set of 'input' arguments unspecified. At this point Pollock develops some interesting ideas. At first sight it might be thought that the set S of the just-given definitions is just the set of all arguments that can be constructed with the argument formation rules of Definition 44. However, this is only one of the possibilities that Pollock considers, in which Definition 46 captures so-called *ideal warrant*.

DEFINITION 48. (Ideal warrant.) Let S be the set of all arguments based on INPUT. Then an argument A is *ideally warranted* relative to INPUT iff A is undefeated relative to S.

Pollock wants to respect that in actual reasoning the construction of arguments takes time, and that reasoners have no infinite amount of time available. Therefore, he also considers two other definitions, both of which have a computational flavour. To capture an actual reasoning process, Pollock makes them relative to a sequence $\mathcal{S}$ of closed finite sets $S_0 \subseteq \ldots \subseteq S_i \ldots$ of arguments. Let us call this an *argumentation sequence*. Such a sequence contains all arguments constructed by a reasoner, in the order in which they are produced. It (and any of its elements) is based on INPUT if all its arguments are based on INPUT. [13]

Now the first 'computational' status definition determines what a reasoner must believe at any given time.

DEFINITION 49. (Justification.) Let $\mathcal{S}$ be an argumentation sequence based on INPUT, and S_i an element of $\mathcal{S}$. Then an argument A is *justified* relative to INPUT at stage i iff A is undefeated relative to S_i.

In this definition the set S_i contains just those arguments that have actually been constructed by a reasoner. Thus this definition captures the *current* status of

[13] Note that we again translate Pollock's inference graphs into (structured) sets of arguments.

a belief; it may be that further reasoning (without adding new premises) changes the status of a conclusion.

This cannot happen for the other 'computational' consequence notion defined by Pollock, called *warrant*. Intuitively, an argument A is warranted iff eventually in an argumentation sequence a stage is reached where A remains justified at every subsequent stage. To define this, the notion of a 'maximal' argumentation sequence is needed, i.e., a sequence that cannot be extended. Thus it contains all arguments that a reasoner with unlimited resources would construct (in a particular order).

DEFINITION 50. (Warrant.) Let $\mathcal{S}$ be a maximal argumentation sequence $S_0 \subseteq \ldots \subseteq S_i \ldots$ based on INPUT. Then an argument A is *warranted* (relative to INPUT) iff there is an i such that for all $j > i$, A is undefeated relative to S_j.

The difference between warrant and ideal warrant is subtle: it has to do with the fact that, while in determining warrant every set $S_j \supseteq S_i$ that is considered is *finite*, in determining ideal warrant the set of all possible arguments has to be considered, and this set can be infinite.

EXAMPLE 51. (Warrant does not entail ideal warrant.) Suppose $A_1, A_2, A_3, \ldots$ are arguments such that every A_i is defeated by its successor A_{i+1}. Further, suppose that the arguments are produced in the order $A_2, A_1, A_4, A_3, A_6, A_5, A_8, \ldots$ Then

Stage	Produced	Justified
1	A_2	A_2
2	A_2, A_1	A_2
3	A_2, A_1, A_4	A_2, A_4
4	A_2, A_1, A_4, A_3	A_2, A_4
5	A_2, A_1, A_4, A_3, A_6	A_2, A_4, A_6
6	$A_2, A_1, A_4, A_3, A_6, A_5$	A_2, A_4, A_6
7	$A_2, A_1, A_4, A_3, A_6, A_5, A_8$	A_2, A_4, A_6, A_8
$\vdots$	$\vdots$	$\vdots$

From stage 1, A_2 is justified and stays justified. Thus, A_2 is warranted. At the same time, however, A_2 is not ideally warranted, because there exist two status assignments for all A_i's. One assignment in which all and only all odd arguments are 'in', and one assignment in which all and only all odd arguments are 'out'. Hence, according to ideal warrant, every argument is only provisionally defeated. In particular, A_2 is provisionally defeated. A remarkable aspect of this example is that, eventually, every argument will be produced, but without reaching the right result for A_2.

EXAMPLE 52. (Ideal warrant does not imply warrant.) Suppose that A, B_1, B_2, B_3, ... and C_1, C_2, C_3, ... are arguments such that A is defeated by every B_i, and every B_i is defeated by C_i. Further, suppose that the arguments are produced in the order $A, B_1, C_1, B_2, C_2, B_3, C_3, \ldots$ Then

Stage	Produced	Justified
1	A	A
2	A, B_1	B_1
3	A, B_1, C_1	A, C_1
4	A, B_1, C_1, B_2	C_1, B_2
5	A, B_1, C_1, B_2, C_2	A, C_1, C_2
6	$A, B_1, C_1, B_2, C_2, B_3$	C_1, C_2, B_3
$\vdots$	$\vdots$	$\vdots$

Thus, in this sequence, A is provisionally defeated. However, according to the definition of ideal warrant, every B_i is defeated by C_i, so that A remains undefeated.

Although the notion of warrant is computationally inspired, as Pollock observes there is no automated procedure that can determine of any warranted argument that it is warranted: even if in fact a warranted argument stays undefeated after some finite number n of computations, a reasoner can in state n not *know* whether it has reached a point where the argument stays undefeated, or whether further computation will change its status.

Pollock's reasoning architecture

We now discuss Pollock's reasoning architecture for computing the ideally warranted propositions, i.e. the propositions that are the conclusion of an ideally warranted argument. (According to Pollock, ideal warrant is what every reasoner should ultimately strive for.) In deductive logic such an architecture would be called a 'proof theory', but Pollock rejects this term. The reason is that one condition normally required of proof theories, viz. that the set of theorems is recursively enumerable, cannot in general be satisfied for a defeasible reasoner. Pollock assumes that a reasoner reasons by constantly updating its beliefs, where an update is an elementary transition from one set of propositions to the next set of propositions. According to this view, a reasoner would be adequate if the resulting sequence is a recursively enumerable approximation of ideal warrant. However, this is impossible. Ideal warrant contains all theorems of predicate logic, and it is known that all theorems of predicate logic form a set that is not recursive. And since in defeasible reasoning some conclusions depend on the failure to derive other conclusions, the set of defeasible conclusions is not recursively enumerable. Therefore, Pollock suggests an alternative criterion of adequacy. A reasoner is called *defeasibly adequate* if the resulting sequence is a defeasibly enumerable approximation of ideal warrant.

DEFINITION 53. A set A is *defeasibly enumerable* if there is a sequence of sets $\{A_i\}_{1 \leq i}$ such that for all x

1. If $x \in A$, then there is an N such that $x \in A_i$ for all $i > N$.
2. If $x \notin A$, then there is an M such that $x \notin A_i$ for all $i > M$.

If A is recursively enumerable, then a reasoner who updates his beliefs in Pollock's way can approach A 'from below': the reasoner can construct sets that are all supersets of the preceding set and subsets of A. However, when A is only defeasibly enumerable, a reasoner can only approach A from below and above simultaneously, in the sense that the sets A_i the reasoner constructs may contain elements not contained in A. Every such element must eventually be taken out of the A_i's, but there need not be any point at which they have *all* been removed.

To ensure defeasible adequacy, Pollock introduces the following three operations:

1. The reasoner must adopt beliefs in response to constructing arguments, provided no counterarguments have already been adopted for any step in the argument. If a defeasible inference occurs, a check must be made whether a counterargument for it has not already been adopted as a belief.

2. The reasoner must keep track of the bases upon which its beliefs are held. When a new belief is adopted that is a defeater for a previous inference step, then the reasoner must retract that inference step and all beliefs inferred from it.

3. The reasoner must keep track of defeated inferences, and when a defeater is itself retracted (2), this should reinstate the defeated inference.

To achieve the functions just described, Pollock introduces a so-called *flag-based* reasoner. A flag-based reasoner consists of an inference engine that produces all arguments eventually, and a component computing the defeat status of arguments.

```
LOOP    BEGIN
                make-an-inference
                recompute-defeat-statuses
        END
```

The procedure recompute-defeat-statuses determines which arguments are defeated outright, undefeated and provisionally defeated at each iteration of the loop. That is, at each iteration it determines justification.

Pollock then identifies certain conditions under which a flag-based reasoner is defeasibly adequate. For these conditions, the reader is referred to [Pollock, 1995, ch. 4].

Evaluation

Pollock's theory of defeasible reasoning is based on more than thirty years of research in logic and epistemology. This large time span perhaps explains the richness of his theory. It includes both linear and suppositional arguments, and deductive as well as non-deductive (mainly statistical and inductive) arguments, with a

corresponding distinction between two types of conflicts between arguments. Pollock's definition of the status of arguments takes the multiple-status-assignments approach, being related to Dung's preferred semantics. This semantics can deal with certain types of floating statuses and conclusions, but we have seen that certain other types are still ignored. In fact, this seems one of the main unsolved problems in argument-based semantics. An interesting aspect of Pollock's work is his study of the resource-bounded nature of practical reasoning, with the idea of partial computation embodied in the notions of warrant and especially justification. And for artificial intelligence it is interesting that Pollock has implemented his system as a computer program.

Since Pollock focuses on epistemological issues, his system is not immediately applicable to some specific features of practical (including legal) reasoning. For instance, the use of probabilistic notions seems to make it difficult to give an account of reasoning with and about priority relations between arguments (see below in Subsection 5.7). Moreover, it would be interesting to know what Pollock would regard as suitable reasons for normative reasoning. It would also be interesting to study how, for instance, analogical and abductive arguments can be analysed in Pollock's system as giving rise to *prima facie* reasons.

5.3 Inheritance systems

A forerunner of argumentation systems is work on so-called inheritance systems, especially of Horty *et al.*, e.g. [1990], which we shall briefly discuss. Inheritance systems determine whether an object of a certain kind has a certain property. Their language is very restricted. The network is a directed graph. Its initial nodes represent individuals and its other nodes stand for classes of individuals. There are two kinds of links, $\rightarrow$ and $\not\rightarrow$, depending on whether something does or does not belong to a certain class. Links from an individual to a class express class membership, and links between two classes express class inclusion.

A path through the graph is an *inheritance path* iff its only negative link is the last one. Thus the following are examples of inheritance paths.

P_1: $\texttt{Tweety} \rightarrow \texttt{Penguin} \rightarrow \texttt{Bird} \rightarrow \texttt{Canfly}$
P_2: $\texttt{Tweety} \rightarrow \texttt{Penguin} \not\rightarrow \texttt{Canfly}$

Another basic notion is that of an *assertion*, which is of the form $x \rightarrow y$ or $x \not\rightarrow y$, where y is a class. Such an assertion is *enabled* by an inheritance path if the path starts with x and ends with the same link to y as the assertion. Above, an assertion enabled by P_1 is $\texttt{Tweety} \rightarrow \texttt{Canfly}$, and an assertion enabled by P_2 is $\texttt{Tweety} \not\rightarrow \texttt{Canfly}$.

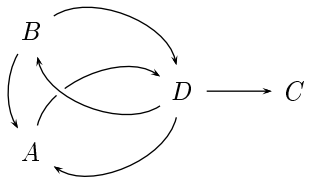
As the example shows, two paths can be conflicting. They are compared on specificity, which is read off from the syntactic structure of the net, resulting in relations of *neutralisation* and *preemption* between paths. The assignment of a status to a path (whether it is *permitted*) is similar to the recursive variant of the unique-status-assignment approach of Definition 17. This means that the system

has problems with Zombie paths and floating conclusions (as observed by Makinson & Schlechta [1991]).

Although Horty *et al.* present their system as a special-purpose formalism, it clearly has all the elements of an argumentation system. An inheritance path corresponds to an argument, and an assertion enabled by a path to a conclusion of an argument. Their notion of conflicting paths corresponds to rebutting attack. Furthermore, neutralisation and preemption correspond to defeat, while a permitted path is the same as a justified argument.

Because of the restricted language and the rather complex definition of when an inheritance path is permitted, we shall not present the full system. However, Horty *et al.* should be credited for anticipating many distinctions and discussions in the field of defeasible argumentation. In particular, their work is a rich source of benchmark examples. We shall discuss one of them.

EXAMPLE 54. 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 D defeat each other.



Here is a natural-language version (due to Horty, personal communication), in which the defeat relations are based on specificity considerations.

- $A =$ 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 Brentwood, and people who live in Brentwood are rich;
- $D =$ Larry is not rich because he rents in Brentwood, and people who rent in Brentwood are not rich.

If we apply the various semantics of the BDKT approach to this example, we see that since no argument is undefeated, none of them is in the grounded extension. Moreover, there are preferred extensions in which Larry is rich, and preferred extensions in which Larry is not rich. Yet it might be argued that since both arguments that Larry is rich are strictly defeated by an argument that Larry is not rich, the sceptical conclusion should be that Larry is not rich. This is the outcome obtained by Horty *et al.*

5.4 Lin and Shoham

Before the BDKT approach, an earlier attempt to provide a unifying framework for nonmonotonic logics was made by Lin & Shoham [1989]. They show how any logic, whether monotonic or not, can be reformulated as a system for constructing arguments. However, in contrast with the other theories in this section, they are not concerned with comparing incompatible arguments, and so their framework cannot be used as a theory of defeat among arguments.

The basic elements of Lin & Shoham's abstract framework are an unspecified logical language, only assumed to contain a negation symbol, and an also unspecified set of inference rules defined over the assumed language. Arguments can be constructed by chaining inference rules into trees.

Inference rules are either monotonic or nonmonotonic. For instance,

$$\texttt{Penguin}(a) \rightarrow \texttt{Bird}(a)$$
$$\texttt{Penguin}(a), \neg\texttt{ab}(\texttt{penguin}(a)) \rightarrow \neg\texttt{Fly}(a)$$

are monotonic rules, and

$$\texttt{True} \Rightarrow \neg\texttt{ab}(\texttt{penguin}(a))$$
$$\texttt{True} \Rightarrow \neg\texttt{ab}(\texttt{bird}(a))$$

are nonmonotonic rules. Note that these inference rules are, as in default logic, domain specific. In fact, Lin & Shoham do not distinguish between general and domain-dependent inference rules, as is shown by their reconstruction of default logic, to be discussed below.

Although the lack of a notion of defeat is a severe limitation, in capturing nonmonotonic consequence Lin & Shoham introduce a notion which for defeasible argumentation is very relevant viz. that of an *argument structure*.

DEFINITION 55. (argument structures) A set T of arguments is an *argument structure* if T satisfies the following conditions:

1. The set of 'base facts' (which roughly are the premises) is in T;
2. Of every argument in T all its subarguments are in T;
3. The set of conclusions of arguments in T is deductively closed and consistent.

Note that the notion of a 'closed' set of arguments that we used above in Pollock's Definition 45 satisfies the first two but not the third of these conditions. Note also that, although argument structures are closed under monotonic rules, they are not closed under defeasible rules.

Lin & Shoham then reformulate existing nonmonotonic logics in terms of monotonic and nonmonotonic inference rules, and show how the alternative sets of conclusions of these logics can be captured in terms of argument structures with certain completeness properties. Bondarenko *et al.* [1997] remark that structures with these properties are very similar to their stable extensions.

The claim that existing nonmonotonic logics can be captured by an argument system is an important one, and Lin & Shoham were among the first to make it. The remainder of this section is therefore devoted to showing with an example how Lin & Shoham accomplish this, viz. for default logic [Reiter, 1980].

In default logic (see also Subsection 2.1), a *default theory* is a pair $\Delta = (W, D)$, where W is a set of first-order formulas, and D a set of defaults. Each default is of the form $A : B_1, \ldots, B_n/C$, where A, B_i and C are first-order formulas. Informally, a default reads as 'If A is known, and $B_1, \ldots, B_n$ are consistent with what is known, then C may be inferred'. An *extension* of a default theory is any set of formulas E satisfying the following conditions. $E = \cup_{i=0}^{\infty}$, where

$$\begin{array}{rcl} E_0 & = & W, \\ E_{i+1} & = & Th(E_i) \cup \{C \mid A : B_1, \ldots, B_n/C \in D \\ & & \text{where } A \in E_i \text{ and } \neg B_1, \ldots \neg B_n \notin E\} \end{array}$$

We now discuss the correspondence between default logic and argument systems by providing a global outline of the translation and proof. Lin & Shoham perform the translation as follows. Let $\Delta = (W, D)$ be a closed default theory. Define $R(\Delta)$ to be the set of the following rules:

1. `True` is a base fact.
2. If $A \in W$, then A is a base fact of $R(\Delta)$.
3. If $A_1, \ldots, A_n$, and B are first-order sentences and B is a consequence of $A_1, \ldots, A_n$ in first-order logic, then $A_1, \ldots, A_n \rightarrow B$ is a monotonic rule.
4. If A is a first-order sentence, then $\neg A \rightarrow \mathtt{ab}(A)$ is a monotonic rule.
5. If $A : B_1, \ldots, B_n/C$ is a default in D, then

   $$A, \neg\mathtt{ab}(B_1), \ldots, \neg\mathtt{ab}(B_n) \rightarrow C$$

   is a monotonic rule.
6. If B is a first-order sentence, then $\mathtt{True} \Rightarrow \neg\mathtt{ab}(B)$ is a nonmonotonic rule.

Lin & Shoham proceed by introducing the concept of DL-complete argument structures.

DEFINITION 56. An argument structure T of $R(\Delta)$ is said to be *DL-complete* if for any first-order sentence A, either $\mathtt{ab}(A)$ or $\neg\mathtt{ab}(A)$ is in Wff(T).

Thus, a DL-complete argument structure is explicit about the abnormality of every first-order sentence. For DL-complete argument structures, the following lemma is established.

LEMMA 57. *If T is a DL-complete argument structure of $R(\Delta)$, then for any first-order sentence A, $\mathtt{ab}(A) \in$ Wff(T) iff $\neg A \in$ Wff(T).*

On the basis of this result, Lin & Shoham are able to establish the following correspondence between default logic and argument systems.

THEOREM 58. *Let E be a consistent set of first-order sentences. E is an extension of Δ iff there is a DL-complete argument structure T of $R(\Delta)$ such that E is the restriction of Wff(T) to the set of first-order sentences.*

This theorem is proven by constructing extensions for given argument structures and *vice versa*. If E is an extension of Δ, Lin & Shoham define T as the set of arguments with all nodes in E', where

$$E' = E \cup \{\mathtt{ab}(B) \mid \neg B \in E\} \cup \{\neg\mathtt{ab}(B) \mid \neg B \notin E\}$$

and prove that *Wff*$(T) = E'$. Conversely, for a DL-complete argument structure T of $R(\Delta)$, Lin & Shoham prove that the first-order restriction E of *Wff*(T) is a default extension of Δ. This is proven by induction on the definition of an extension.

Two features in the translation are worth noticing. First, default logic makes a distinction between meta-logic default rules and first-order logic, while argument systems do not. Second, the notion of groundedness of default extensions corresponds to that of an argument in argument systems, and the notion of fixed points in default logic corresponds to that of DL-completeness of argument structures.

Lin & Shoham further show that, for normal default theories, the translation can be performed without second-order predicates, such as `ab`. This result however falls beyond the scope of this chapter.

5.5 *Vreeswijk's Abstract Argumentation Systems*

Like the BDKT approach and Lin & Shoham [1989], Vreeswijk [1993a, 1997] also aims to provide an abstract framework for defeasible argumentation. His framework builds on the one of Lin & Shoham, but contains the main elements that are missing in their system, namely, notions of conflict and defeat between arguments. As Lin & Shoham, Vreeswijk also assumes an unspecified logical language $\mathcal{L}$, only assumed to contain the symbol $\bot$, denoting 'falsum' or 'contradiction,' and an unspecified set of monotonic and nonmonotonic inference rules (which Vreeswijk calls 'strict' and 'defeasible'). This also makes his system an abstract framework rather than a particular system. A point in which Vreeswijk's work differs from Lin & Shoham is that Vreeswijk's inference rules are not domain specific but general logical principles.

DEFINITION 59. (Rule of inference.) Let $\mathcal{L}$ be a language.

1. A *strict rule of inference* is a formula of the form $\phi_1, \ldots, \phi_n \rightarrow \phi$ where $\phi_1, \ldots, \phi_n$ is a finite, possibly empty, sequence in $\mathcal{L}$ and ϕ is a member of $\mathcal{L}$.

2. A *defeasible rule of inference* is a formula of the form $\phi_1, \ldots, \phi_n \Rightarrow \phi$ where $\phi_1, \ldots, \phi_n$ is a finite, possibly empty, sequence in $\mathcal{L}$ and ϕ is a member of $\mathcal{L}$.

A *rule of inference* is a strict or a defeasible rule of inference.

Another aspect taken from Lin & Shoham is that in Vreeswijk's framework, arguments can also be formed by chaining inference rules into trees.

DEFINITION 60. (Argument.) Let R be a set of rules. An argument has *premises*, a *conclusion*, *sentences* (or propositions), *assumptions*, *subarguments*, *top arguments*, a *length*, and a *size*. These are abbreviated by corresponding prefixes. An *argument* σ is

1. A member of $\mathcal{L}$; in that case,

   $$prem(\sigma) = \{\sigma\},\ conc(\sigma) = \sigma,\ sent(\sigma) = \{\sigma\},\ asm(\sigma) = \emptyset,$$
   $$sub(\sigma) = \{\sigma\}, top(\sigma) = \{\sigma\}, length(\sigma) = 1, \text{and } size(\sigma) = 1;$$

   or

2. A formula of the form $\sigma_1, \ldots, \sigma_n \rightarrow \phi$ where $\sigma_1, \ldots, \sigma_n$ is a finite, possibly empty, sequence of arguments, such that $conc(\sigma_1) = \phi_1, \ldots, conc(\sigma_n) = \phi_n$ for some rule $\phi_1, \ldots, \phi_n \rightarrow \phi$ in R, and $\phi \notin sent(\sigma_1) \cup \ldots \cup sent(\sigma_n)$— in that case,

   $$prem(\sigma) = prem(\sigma_1) \cup \ldots \cup prem(\sigma_n),$$
   $$conc(\sigma) = \phi,$$
   $$sent(\sigma) = sent(\sigma_1) \cup \ldots \cup sent(\sigma_n) \cup \{\phi\},$$
   $$asm(\sigma) = asm(\sigma_1) \cup \ldots \cup asm(\sigma_n),$$
   $$sub(\sigma) = sub(\sigma_1) \cup \ldots \cup sub(\sigma_n) \cup \{\sigma\},$$
   $$top(\sigma) = \{\tau_1, \ldots, \tau_n \rightarrow \phi \mid \tau_1 \in top(\sigma_1), \ldots, \tau_n \in top(\sigma_n)\} \cup \{\phi\},$$
   $$length(\sigma) = max\{length(\sigma_1), \ldots, length(\sigma_n)\} + 1, \text{ and}$$
   $$size(\sigma) = size(\sigma_1) + \ldots + size(\sigma_n) + 1;$$

   or

3. A formula of the form $\sigma_1, \ldots, \sigma_n \Rightarrow \phi$ where $\sigma_1, \ldots, \sigma_n$ is a finite, possibly empty, sequence of arguments, such that $conc(\sigma_1) = \phi_1, \ldots, conc(\sigma_n) = \phi_n$ for some rule $\phi_1, \ldots, \phi_n \Rightarrow \phi$ in R, and $\phi \notin sent(\sigma_1) \cup \ldots \cup sent(\sigma_n)$; for assumptions we have

   $$asm(\sigma) = asm(\sigma_1) \cup \ldots \cup asm(\sigma_n) \cup \{\phi\};$$

   premises, conclusions, and other attributes are defined as in (2).

Arguments of type (1) are *atomic* arguments; arguments of type (2) and (3) are *composite* arguments. Thus, atomic arguments are language elements. An argument σ is said to be *in contradiction* if $conc(\sigma) = \bot$. An argument is *defeasible* if it contains at least one defeasible rule of inference; else it is *strict*.

Unlike Lin & Shoham, Vreeswijk assumes an ordering on arguments, indicating their difference in strength (on which more below).

As for conflicts between arguments, a difference from all other systems of this section (except [Verheij, 1996]; see below in subsection 5.10) is that a counter-argument is in fact a *set* of arguments: Vreeswijk defines a set Σ of arguments *incompatible* with an argument τ iff the conclusions of $\Sigma \cup \{\tau\}$ give rise to a strict argument for $\bot$. Sets of arguments are needed because the language in Vreeswijk's framework is unspecified and therefore lacks the expressive power to 'recognise' inconsistency. The consequence of this lack of expressiveness is that a set of arguments $\sigma_1, \ldots, \sigma_n$ that is incompatible with τ, cannot be joined to one argument σ that contradicts, or is inconsistent, with τ. Therefore, it is necessary to take sets of arguments into account.

Vreeswijk has no explicit notion of undercutting attacks; he claims that this notion is implicitly captured by his notion of incompatibility, viz. as arguments for the denial of a defeasible conditional used by another argument. This requires some extra assumptions on the language of an abstract argumentation system, viz. that it is closed under negation ($\neg$), conjunction ($\wedge$), material implication ($\supset$), and defeasible implication ($>$). For the latter connective Vreeswijk defines the following defeasible inference rule.

$\frac{\varphi,\ \varphi > \psi}{\psi}$ defeasible rule of inference

With these extra language elements, it is possible to express rules of inference (which are meta-linguistic notions) in the object language. Meta-level rules using $\rightarrow$ (strict rule of inference) and $\Rightarrow$ (defeasible rule of inference) are then represented by corresponding object language implication symbols $\supset$ and $>$. Under this condition, Vreeswijk claims to be able to define rebutting and undercutting attackers in a formal fashion. For example, let σ and τ be arguments in Vreeswijk's system with conclusions φ and ψ, respectively. Let $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ be the top rule of σ.

Rebutting attack. If $\psi = \neg\varphi$, then Vreeswijk calls τ a *rebutting* attacker of σ. Thus, the conclusion of a rebutting attacker contradicts the conclusion of the argument it attacks.

Undercutting attack. If $\psi = \neg(\varphi_1 \wedge \ldots \wedge \varphi_n > \varphi)$, i.e. if ψ is the negation of the last rule of σ stated in the object language, then τ is said to be an *undercutting* attacker of σ. Thus, the conclusion of an undercutting attacker contradicts the last inference of the argument it attacks.

Vreeswijk's notion of defeat rests on two basic concepts, viz. the above-defined notion of incompatibility and the notion of undermining. An argument is said to *undermine* a set of arguments, if it dominates at least one element of that set. Formally, a set of arguments Σ is *undermined* by an argument τ if $\sigma < \tau$ for some $\sigma \in \Sigma$. If a set of arguments is undermined by another argument, it cannot uphold or maintain all of its members in case of a conflict.

Vreeswijk then defines the notion of a defeater as follows:

DEFINITION 61. (Defeater.) Let P be a base set, and let σ be an argument. A set of arguments Σ is a *defeater* of σ if it is incompatible with σ and not undermined by it; in this case σ is said to be *defeated* by Σ, and Σ *defeats* σ. Σ is a *minimal defeater* of σ if all its proper subsets do not defeat σ.

As for the assessment of arguments, Vreeswijk's declarative definition, (which he says is about "warrant") is similar to Pollock's definition of a defeat status assignment: both definitions have an explicit recursive structure and both lead to multiple status assignments in case of irresolvable conflicts. However, Vreeswijk's status assignments cannot be partial, for which reason Vreeswijk's definition is closer to stable semantics than to preferred semantics.

DEFINITION 62. (Defeasible entailment.) Let P be a base set. A relation $\mid\!\sim$ between P and arguments based on P is a *defeasible entailment relation* if, for every argument σ based on P, we have $P \mid\!\sim \sigma$ *(σ is in force on the basis of P)* if and only if

1. The set P contains σ; or
2. For some arguments $\sigma_1, \ldots, \sigma_n$ we have $P \mid\!\sim \sigma_1, \ldots, \sigma_n$ and $\sigma_1, \ldots, \sigma_n \rightarrow \sigma$; or
3. For some arguments $\sigma_1, \ldots, \sigma_n$ we have $P \mid\!\sim \sigma_1, \ldots, \sigma_n$ and $\sigma_1, \ldots, \sigma_n \Rightarrow \sigma$ and every set of arguments Σ such that $P \mid\!\sim \Sigma$ does not defeat σ.

In the Nixon Diamond of Example 3 this results in 'the Quaker argument is in force iff the Republican argument is not in force'. To deal with such circularities Vreeswijk defines for every $\mid\!\sim$ satisfying the above definition an extension

$$\Sigma = \{\phi \mid P \mid\!\sim \phi\} \tag{1}$$

On the basis of Definition 62 it can be proven that (1) is stable, i.e., it can be proven that $\phi \notin \Sigma$ iff Σ' defeats ϕ for some $\Sigma' \subseteq \Sigma$. With equally strong conflicting arguments, as in the Nixon Diamond, this results in multiple stable extensions (cf. Definition 38).

Just as in Dung's stable semantics, in Vreeswijk's system examples with odd defeat loops have no extensions. However, an exception holds for the special case of self-defeating arguments, since Definition 61 implies that every argument of which the conclusion strictly implies $\bot$ is defeated by the empty set.

Argumentation sequences

Vreeswijk extensively studies various other characterisations of defeasible argumentation. Among other things, he develops the notion of an 'argumentation sequence'. An argumentation sequence can be regarded as a sequence

$$\Sigma_1 \longrightarrow \Sigma_2 \longrightarrow \ldots \longrightarrow \Sigma_n \longrightarrow \ldots$$

of Lin & Shoham's [1989] argument structures, but without the condition that these structures are closed under deduction. Each following structure is constructed by applying an inference rule to the arguments in the preceding structure. An important addition to Lin & Shoham's notion is that a newly constructed argument is only appended to the sequence if it survives all counterattacks from the argument structure developed thus far. Thus the notion of an argumentation sequence embodies, like Pollock's notion of 'justification', the idea of partial computation, i.e., of assessing arguments relative to the inferences made so far. Vreeswijk's argumentation sequences also resemble BDKT's procedure for computing admissible semantics. The difference is that BDKT adopt arguments that are defended (admissible semantics), while Vreeswijk argumentation sequences adopt arguments that are not defeated (stable semantics).

Vreeswijk also develops a procedural version of his framework in dialectical style. It will be discussed below in Section 6.

Plausible reasoning

Vreeswijk further discusses a distinction between two kinds of nonmonotonic reasoning, 'defeasible' and 'plausible' reasoning. According to him, the above definition of defeasible entailment captures defeasible reasoning, which is unsound (i.e., defeasible) reasoning from firm premises, like in 'typically birds fly, Tweety is a bird, so presumably Tweety flies'. Plausible reasoning, by contrast, is sound (i.e., deductive) reasoning from uncertain premises, as in 'all birds fly (we think), Tweety is a bird, so Tweety flies (we think)' [Rescher, 1976]. The difference is that in the first case a default proposition is accepted categorically, while in the second case a categorical proposition is accepted by default. In fact, Vreeswijk would regard reasoning with ordered premises, as studied in many nonmonotonic logics, not as defeasible but as plausible reasoning.

One element of this distinction is that for defeasible reasoning the ordering on arguments is not part of the input theory, reflecting priority relations between, or degrees of belief in premises, but a general ordering of *types* of arguments, such as 'deductive arguments prevail over inductive arguments' and 'statistical inductive arguments prevail over generic inductive arguments'. Accordingly, Vreeswijk assumes that the ordering on arguments is the same for all sets of premises (although relative to a set of inference rules). Vreeswijk formalises plausible reasoning independent of defeasible reasoning, with the possibility to define input orderings on the premises, and he then combines the two formal treatments. To our knowledge, Vreeswijk's framework is unique in treating these two types of reasoning in one formalism as distinct forms of reasoning; usually the two forms are regarded as alternative ways to look at the same kind of reasoning.

Evaluating Vreeswijk's framework, we can say that it has little attention for the details of comparing arguments and that, as Pollock but in contrast to BDKT, it formalises only one type of defeasible consequence, but that it is philosophically well-motivated, and quite detailed with respect to the structure of arguments and

the process of argumentation.

5.6 Simari & Loui

Simari & Loui [1992] present a declarative system for defeasible argumentation that combines ideas of Pollock [1987] on the interaction of arguments with ideas of Poole [1985] on specificity and ideas of Loui [1987] on defaults as twoplace meta-linguistic rules. Simari & Loui divide the premises into sets of contingent first-order formulas $\mathcal{K}_C$, and necessary first-order formulas $\mathcal{K}_N$, and one-directional default rules Δ, e.g.

$$\begin{array}{rcl} \mathcal{K}_C & = & \{P(a)\} \\ \mathcal{K}_N & = & \{\forall x.P(x) \supset B(x)\} \\ \Delta & = & \{B(x) \succ\!\!- F(x),\ P(x) \succ\!\!- \neg F(x)\}. \end{array}$$

Note that Simari & Loui's default rules are not threeplace as Reiter's defaults, but twoplace. The set of *grounded instances* of Δ, i.e., of defeasible rules without variables, is denoted by $\Delta^{\downarrow}$. The notion of argument that Simari & Loui maintain is somewhat uncommon:

DEFINITION 63. (Arguments.) Given a context $\mathcal{K} = \mathcal{K}_N \cup \mathcal{K}_C$ and a set Δ of defeasible rules we say that a subset T of $\Delta^{\downarrow}$ is an *argument* for $h \in Sent_C(\mathcal{L})$ in the context $\mathcal{K}$, denoted by $\langle T, h\rangle_{\mathcal{K}}$ if and only if

1. $\mathcal{K} \cup T \mid\!\sim h$
2. $\mathcal{K} \cup T \not\mid\!\sim \bot$
3. $\nexists\, T' \subset T :\ \mathcal{K} \cup T' \mid\!\sim h$

An argument $\langle T, h_1\rangle_{\mathcal{K}}$ is a subargument of an argument $\langle S, h_2\rangle_{\mathcal{K}}$ iff $T \subseteq S$.

That $\mathcal{K} \cup T \mid\!\sim h$ means that h is derivable from $\mathcal{K} \cup T$ with first-order inferences applied to first-order formulas and modus ponens applied to defaults. Thus, an argument T is a set of grounded instances of defeasible rules containing sufficient rules to infer h (1), containing no rules irrelevant for inferring h (3), and not making it possible to infer $\bot$ (2). This notion of argument is somewhat uncommon because it does not refer to a tree or chain of inference rules. Instead, Definition 63 merely demands that an argument is a unordered collection of rules that together imply a certain conclusion.

Simari & Loui define conflict between arguments as follows. An argument $\langle T, h_1\rangle_{\mathcal{K}}$ *counterargues* an argument $\langle S, h_2\rangle_{\mathcal{K}}$ iff the latter has a subargument $\langle S', h\rangle_{\mathcal{K}}$ such that $\langle T, h_1\rangle_{\mathcal{K}}$ *disagrees* with $\langle S', h\rangle_{\mathcal{K}}$, i.e., $\mathcal{K} \cup \{h_1, h\} \vdash \bot$.

Arguments are compared with Poole's [1985] definition of specificity: an argument A *defeats* an argument B iff A disagrees with a subargument B^- of B and A is more specific than B^-. Note that this allows for subargument defeat:

this is necessary since Simari & Loui's definition of the status of arguments is not explicitly recursive. In fact, they use Pollock's theory of level-n arguments. Since they exclude self-defeating arguments by definition, they can use the version of Definition 11.

An important component of Simari & Loui's system is the Σ^k-operator. Of all the conclusions that can be argued, the Σ^k-operator returns the conclusions that are supported by level-k arguments. Simari & Loui prove that arguments for which $\Sigma^k = \Sigma^{k+1}$, are justified. The main theorem of the paper states that the set of justified conclusions is uniquely determined, and that a repeated application of the Σ-operator will bring us to that set.

A strong point of Simari & Loui's approach is that it combines the ideas of specificity (Poole) and level-n arguments (Pollock) into one system. Another strong point of the paper is that it presents a convenient calculus of arguments, that possesses elegant mathematical properties. Finally, Simari & Loui sketch an interesting architecture for implementation, which has a dialectical form (see below, Section 6 and, for a full description, [Simari *et al.*, 1994, Garcia *et al.*, 1998]). However, the system also has some limitations. Most of them are addressed by Prakken & Sartor [1996, 1997b], to be discussed next.

5.7 Prakken & Sartor

Inspired by legal reasoning, Prakken & Sartor [1996, 1997b] have developed an argumentation system that combines the language (but not the rest) of default logic with the grounded semantics of the BDKT approach.[14] Actually, Prakken & Sartor originally used the language of extended logic programming, but Prakken [1997] generalised the system to default logic's language. Below we present the latter version. The main contributions to defeasible argumentation are a study of the relation between rebutting and assumption attack, and a formalisation of argumentation about the criteria for defeat. The use of default logic's language and grounded semantics make Prakken & Sartor's system rather similar to Simari & Loui's. However, as just noted, they extend and revise it in a number of respects, to be indicated in more detail below.

As for the logical language, the premises are divided into factual knowledge $\mathcal{F}$, a set of first-order formulas subdivided into the necessary facts $\mathcal{F}_n$ and the contingent facts $\mathcal{F}_c$, and defeasible knowledge Δ, consisting of Reiter-defaults. The set $\mathcal{F}$ is assumed consistent. Prakken & Sartor write defaults as follows.

$$d\text{: } \varphi_1 \wedge \ldots \wedge \varphi_j \wedge \sim \varphi_k \wedge \ldots \wedge \sim \varphi_n \Rightarrow \psi$$

where d, a term, is the informal name of the default, and each φ_i and ψ is a first-order formula. The part $\sim \varphi_k \wedge \ldots \wedge \sim \varphi_n$ corresponds to the middle part of a Reiter-default. The symbol $\sim$ can be informally read as 'not provable that'. For each $\sim \varphi_i$ in a default, $\neg\varphi_i$ is called an *assumption* of the default. The language is defined such that defaults cannot be nested, nor combined with other formulas.

[14] A forerunner of this system was presented in [Prakken, 1993].

Arguments are, as in [Simari & Loui, 1992], chains of defaults 'glued' together by first-order reasoning. More precisely, consider the set $\mathcal{R}$ consisting of all valid first-order inference rules plus the following rule of *defeasible modus ponens (DMP)*:

$$\frac{d: \quad \varphi_0 \wedge \ldots \wedge \varphi_j \wedge \sim \varphi_k \wedge \ldots \wedge \sim \varphi_m \Rightarrow \varphi_n, \quad \varphi_0 \wedge \ldots \wedge \varphi_j}{\varphi_n}$$

where all φ_i are first-order formulas. Note that *DMP* ignores a default's assumptions; the idea is that such an assumption is untenable, this will be reflected by a successful attack on the argument using the default.

An argument is defined as follows.

DEFINITION 64. (Arguments.) Let Γ be any default theory $(\mathcal{F}_c \cup \mathcal{F}_n \cup \Delta)$. An *argument based on* Γ is a sequence of distinct first-order formulas and/or ground instances of defaults $[\varphi_1, \ldots, \varphi_n]$ such that for all φ_i:

- $\varphi_i \in \Gamma$; or
- There exists an inference rule $\psi_1, \ldots, \psi_m / \varphi_i$ in $\mathcal{R}$ such that $\psi_1, \ldots, \psi_m \in \{\varphi_1, \ldots, \varphi_{i-1}\}$

For any argument A

- $\varphi \in A$ is a *conclusion* of A iff φ is a first-order formula;
- $\varphi \in A$ is an *assumption* of A iff φ is an assumption of a default in A;
- A is *strict* iff A does not contain any default; A is *defeasible* otherwise.

The set of conclusions of an argument A is denoted by $CONC(A)$ and the set of its assumptions by $ASS(A)$.

Note that unlike in Simari & Loui, arguments are not assumed consistent. Here is an example of an argument:

$$[a,\ r_1\colon a \wedge \sim \neg b \Rightarrow c,\ c,\ a \wedge c,\ r_2\colon a \wedge c \Rightarrow d,\ d,\ d \vee e]$$

$CONC(A) = \{a, c, a \wedge c, d, d \vee e\}$ and $ASS(A) = \{b\}$.

The presence of assumptions in a rule gives rise to two kinds of conflicts between arguments, conclusion-to-conclusion attack and conclusion-to-assumption attack.

DEFINITION 65. (Attack.) Let A and B be two arguments. A *attacks* B iff

1. $CONC(A) \cup CONC(B) \cup \mathcal{F}_n \vdash \bot$; or
2. $CONC(A) \cup \mathcal{F}_n \vdash \neg\varphi$ for any $\varphi \in ASS(B)$.

Prakken & Sartor's notion of defeat among arguments is built up from two other notions, 'rebutting' and 'undercutting' an argument. An argument A *rebuts* an argument B iff A conclusion-to-conclusion attacks B and either A is strict and B is defeasible, or A's default rules involved in the conflict have no lower priority than B's defaults involved in the conflict. Identifying the involved defaults and

applying the priorities to them requires some subtleties for which the reader is referred to Prakken & Sartor [1996, 1997b] and Prakken [1997]. The source of the priorities will be discussed below.

An argument A *undercuts* an argument B precisely in case of the second kind of conflict (attack on an assumption). Note that it is not necessary that the default(s) responsible for the attack on the assumption has/have no lower priority than the default containing the assumption. Note also that Prakken & Sartor's undercutters capture a different situation than Pollock's: their undercutters attack an explicit non-provability assumption of another argument (in Section 3 called 'assumption attack'), while Pollock's undercutters deny the relation between premises and conclusion in a non-deductive argument.

Prakken & Sartor's notion of defeat also differs from that of Pollock [1995]. An inessential difference is that their notion allows for 'subargument defeat'; this is necessary since their definition of the status of arguments is not explicitly recursive (cf. Subsection 4.1). More importantly, Prakken & Sartor regard undercutting defeat as prior to rebutting defeat.

DEFINITION 66. (Defeat.) An argument A *defeats* an argument B iff $A = []$ and B attacks itself, or else if

- A undercuts B; or
- A rebuts B and B does not undercut A.

As mentioned above in Subsection 4.1, the empty argument serves to adequately deal with self-defeating arguments. By definition the empty argument is not defeated by any other argument.

The rationale for the precedence of undercutters over rebutters is explained by the following example.

EXAMPLE 67. Consider

r_1 : $\sim \neg$ *Brutus* `is innocent` $\Rightarrow$ *Brutus* `is innocent`
r_2 : $\varphi \Rightarrow \neg$ *Brutus* `is innocent`

Assume that for some reason r_2 has no priority over r_1 and consider the arguments $[r_1]$ and $[\ldots, r_2]$.[15] Then, although $[r_1]$ rebuts $[\ldots, r_2]$, $[r_1]$ does not defeat $[\ldots, r_2]$, since $[\ldots, r_2]$ undercuts $[r_1]$. So $[\ldots, r_2]$ strictly defeats $[r_1]$.

Why should this be so? According to Prakken & Sartor, the crux is to regard the assumption of a rule as one of its conditions (albeit of a special kind) for application. Then the only way to accept both rules is to believe that Brutus is not innocent: in that case the condition of r_1 is not satisfied. By contrast, if it is believed that Brutus is innocent, then r_2 has to be rejected, in the sense that its conditions are believed but its consequent is not ('believing an assumption' here means not believing its negation). Note that this line of reasoning does not naturally apply to

[15] We abbreviate arguments by omitting their conclusions and only giving the names of their defaults. Furthermore, we leave implicit that r_2's antecedent φ is derived by a subargument of possibly several steps.

undercutters Pollock-style, which might explain why in Pollock's [1995] rebutting and undercutting defeaters stand on equal footing.

Finally, we come to Prakken & Sartor's definition of the status of arguments. As remarked above, they use the grounded semantics of Definition 7. However, they change it in one important respect. This has to do with the origin of the default priorities with which conflicting arguments are compared.

In artificial intelligence research the question where these priorities can be found is usually not treated as a matter of common-sense reasoning. Either a fixed ordering is simply assumed, or use is made of a specificity ordering, read off from the syntax or semantics of an input theory. However, Prakken & Sartor want to capture that in many domains of common-sense reasoning, like the law or bureaucracies, priority issues are part of the domain theory. This even holds for specificity; although checking which argument is more specific may be a logical matter, deciding to prefer the most specific argument is an extra-logical decision. Besides varying from domain to domain, the priority sources can also be incomplete or inconsistent, in the same way as 'ordinary' domain information can be. In other words, reasoning about priorities is defeasible reasoning. (This is why our example of the introduction contains a priority argument, viz. A's use of (9) and (10).) For these reasons, Prakken & Sartor want that the status of arguments does not only *depend* on the priorities, but also *determines* the priorities. Accordingly, priority conclusions can be defeasibly derived within their system in the same way as conclusions like 'Tweety flies'.[16]

To formalise this, Prakken & Sartor need a few technicalities. First the first-order part of the language is extended with a special twoplace predicate $\prec$. That $x \prec y$ means that y has priority over x. The variables x and y can be instantiated with default names. This new predicate symbol should denote a strict partial order on the set of defaults that is assumed by the metatheory of the system. For this reason, the set $\mathcal{F}_n$ must contain the axioms of a strict partial order:

transitivity: $\forall x, y, z.\ x \prec y \wedge y \prec z \supset x \prec z$
asymmetry: $\forall x, y.\ x \prec y \supset \neg\, y \prec x$

For simplicity, some restrictions on the syntactic form of priority expressions are assumed. $\mathcal{F}_c$ may not contain any priority expressions, while in the defaults priority expressions may only occur in the consequent, and only in the form of conjunctions of literals (a literal is an atomic formula or a negated atomic formula). This excludes, for instance, disjunctive priority expressions.

Next, the rebut and defeat relations must be made relative to an ordering relation that might vary during the reasoning process.

DEFINITION 68. For any set S of arguments

- $<_S = \{r < r' \mid r \prec r'$ is a conclusion of some $A \in S\}$

[16] For some non-argument-based nonmonotonic logics that deal with this phenomenon, see Grosof [1993], Brewka [1994a, 1996], Prakken [1995] and Hage [1997]; see also Gordon's [1995] use of [Geffner & Pearl, 1992].

- A (strictly) S-defeats B iff, assuming the ordering $<_S$ on Δ, A (strictly) defeats B.

The idea is that when it must be determined whether an argument is acceptable with respect to a set S of arguments, the relevant defeat relations are verified relative to the priority conclusions drawn by the arguments in S.

DEFINITION 69. An argument A is *acceptable* with respect to a set S of arguments iff all arguments S-defeating A are strictly S-defeated by some argument in S.

Note that this definition also replaces the second occurrence of defeat in Definition 6 with strict defeat. This is because otherwise it cannot be proven that no two justified arguments are in conflict with each other.

Prakken & Sartor then apply the construction of Proposition 9 with Definition 69. They prove that the resulting set of justified arguments is unique and conflict-free and that, when S is this set, the ordering $<_S$ is a strict partial order. They also prove that if an argument is justified, all its subarguments are justified.

We illustrate the system with the following example.

EXAMPLE 70. Consider an input theory with empty $\mathcal{F}_c$, $\mathcal{F}_n$ containing the above axioms for $\prec$, and Δ containing the following defaults.

r_0: $\Rightarrow a$ $\quad$ r_4: $\Rightarrow r_0 \prec r_3$
r_1: $a \Rightarrow b$ $\quad$ r_5: $\Rightarrow r_3 \prec r_0$
r_2: $\sim b \Rightarrow c$ $\quad$ r_6: $\Rightarrow r_5 \prec r_4$
r_3: $\Rightarrow \neg a$

The set of justified arguments is constructed as follows (for simplicity we ignore combinations of the listed arguments).

$F^0 = \emptyset$ $\quad$ $<_0 = \emptyset$
$F^1 = \{[], [r_6]\}$ $\quad$ $<_1 = \{r_5 < r_4\}$
$F^2 = F^1 \cup \{[r_4]\}$ $\quad$ $<_2 = \{r_5 < r_4, r_0 < r_3\}$
$F^3 = F^2 \cup \{[r_3]\}$ $\quad$ $<_3 = <_2$
$F^4 = F^3 \cup \{[r_2]\}$ $\quad$ $<_4 = <_3$
$F^5 = F^4$ $\quad$ $<_5 = <_4$

Kowalski & Toni [1996] propose an alternative formalisation of reasoning about priorities, which does not require a change of the logic. They show how within the BDKT approach priority statements can be encoded with assumptions. This method requires that the notion of conflicting rules is expressed in the logical language of the system. Similar methods in non-argument-based approaches have been proposed by Gordon [1995] and Hage [1997].

Procedural form

Like several other systems, Prakken & Sartor define a procedural version of their system in dialectical form. Compared to the other systems, its main feature is

that it also covers debates about priorities. It will be discussed in some detail in Section 6.

Comparison with Simari & Loui [1992]

As remarked above, Prakken & Sartor's system is (in the version of [Prakken, 1997]) similar to Simari & Loui's. They both use the language of default logic, and their notions of an argument are quite similar: in particular, both systems use a modus ponens rule for defaults. Finally, both systems use grounded semantics and both have a procedural version in dialectical form. However, we have also seen that Prakken & Sartor extend Simari & Loui's system in a number of respects: their defaults are not twoplace but threeplace, which makes it possible to distinguish rebutting from assumption attack; they allow for comparing arguments on any ground, and they allow for debates on these grounds.

5.8 Nute's Defeasible Logic

A development closely related to defeasible argumentation is so-called 'defeasible logic', initiated by Donald Nute, e.g. [1994].[17] In both fields the notion of defeat is central. However, while in defeasible argumentation defeat is among arguments, in defeasible logic it happens between rules. Nevertheless, the approaches are sufficiently similar to warrant a discussion of defeasible logic in this chapter.

In several publications Nute has developed a family of such logics. For explanatory purposes we discuss the simplest version, described in [Nute & Erk, 1995]. In a way this is unfair, since this version has a problem that is absent in the other versions. However, it is instructive to see what the problem is, and we shall indicate how Nute deals with it in his other work.

Nute's systems are based on the idea that defaults are not propositions but inference licenses. Thus Nute's defeasible rules are, like Reiter's defaults, one-directional. However, unlike Reiter's defaults they are twoplace; assumption attacks are dealt with by an explicit category of defeater rules, which are comparable to Pollock's undercutting defeaters, although in Nute's case they are, like his defeasible rules, not intended to express general principles of inference but, as in default logic, domain specific generalisations.

As for the underlying logical language, since Nute's aim is to develop a logic that is efficiently implementable, he keeps the language as simple as possible. It has three categories of one-direction rules, viz. *strict* rules $A \rightarrow p$, *defeasible rules* $A \Rightarrow p$ and *defeaters* $A \rightsquigarrow p$. In all three cases p is a strong literal, i.e., an atomic proposition or a classically negated atomic proposition, and A is a finite set of strong literals. Defeaters must be read as 'if A then it might be that p'. Defeaters cannot be used to derive formulas; they can only be used to block an application of a rule $B \Rightarrow \neg p$. An example is 'Genetically altered penguins might fly', which

[17] In fact, Nute [1994] also counts systems for defeasible argumentation as defeasible logics.

undercuts 'Penguins don't fly'. Thus Nute has, like Pollock, both rebutting and undercutting conflicts between arguments.

Arguments can be formed by chaining rules into trees, and conflicting arguments are compared with the help of an ordering on the rules. Actually, Nute does not work with an explicit notion of argument; instead he incorporates it in two notions of derivability, strict ($\vdash$) and defeasible ($|\!\sim$) derivability, to be explained below. To capture non-derivability, Nute does not use the familiar notions $\nvdash$ (meaning 'not $\vdash$') and $|\!\not\sim$ (meaning 'not $|\!\sim$'). Instead, his aim of designing a tractable system leads him to define two notions of *demonstrable* non-derivability $\dashv$ and $\sim\!|$, which require that a proof of a formula fails after finitely many steps.

As just stated, Nute's assessment of arguments is implicit in his definitions of derivability. Nute has two core definitions, depending on when the last rule of the tree is strict or defeasible. (He has similar rules for $\dashv$ and $\sim\!|$.) The first definition detaches consequences of strict rules.

DEFINITION 71. (Strict derivability.) $T \vdash p$ if

1. $p \in T$, or
2. There is a $A \rightarrow p \in T$ such that for every $a \in A$, $T \vdash a$.

The second definition detaches consequences of defeasible rules, taking into account all nonmonotonic proofs that derive the contrary:

DEFINITION 72. (Defeasible derivability.) $T |\!\sim p$ if there is a rule $A \Rightarrow p \in T$ such that

1. $T \dashv \neg p$, and
2. for each $a \in A$, $T |\!\sim a$, and
3. for each $B \rightarrow \neg p \in T$ there is $b \in B$ such that $T \sim\!| b$, and
4. for each $C \Rightarrow \neg p \in T$ or $C \rightsquigarrow \neg p \in T$, either
   (a) there is a $c \in C$ such that $T \sim\!| c$ or
   (b) $A \Rightarrow p$ has higher priority than $C \rightarrow \neg p$ (or than $C \rightsquigarrow \neg p$).

Condition (1) says that the opposite of p must demonstrably be not strictly derivable. This gives strict arguments priority over defeasible arguments. For the rest, this definition has the recursive structure discussed above in Section 4.1. There must be a defeasible rule for p which, firstly, 'fires', i.e., of which all antecedents are themselves defeasibly derivable (condition 2) and which, secondly, is of higher priority than any conflicting rule which also fires: for any rule which is not lower, at least one antecedent must be demonstrably non-derivable (conditions 3–4). As a special case, condition (3) implicitly gives priority to strict rules over defeasible rules; for the rest these priorities must be defined by the user (condition 4),

although Nute pays much attention to the specificity criterion. Note that like Pollock [1995], Nute applies priorities to decide whether undercutting attack succeeds.

A literal can also be derived defeasibly from a strict rule, namely, when one of its antecedents is itself derived defeasibly. When there is a strict rule for p, the definition of defeasible derivability is simpler: since strict rules have priority over the other two categories, condition 4 can be dropped. In consequence, defeasible derivability from a strict rule can only be blocked by derivability from a conflicting strict rule.

Since Definitions 71 and 72 have the recursive structure of Definition 17, they share with this definition the problem that multiple assignments are not always avoided. Consider the following variant of Example 23.

EXAMPLE 73. Assume we have the following rules

1. $\Rightarrow p$
2. $p \Rightarrow q$
3. $\Rightarrow \neg q$
4. $\neg q \Rightarrow \neg p$

Three status assignments satisfy the above definitions.

Status assignment 1: $T \mid\!\sim p$, $T \mid\!\sim q$, $T \sim\!\!\mid \neg p$, $T \sim\!\!\mid \neg q$;
Status assignment 2: $T \mid\!\sim \neg p$, $T \mid\!\sim \neg q$, $T \sim\!\!\mid p$, $T \sim\!\!\mid q$
Status assignment 3: $T \sim\!\!\mid p$, $T \sim\!\!\mid q$, $T \sim\!\!\mid \neg p$, $T \sim\!\!\mid \neg q$

Only the third assignment is intended by Nute. In his other work, e.g. [Nute, 1994], he reformulates Definitions 71 and 72, and also the rules for $\sim\!\!\mid$, as conditions on finite proof trees for a formula. This solves the problem, since for the unintended status assignments no proof trees can be constructed. The crux is that $\sim\!\!\mid$ must also be established by constructing a finite proof tree (being a finite proof that a formula cannot be derived). And in the above example this is impossible.

Another problem inherited from Definition 17 is that Nute's system cannot capture floating conclusions (cf. Example 24). This is since an inference of p can only be blocked by a rule for $\neg p$ if all antecedents for that rule are derivable. Since Nute has no third category 'defensible' in between '(demonstrably) derivable' and '(demonstrably) not derivable', two rules that are in an irresolvable conflict do not give rise to conclusions and thus cannot block other inferences.

Finally, Nute's system behaves in a somewhat peculiar way when a conflict involves strict rules, as in the following example:

1. x `has children` $\Rightarrow x$ `is married`
2. x `lives alone` $\Rightarrow x$ `is a bachelor`
3. x `is married` $\rightarrow \neg\, x$ `is a bachelor`

In Nute's system only rules with directly contradicting heads are compared, and since strict rules prevail over defeasible rules, the outcome is that x is a bachelor,

even if the first defeasible rule has priority over the second. This seems counter-intuitive. In [Simari & Loui, 1992] and [Prakken & Sartor, 1997b] this problem does not occur, since there (3) is in the necessary facts $\mathcal{F}_n$, which count in testing whether conclusions contradict each other, for which reason the conflict is recognised as being between (1) and (2). It should be noted that in his most recent work Nute deals with this problem [Nute, 1997].

Evaluation

Evaluating Nute's defeasible logic, we see that it is an instance of the recursive-definition variant of the multiple-status-assignments approach, without an intermediate notion of defensible arguments. Consequently, his system has some problems with zombie arguments and floating conclusions. On the positive side, Nute's system gives intuitive results for a large class of benchmark examples and is, due to its simple language and its transparent definitions, very suitable for implementation.

As for the relation with defeasible argumentation, although Nute never introduced 'argument' as a concept in his defeasible logics, his theory can easily be recast in terms of arguments. One way to do this is to chain Nute's rules into trees (analogously to Lin & Shoham or Vreeswijk) and call them arguments (these trees must not be confused with the above-mentioned proof trees, which are proofs that a formula is defeasibly derivable). With this definition of arguments, Definition 72 can be stated alternatively in the way Vreeswijk defines defeat among arguments. A first conclusion that may be drawn from such a translation is that Nute's logic for defeasible reasoning is closely related to other approaches discussed here. This close relation justifies the discussion of defeasible logic in this chapter. Another conclusion is that arguments in Nute's logic defeat each other on the basis of information in top-rules only. This is due to the fact that a strong literal in Nute's system is defeasibly derivable only if the antecedent of the last rule applied is defeasibly derivable. This is in contrast with Vreeswijk's theory, in which arguments are compared and defeated in their entirety.

5.9 Defeasible argumentation in reasoning about events (Konolige, 1988)

Konolige's [1988] system ARGH (Argumentation with Hypotheses) was presented as a solution to the Yale Shooting Problem (YSP) [Hanks & McDermott, 1987]. Although the resulting formalism is still rather rudimentary, Konolige's discussion anticipates many issues and distinctions of later work, so that ARGH can be regarded as one of the forerunners of the field of defeasible argumentation.

The YSP concerns reasoning about events. The main problem to be dealt with is that sometimes the tendency of facts to 'persist' over time conflicts with the change of these facts by certain events. Konolige uses argumentation to allow various types of arguments based on considerations of persistence or change, and to adjudicate between conflicting arguments by means of principles of defeat. One

such principle says that arguments based on change caused by events defeat arguments based on persistence.

The logical language of ARGH resembles McCarthy's [1969] situation calculus, were properties are attached to situations and events bring us in new situations, with new properties. This language is used for giving *world descriptions*. An example of a world-description is

$W =$	$\{$	$p, q, \neg r, s \mid s_0$,	$s_0 \rightarrow_\alpha s_1$,	$p, \neg q \mid s_1$ $\}$
		The propositions p, q, $\neg r$ and s hold at situation s_0.	At situation s_0, action α brings us to situation s_1.	At situation s_1, the proposition p still holds, but q does not.

This scheme forms a single world description, consisting of three statements. The second statement is an *event description*, connecting the two *situation descriptions* that are stated on the first and the third line. Thus, typically, the letters $s_0, s_1, \ldots$ denote *situations*, the letters $p, q, r, \ldots$ denote propositions or *properties* that hold at situations, the letters $\alpha, \beta, \ldots$ denote actions or *events*. In ARGH, a world description can be partial: in the example above, neither $\neg r$, s, nor their negations are specified at s_1.

The purpose of argumentation in ARGH is to fill in partially described worlds as much as possible, by drawing conclusions regarding missing propositional values. Konolige considers three elementary types of inference rules for constructing arguments (which because of their generality are comparable to Pollock's notion of defeasible reasons).

	Notation	*Meaning*
Forward persistence:	$p \mid s_i \rightarrow_{\text{persist}} p \mid s_i + 1$	If p holds at s_i, then it is likely that p holds at the next situation s_{i+1}.
Backward persistence:	$p \mid s_i + 1 \rightarrow_{\text{persist}} p \mid s_i$	If p holds at s_{i+1}, then it is likely that p is inherited from the previous situation s_i.
A p-establishing action:	$\mid s_i \rightarrow_\alpha p \mid s_i + 1$	Doing α in s_i results in p at $s_i + 1$, defeasibly.

Labels such as 'persist', α and β, are not typed, that is, do not belong to a certain class of actions or propositions.

The above notation is used as a basis for constructing compound arguments and for performing defeasible reasoning. For example, the world

$W =$	$p \mid s_1$	Proposition p holds at s_1
	$s_1 \rightarrow_{\text{wait}} s_2 \rightarrow_\beta s_3$	At s_1, waiting brings us in s_2; then, performing β in s_2, brings us in s_3.

enables a number of arguments such as

	Argument	*For*
A	$p \mid s_1 \rightarrow_{\text{persist}} p \mid s_2$	$p \mid s_2$
B	$p \mid s_2 \rightarrow_{\text{persist}} p \mid s_3$	$p \mid s_3$
$A; B$	A followed by B	$p \mid s_3$
B'	$p \mid s_2 \rightarrow_{\beta} \neg p \mid s_3$	$\neg p \mid s_3$
$A; B'$	A followed by B'	$\neg p \mid s_3$
C	$\neg p \mid s_3 \rightarrow_{\text{persist}} \neg p \mid s_2$	$\neg p \mid s_2$
$A; B'; C$	A, B' followed by C	$\neg p \mid s_2$

$A; B$ and $A; B'$ are conflicting arguments. An argument for $\neg p \mid s_2$ is $A; B'$, followed by C (backward persistence). In this way, the arguments A and $A; B'; C$ compete for p.

To adjudicate among competing arguments, Konolige formulates a number of *rules of defeat*, such as the rule that event arguments have priority over persistence arguments. However, he also observes that this priority rule is defeasible, by giving an example in which backwards persistence is stronger than that change-by-event. In fact, one of Konolige's main observations is that any general, domain-independent priority principle will be very weak, and that information from the semantics of the domain will be the most important way of deciding among competing arguments. Such semantic information could, for instance, express the strength of the tendency of certain facts to persist over time. For example, the fact that a house will remain at its place is more likely to persist over time than the fact that a car will remain at its place. Thus Konolige anticipates later research on reasoning with and about domain specific priorities (see above, Section 5.7).

Evaluation

Evaluating Konolige's formalism, we can say that it is tailored to one particular problem, viz. reasoning about a changing world. However, for defeasible argumentation the main value of Konolige's system is not this application but the fact that it was one of the earliest argument-based accounts of defeasible reasoning, anticipating many of the issues arising in later work.

5.10 A brief overview of other work

We end this section with a brief overview of other work on logics for defeasible argumentation.

Loui [1987]

One of the initiators of the field of defeasible argumentation was Loui [1987]. On the basis of the same language as later used in [Simari & Loui, 1992], Loui defines arguments as graphs in which the links are formed by first-order inferences or default applications. Since defaults are twoplace, Loui only has rebutting attack.

In particular, an argument A is a counterargument of an argument B if the root of A is inconsistent with some node in B. Loui orders conflicting arguments in terms of four syntactic specificity criteria, and then defines an argument A to be justified iff it is undefeated (with respect to its top node) and all its counterarguments (i.e., all argument attacking another node of A) are defeated by a counterargument.

As this brief description shows, Loui's [1987] system already has all the elements of an argumentation system. The ideas of this paper have been very influential, but the formalism has some technical flaws, for which reason it has not survived. His paper with Simari was Loui's own attempt to overcome the flaws. Loui's most recent work (e.g. [Loui & Norman, 1995, Loui, 1998]) addresses the procedural aspects of argumentation.

Connection with truth-maintenance systems

Systems for defeasible argumentation are related to so-called truth-maintenance systems (TMSs). A TMS is a bookkeeping system for a reasoning system, in which logical dependencies among propositional beliefs, or assertions, are represented and maintained to preserve consistency of the reasoning system. There exists several TMSs, such as Justification-Based [Doyle, 1979], Assumption-Based, [De Kleer, 1986] and Logic-Based TMSs. Basically, in all TMSs all assertions are connected via a network of dependencies and all TMSs do some form of dependency-directed backtracking. In Justification-Based TMSs, for example,

- The structure of the assertions themselves is left unspecified. Each supported belief (assertion) has a so-called *justification*.
- Each justification has two parts:
  1. An IN-List which supports beliefs held.
  2. An OUT-List which supports beliefs not held.
- An assertion is connected to its justification by an arrow; one assertion can feed another justification thus creating the network.
- Assertions may be labelled with a belief status.
- An assertion is valid if every assertion in its IN-List is believed and none in its OUT-List are believed.
- An assertion is *non-monotonic* if the OUT-List is not empty or if any assertion in the IN-List is *non-monotonic*.

Thus, the concepts and ideas are similar in spirit to those underlying argumentation systems. For instance, the issues of multiple and nonexisting status assignments have been studied in the literature on Justification-Based TMSs as the issues of multiple and nonexisting labellings of a dependency network. Since [Doyle, 1979], a variety of TMSs have been developed as a means of implementing nonmonotonic

reasoning. The relation between TMSs and nonmonotonic reasoning is further discussed in [Martins & Reinfrank, 1991]. Baker & Ginsberg [1989] establish a connection with argument and debate.

Krause et al. [1995]

Recently, the system of Krause *et al.* [1995], further explored by Elvang-Gøransson & Hunter [1995], has attracted some attention in the multi-agent community, as a component of models of negotiation; cf. [Parsons *et al.*, 1998]. In this system, arguments are essentially a (Premises, Conclusion) pair, where the conclusion follows from the set Premises according to a system of intuitionistic logic. The conclusion of an argument can, as in Pollock's system, have a degree of belief, which allows arguments to be ordered using numerical (e.g. probabilistic) information. The only type of conflict is conclusion-to-conclusion attack. However, Krause *et al.* distinguish two subtypes, "rebutting" and "undercutting" conflict, with a deviating use of the term 'undercutter': in their terms, A undercuts B iff A rebuts (i.e., conclusion-to-conclusion-attacks) a subargument of B. (An argument (S, φ) is a subargument of an argument (T, ψ) iff $S \subseteq T$.)

The main feature that sets this system apart from other systems, is the definition of the status of arguments. Given rebutting and undercutting relations between arguments, arguments are divided into the following categories (relative to a certain input theory Δ).

DEFINITION 74. (Argument classes.)

- A1 is the class of all arguments that can be made from Δ.
- A2 is the class of all consistent arguments that can be made from Δ.
- A3 is the class of all consistent arguments from Δ without rebutting arguments.
- A4 is the class of all consistent arguments from Δ without undercutting arguments.
- A5 is the class of all arguments with empty set of Premises.

Observe that A5 $\subseteq$ A4 $\subseteq$ A3 $\subseteq$ A2 $\subseteq$ A1. (Note that rebutting an argument implies undercutting it.) Accordingly, arguments in smaller classes are regarded as better than arguments in larger classes. Krause *et al.* also consider a refinement of this ordering in terms of the degrees of belief of arguments.

In our opinion, a drawback of this definition is that it does not capture reinstatement.

Argument-based proof theories for preferential entailment

Two argumentation-theoretic proof theories have been proposed for a preferred-model semantics. As explained in Section 2, in preferential entailment defaults are represented as first-order material implications with special 'normality conditions', as in

(1) $\forall x.\mathtt{Bird}(x) \wedge \neg \mathtt{ab}_1(x) \supset \mathtt{Canfly}(x)$
(2) $\forall x.\mathtt{Penguin}(x) \wedge \neg \mathtt{ab}_2(x) \supset \neg \mathtt{Canfly}(x)$

First-order theories containing such defaults are then semantically interpreted by only looking at those models where the extension of the $\mathtt{ab}_i$ predicates are minimal (with respect to set inclusion), which captures the assumption that the world is as normal as possible.

The proof-theoretic idea is that arguments are (in their simplest form) a set of normality statements that can be added to a certain theory to derive certain conclusions. (This is essentially a special case of Bondarenko *et al.*'s [1997] assumption-based definition of an argument.) For instance, suppose that the defaults (1) and (2) are part of a first-order theory

$$T = \{1, 2\} \cup \{\mathtt{Penguin}(\textit{Tweety}), \forall x.\mathtt{Penguin}(x) \supset \mathtt{Bird}(x)\}$$

Then $A = \{\neg \mathtt{ab}_1(\textit{Tweety})\}$ is an argument for the conclusion $\mathtt{Canfly}(\textit{Tweety})$, since $T \cup A \vdash \mathtt{Canfly}(\textit{Tweety})$, and $B = \{\neg \mathtt{ab}_2(\textit{Tweety})\}$ is an argument for $\neg\mathtt{Canfly}(\textit{Tweety})$, since $T \cup B \vdash \neg\ \mathtt{Canfly}(\textit{Tweety})$. In order to capture floating conclusions (cf. Example 25), the general form of an argument is not that of a set but of a collection of sets of normality assumptions (an alternative form is that of a disjunction of conjunctions of such assumptions). Conflicting arguments can be compared in terms of an ordering of the normality assumptions.

Baker & Ginsberg [1989]

Baker & Ginsberg [1989] have applied this idea to the semantics of so-called prioritised circumscription. In their proof theory, an argument A *rebuts* another argument B if A and B have contradictory conclusions, and if A's least default is not inferior to B's least default, while A *refutes* B if in addition its least default has priority over the least default of B. A defeasible proof then has a dialectical form, which form will be discussed in detail in Section 6. Baker & Ginsberg prove that this proof theory is sound and complete with respect to the model theory of prioritised circumscription.

Geffner & Pearl [1992]

Geffner & Pearl [1992] have proposed similar ideas, in a proof theory for their "conditional entailment" (see also [Geffner, 1991], for an application to logic programming's negation as failure). When representing default rules, a minor difference with Baker & Ginsberg is that they use positive 'applicability' atoms δ_i instead of negated abnormality atoms. In their preferred model semantics they then prefer those models which make as few applicability atoms false as possible. In

ordering applicability atoms, Geffner and Pearl define a class of "admissible orderings" which, if respected by the preference relation on models, reflects the notion of specificity. Although this notion is the only source of priorities that Geffner & Pearl consider, their formalism seems not to exclude orderings on the δ_i's based on other standards.

Geffner & Pearl's proof theory is sound and complete with respect to conditional entailment. They also define an architecture for (incompletely) implementing the proof theory as a computer program, which has the dialectical flavour that will be the topic of Section 6. Bondarenko *et al.* [1997] conjecture that it computes the grounded semantics of Definition 7.

Evaluation

The idea of providing a model-theoretic foundation for defeasible argumentation is interesting, but as we remarked at the end of Section 3, a critical test for such approaches is whether the resulting criteria for model preference are sufficiently natural. For certain restricted applications this test might succeed, but it remains to be seen to what extent this approach can be generalised; for instance, to argumentation systems that allow for inductive, analogical or abductive arguments.

Verheij [1996]

Verheij combines ideas of Lin & Shoham and Vreeswijk on the structure of arguments with Pollock's partial status assignments into a formalism called CumulA. This system has three distinctive features. The first is a new type of argument called 'coordinated argument', which combines two arguments for the same conclusion. For instance, from the arguments *'The sun is shining. So, it is a beautiful day'* and *'The sky is blue. So, it is a beautiful day'* it is possible to construct a new argument *'The sun is shining; the sky is blue. So, it is a beautiful day'*. Verheij stresses that this is not the same as an ordinary argument with two premises: the semicolon expresses that each premise on its own also supports the conclusion. With coordinated arguments Verheij wants to capture the 'accrual of arguments', i.e., the phenomenon that a combination of arguments that are individually defeated by another argument, possibly defeats that argument.

EXAMPLE 75. (Accrual of arguments.) Consider the arguments

- A: Peter robbed a person, therefore Peter should be punished.
- B: Peter injured a person, therefore Peter should be punished.
- C: Peter is a minor offender and should therefore not be punished.
- D: Peter robbed a person. He injured that person too. Therefore, Peter should be punished.

According to Verheij, it is conceivable that the coordination D of A and B prevails over C, even if C would prevail over A and B when these are considered

individually. Accordingly, Verheij allows that a status assignment makes a coordinated argument 'in' even when any of its components would be 'out' when present without the others. On the other hand, any status assignment should make a coordinated argument 'in' if already one of its components is 'in'.

A second feature of CumulA is that it generalises other argumentation systems by making defeat a relation between *sets* of arguments. According to Verheij this enables a more natural formalisation of certain types of defeat. Verheij also argues that several types of defeat, such as Pollock's undercutters, cannot be defined in terms of inconsistency between conclusions of arguments. For this reason, in CumulA the relation of 'defeat' is, as in [Dung, 1995], a primitive notion and can be further defined in various ways, which may but need not be triggered by inconsistency of conclusions. Verheij claims that his treatment of defeaters is able to capture a wide range of types of defeat proposed in the literature.

A final feature of CumulA is its further development of Lin & Shoham's and Vreeswijk's notions of argument structures and sequences. In particular, Cumula models the replacement of a premise with an argument that has this premise as conclusion. Such a move is very common in actual debates but has not yet received much attention in the field of defeasible argumentation (but see Loui, 1998). Verheij also develops an elegant notation that shows how the status of arguments can change when more arguments are taken into account (Figure 9).

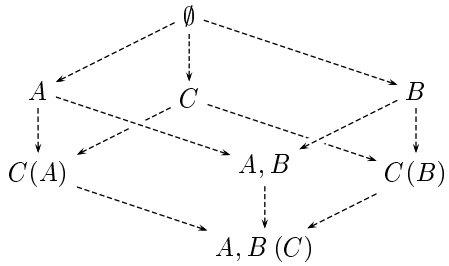


Figure 9. Stages of argumentation when C defeats A, C defeats B, but $\{A, B\}$ defeats C. Each node represents a partial defeat status assignment (cf. Definition 45), and reflects a 'stage' in the argumentation process. Arguments between parentheses have the status 'defeated', the other arguments have the status 'undefeated'.

Other work

Finally, we mention other relevant work on logics for defeasible argumentation.

Marek *et al.* [1990, 1992] aim to capture the main existing nonmonotonic logics in a general framework of so-called 'nonmonotonic rule systems'. The basic notion is not that of an argument but that of a (one-direction) rule. They define

a notion of extensions of a given rule system as a set of formulas that has certain closure and completeness properties with regard to rule application. Bondarenko *et al.* [1997] prove that these extensions correspond to stable semantics. Marek *et al.*'s ideas bear some resemblance to Lin & Shoham's system. Both systems aim to be a general framework for capturing nonmonotonic logics, both work with one-direction rules, and Marek *et al.*'s notion of extensions is related to Lin & Shoham's notion of a complete argument structure. Finally, neither have a mechanism of defeat among arguments (or proofs).

Benferhat *et al.* [1993] study argumentative reasoning with inconsistent databases. An argument for a formula is a consistent subset of a database (which is a set of logical formulas) that classically entails the formula. Conflicts between arguments are resolved with an ordering on the elements of the database. The approach and its relation with inconsistency handling approaches (cf. Section 2.1) and other argumentation systems is further investigated by Benferhat *et al.* [1995], Cayrol [1995] and Amgoud & Cayrol [1997].

The BDKT framework has triggered further work in the area of logic programming. For instance, Dung [1993] has applied his own framework to the semantics of extended logic programming. Thielscher [1996] has defined a semantics and proof theory for drawing sceptical conclusions from multiple status assignments, based on an adapted version of Dung's [1995] framework. And Jakobovits & Vermeir [1999] have generalised Dung's version of the BDKT framework by defining several weak notions of argument extensions, and examining the relation with the various BDKT semantics.

Finally, Starmans [1996] carries the ideas of defeasible argumentation to a multi-agent environment, where more than two parties participate in a dispute. Part of this endeavour is to show that n-party disputes, where $n \geq 3$, involve a richer arsenal of speech acts (question, demand for clarification, refusal of adduced evidence) and other types of attack than just rebutting or undercutting counterarguments (such as such as just refusing to accept a certain claim). As debate proceeds, on the basis of the individual theories a so-called *aggregated* theory is formed, which contains the claims that are supported collectively by the group of disputants. this process can be constrained by so-called *principles of preservation*. Starmans discusses a number of such principles analogous to choice principles in the theory of social choice.

6 DIALECTICAL FORMS OF ARGUMENTATION SYSTEMS

So far mainly semantical aspects have been discussed, where the main focus was on properties of *sets* of arguments. In this section we shall go deeper into proof-theoretical, or procedural aspects of argumentation, where the chief concern is to establish the status of *individual* arguments. Several argumentation systems have been formulated in dialectical style [Baker & Ginsberg, 1989, Simari & Loui, 1992, Vreeswijk, 1993b, Simari *et al.*, 1994, Dung, 1994, Brewka, 1994b,

Prakken & Sartor, 1996, Loui, 1998, Garcia *et al.*, 1998, Prakken, 1999, Kakas & Toni, 1999]. (It should be noted that Loui [1998] does not regard the dialectical style merely as a reformulation of declarative nonmonotonic logics, but as a formalism in its own right, capturing the "essentially constructive" nature of defeasible reasoning, which, Loui argues, cannot be captured by declarative formalisms.)

The common idea can be explained in terms of a dialogue game between two players, a proponent and an opponent of an argument. A dialogue 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 attacks the last move of the other party with a certain minimum force. The initial argument provably has a certain status if the proponent has a winning strategy, i.e., if he can make the opponent run out of moves whatever moves the opponent makes. The exact rules of the game depend on the semantics it is meant to capture. A natural idea here is that of dialectical asymmetry. For instance, if the game reflects sceptical reasoning, i.e., if it is meant to test whether an argument is justified, the proponent's arguments have to be strictly defeating, while the opponent's moves may be just defeating. If, on the other hand, the game reflects credulous reasoning, these rules must be reversed (as suggested by Prakken [1999]).

Let us introduce the concept of dispute more formally by making use of an adapted version of what is called a 'dialogue' in [Prakken & Sartor, 1996] and 'argument game' in [Loui, 1998]. It is meant to capture sceptical reasoning.

DEFINITION 76. (Disputes.) A *dispute* on an argument A is a non-empty sequence of arguments $move_i = (Player_i, A_i)$ $(i > 0)$ with $A_1 = A$, in which $Player_1$, denoted by **PRO**, uses odd-numbered moves to try to establish A and $Player_2$, denoted by **CON**, uses even-numbered moves to try to prevent $Player_1$'s success.

1. $Player_i =$ **PRO** iff i is odd; and $Player_i =$ **CON** iff i is even;
2. If $Player_i = Player_j =$ **PRO** and $i \neq j$, then $A_i \neq A_j$;
3. If $Player_i =$ **PRO** $(i > 1)$, then A_i strictly defeats A_{i-1};
4. If $Player_i =$ **CON**, then A_i defeats A_{i-1}.

The first condition stipulates that **PRO** begins and then the players take turns, while the second condition prevents the proponent from repeating its attacks. The remaining two conditions form the heart of the definition: they state the burdens of proof for **PRO** and **CON**. Thus, **PRO** is required to establish A while **CON** need only provide nuisance defeaters.

The various authors format their disputes in different ways. Vreeswijk [1993b, 1995] uses a format that displays the depth of the proof tree and is able to represent exhaustive disputes. (See below.) Here we have instead used a simplified version of the format used by [Dung, 1994, Prakken & Sartor, 1997b, Loui, 1998]. This format is simple and compact, but does not represent the depth of the proof tree.

EXAMPLE 77. Let A, B, C and D be arguments such that B and D defeat A, and C defeats B. Then a dispute on A may run as follows:

PRO: A, **CON**: B, **PRO**: C

In this dispute **PRO** advances A as an argument supporting the main thesis. (Arguments are conceived as primitive concepts here, so that the main thesis is left unspecified.) Both B and D defeat A, which means that **CON** has two choices in response to A. **CON** chooses to respond with B in the second move. Then C is the only argument defeating B, so that **PRO** has no choice than to respond with C in the third move. There are no arguments against C, so that **CON** cannot move and loses the dispute. As a result, A and C are established, and B is overruled by C. A dispute in which **CON** follows an optimal strategy is

PRO: A, **CON**: D

So in this game, under these rules, there is no winning strategy for player 1, **PRO**. The only reason why **PRO** wins the first dispute is that **CON** chooses the wrong argument, viz. B, in response to A. In fact, **CON** is in the position to win every game, provided it chooses the right moves. In other words, **CON** possesses a winning strategy.

The concept of dispute presently discussed can be characterised as a so-called *argument game*. An argument game is a 'one-dimensional' dispute in which each player may respond only once to each argument advanced by the opponent, and if that argument turns out to be ineffective, that player may not try a second reply to the same argument. Thus, no backtracking is allowed. This fact makes argument games into what is officially known as *two-player zero-sum games*, including the concepts that come with it, the most important of which is strategy.

Exhaustive dispute

The opposite of an argument game is a so-called *exhaustive dispute*. An exhaustive dispute is a dialogue in which each player is allowed to try out every possible rebuttal in reply to the arguments of its opponent. If a player discovers that it has put forward the wrong argument, it can recover from its mistake by trying another argument, provided there are such alternatives.

In displaying exhaustive disputes, we follow the format of Vreeswijk [1993b, 1995], in which the depth of the proof tree is represented by vertical bars in the left column:

1.	| **PRO** : argument 1	[justification]
2.	|| **CON** : reply	[justification for reply]
3.	||| **PRO** : reply to reply	···
4.	|| **CON** : 2nd reply to argument 1	···
5.	||| **PRO** : reply to 2nd reply	···
6.	|||| **CON** : reply to reply to 2nd reply	···
⋮	⋮	⋮

With the arguments presented in Example 77, **CON** has two strategies: one employing B and one employing D; let us refer to these strategies as *strategy* B and *strategy* D, respectively. As remarked above, when the players are engaged in an argument game, **CON** must choose between strategy B and strategy D. What **CON** cannot do is deploying B and D one after the other. In an exhaustive dispute, on the other hand, **CON** has the opportunity to try both strategies in succession:

1. | **PRO** : A [A]
2. || **CON** : B [B defeats A]
3. ||| **PRO** : C [C defeats B]
4. || **CON** : D [D defeats A]

At line 1, **PRO** advances A as an argument supporting the main thesis. (The main thesis is left unspecified here.) Both B and D defeat A, so that **CON** has two choices in response to A. **CON** chooses to respond with B at line 2. C is the only argument defeating B, so that **PRO** responds with C at line 3. There are no counterarguments to C, so that **CON** backtracks and searches new counterarguments to A. **CON** finds D as a new counterargument to A. At line 4, **CON** advances D in reply to A. There are no arguments against D, so that **PRO** cannot move and loses the dispute. As a result, we know that A cannot be established as justified.

Had **CON** responded with D instead of B at line 2, then the dispute would be settled within 2 moves:

1. | **PRO** : A [A]
2. || **CON** : D [D defeats C]

The choice and order of moves is determined by the players.

In the above definition as well as in most approaches a move consists of a complete argument. This means that the search for an individual argument is conducted in a 'monological' fashion, determined by the nature of the underlying logic; only the process of considering counterarguments is modelled dialectically. A notable exception is [Loui, 1998], in which arguments are constructed piecewise (beginning with the top-rule) and dialogue moves consist of

- attacking the conclusion of an unfinished argument,
- challenging an unfinished argument, or
- extending an unfinished argument in a top-down fashion on request of the opponent.

Another feature of Loui's protocol is that, to reflect the idea of resource-bounded reasoning, every move consumes resources except requests to the opponent to extend unfinished arguments.

Completeness results

An important objective in the dialectic approach is a correspondence between the various argument-based semantics and the different forms of dispute.

Dung [1994] establishes a correspondence between the semantics defined in Definition 7 (grounded semantics) and his notion of argument game. Dung's game is similar to the one of Definition 76, but it is different in two respects: it does not have the nonrepetition rule (2), and it allows that **PRO**'s moves are, like **CON**'s moves, just defeating. On the other hand, Prakken & Sartor[1997b] show that Dung's result also holds for Definition 76. Thus they give a justification to the nonrepetition rule and the dialectical asymmetry, in the sense that these features make debate more efficient while preserving semantical soundness of the game. Intuitively, this is since the only effect of these features is the termination of dialogues that could otherwise go on forever: thus they do not deny **PRO** any chance of winning the debate.

As for some details, Dung's idea is to establish a mapping for which

- arguments in the set F^{2i} map to arguments for which **PRO** has a winning strategy that results in an argument game of at most $2i$ moves
- arguments *not* in F^{2i+1} map to arguments for which **CON** has a winning strategy that results in an argument game of at most $2i + 1$ moves

Another completeness result is established by Vreeswijk [1995], between a particular form of exhaustive dispute and a variant of his argumentation system with grounded instead of stable semantics (in the 'levelled' form of Definition 11). And in a recent paper, Kakas & Toni [1999] define dialectical versions of most of the assumption-based semantics proposed by Bondarenko *et al.* [1997].

As remarked above, a dialectical version of preferred or stable semantics could be developed by reversing the dialectical asymmetry, i.e., by defining that **PRO**'s moves may be simply defeating, while **CON**'s moves must be strictly defeating. This can be illustrated with the Nixon diamond.

EXAMPLE 78. The Nixon diamond of Example 3 consists of two arguments, A and B, which together produce two stable extensions, viz. $S_1 = \{A\}$ and $S_2 = \{B\}$. Now, if **PRO** advances A as a main thesis, **CON** loses since there are no arguments that strictly defeat A:

1. | **PRO** : A

The outcome of this dispute corresponds with S_1. Likewise, if **PRO** advances B as a main thesis, **CON**cannot move either, and **PRO** also wins this dispute. This outcome corresponds with S_2.

Moreover, in case of an odd defeat loop with three arguments (Example 30) we also obtain the desired result, since it is easy to see that the non-repetition rule prevents **PRO** from establishing any of the three arguments (note that the only preferred extension in this example is the empty set).

However, just reversing the dialectical asymmetry is not sufficient: in Example 12 all arguments are in some preferred extension, yet for all those arguments a dispute with the above rules could be infinite. So further research is needed. In

fact, much of the remaining work on defeasible dialogue logics lies on the terrain of proving soundness and completeness results for the various types of semantics proposed so far.

Disputes with defeasible priorities

Prakken & Sartor [1997b] extend their dialectical proof theory (see Definition 76) to the case with defeasible priorities. The main problem is on the basis of which priorities the defeating force of the moves should be determined. In fact, a few very simple conditions suffice. **CON** may completely ignore priorities: it suffices that its moves $\emptyset$-defeat **PRO**'s previous move. And for **PRO** only those priorities count that are stated by **PRO**'s move itself, i.e., moving with an argument A is allowed for **PRO** if A strictly A-defeats **CON**'s previous move; in addition, **PRO** has a new move available, viz. moving a priority argument A such that **CON**'s last move does not A-defeat **PRO**'s previous move.

This results in the following change of conditions (3) and (4) of Definition 76.

(3) If $Player_i =$ ***PRO*** *(*$i > 1$*), then*

- Arg_i *strictly* Arg_i*-defeats* Arg_{i-1}*; or*
- Arg_{i-1} *does not* Arg_i*-defeat* A_{i-2}.

(4) If $Player_i =$ ***CON*** *then* Arg_i $\emptyset$*-defeats* Arg_{i-1}.

Prakken & Sartor [1997b] show that their correctness and completeness results also hold for this definition (although in this case dialectical asymmetry is necessary). The main feature of their system that ensures this is the following property of the defeat relation: if A S-defeats B and $S' \subseteq S$, then A S'-defeats B. Consider by way of illustration the dialectical version of Example 70.

PRO$_1$: $[r_2\colon \sim b \Rightarrow c]$	**CON**$_1$: $[r_0\colon \Rightarrow a,\ r_1\colon a \Rightarrow b]$
PRO$_2$: $[r_3\colon \Rightarrow \neg a,\ r_4\colon \Rightarrow r_0 \prec r_3]$	**CON**$_2$: $[r_5\colon \Rightarrow r_3 \prec r_0]$
PRO$_3$: $[r_6\colon \Rightarrow r_5 \prec r_4]$	

Here, **PRO**$_2$ uses the first available type of move, while **PRO**$_3$ uses the second type.

7 FINAL REMARKS

As we remarked in the introduction, the field of defeasible argumentation is still young, with a proliferation of systems and disagreement on many issues. Nevertheless, we have also observed many similarities and connections between the various systems, and we have seen that a formal meta-theory is emerging. In particular the BDKT approach has shown that a unifying account is possible; not only has it shown that many differences between argument-based systems are variations on just a few basic themes, but also has it shown how many nonmonotonic logics can

be reformulated in argument-based terms. And Pollock's work on partial computation and adequacy criteria for defeasible reasoners paves the way for more meta-theoretical research. This also holds for the work of Lin & Shoham, Vreeswijk and Verheij on argumentation sequences, and for the just-discussed work on argument games and disputes.

In addition, several differences between the various systems appear to be mainly a matter of design, i.e., the systems are, to a large extent, translatable into each other. This holds, for instance, for the conceptions of arguments as sets (Simari & Loui), sequences (Prakken & Sartor) or trees (Lin & Shoham, Nute, Vreeswijk), and for the implicit (BDKT, Simari & Loui, Prakken & Sartor), or explicit (Pollock, Nute, Vreeswijk) stepwise assessment of arguments. Moreover, other differences result from different levels of abstraction, notably with respect to the underlying logical language, the structure of arguments and the grounds for defeat. And some systems extend other systems: for example, Vreeswijk extends Lin & Shoham by adding the possibility to compare conflicting arguments, and Prakken & Sartor extend Simari & Loui with priorities from any source and with assumption attack, and they extend both Simari & Loui and Dung [1995] with reasoning about priorities. Finally, the declarative form of some systems and the procedural form of other systems are two sides of the same coin, as are the semantics and proof theory of standard logic.

The main substantial differences between the systems are probably the various notions of defeasible consequence described in Section 4, often reflecting a clash of intuitions in particular examples. Although the debate on the best definitions will probably continue for some time, in our opinion the BDKT approach has nevertheless shown that to a certain degree a unifying account is possible here also. Moreover, as already explained at the end of Section 4, some of the different consequence notions are not mutually exclusive but can be used in parallel, as capturing different senses in which belief in a proposition can be supported by a body of information. And each of these notions may be useful in a different context or for different purposes. Of course, in some cases this is otherwise. For instance, we would regard a definition as flawed if it does not capture indirect reinstatement (cf. p. 25). However, in general the existence of different definitions is not a problem for, but a feature of the field of defeasible argumentation. An important consequence of this is that the choice between the notions might depend on pragmatic considerations, as is, for instance, the case in legal procedure for the standards of proof. For example, the distinction in Anglo-Saxon jurisdictions between 'beyond reasonable doubt' in criminal cases and 'on the balance of probabilities' in civil cases is of a pragmatic nature; there are no intrinsic reasons to prefer one standard over the other as being 'the' standard of rational belief.

Another important difference is that while some systems formalise 'logically ideal' reasoners, other systems embody the idea of partial computation, i.e., of evaluating arguments not with respect to all possible arguments but only with respect to the arguments that have actually been constructed by the reasoner (Pollock, Loui, Vreeswijk, Verheij). However, here, too, we can say that these notions

are not rivals, but capture different senses of support for beliefs, perhaps useful in different contexts.

We end with listing some of the main open problems in defeasible argumentation.

- Some examples do not receive an adequate treatment in any of the semantics that we have discussed. This holds, for instance, for the 'seemingly defeated' arguments discussed in Section 5.2, and for Horty's example discussed in Section 5.3. And perhaps other 'critical' examples can be discovered.

- Verheij's work raises the question whether the conflict types that have been discussed in this chapter are all types of conflict that can exist between arguments.

- Another question raised by Verheij is what the best treatment is of accrual of arguments.

- Our informal remarks on the relation between the various systems should, where possible, be turned into a formal meta-theory of defeasible argumentation, making use of the work that has already been done.

- The procedural form of defeasible argumentation must be further developed; most current systems only have a semantic form.

- The notion of partial computation should be further studied. This notion is not only relevant for artificial intelligence but also for philosophy. The essence of defeasible reasoning is that it is reasoning under less than perfect conditions, where it is difficult or even impossible to obtain complete and reliable information. Since these conditions are very common in daily life, the correctness conditions for reasoning in such circumstances should be of interest to any logician who wants to study the formal structure of ordinary reasoning.

- Finally, it would be interesting to connect argumentation systems with research in so-called 'formal dialectics', which studies formal systems of procedural rules for dialogues; see e.g. Hamblin [1971], Mackenzie [1979] and Walton & Krabbe [1995]. Both fields would be enriched by such a connection. The argument games discussed in Section 6 are, unlike those of formal dialectics, not rules for real discussions between persons, but just serve as a proof theory for a (nonmonotonic) logic, i.e. they determine the (defeasible) consequences of a given set of premises. The 'players' of these argument games are not real actors but stand for the alternate search for arguments and counterarguments that is required by the proof theory. An embedding of argumentation systems in formal dialectics would yield an account of how their input theories are constructed dynamically during disputes between real discussants, instead of given in advance and fixed. On

the other hand, argumentation systems could also enrich formal dialectics, which lacks notions of counterargument and defeat; its underlying logic is still deductive and its main dialectical speech act is asking for premises that support a certain claim; 'real' counterarguments are impossible. Defeasible argumentation can provide formal dialectics with stronger dialectical features.

Some work of this nature has already been done, mainly in the area of artificial intelligence and law, e.g. [Hage *et al.*, 1994, Gordon, 1995, Loui & Norman, 1995, Prakken & Sartor, 1998, Lodder, 1999, Vreeswijk, 1999]; see also [Starmans, 1996] and especially [Loui, 1998]. Such work could provide a key in meeting Toulmin's [1958] challenge to logicians to study how the properties of disputational procedures influence the validity of arguments. Perhaps in 1958 Toulmin's challenge seemed odd, but 40 years of work in logic, philosophy, artificial intelligence and argumentation theory have brought an answer within reach.

ACKNOWLEDGEMENTS

We thank all those with whom over the years we have had fruitful discussions on the topic of defeasible argumentation. Useful comments on an earlier version of this chapter were given by Jaap Hage, Simon Parsons, John Pollock and Bart Verheij.

Affiliation
Henry Prakken, Department of Computer Science, Utrecht University, The Netherlands
Gerard Vreeswijk, Department of Cognitive Sciences, University of Groningen, The Netherlands

REFERENCES

[Amgoud & Cayrol, 1997] L. Amgoud & C. Cayrol, Integrating preference orderings into argument-based reasoning. *Proceedings of the International Conference on Qualitative and Quantitative Practical Reasoning (ECSQARU-FAPR'97)*. Lecture Notes in Artificial Intelligence 1244, 159–170. Berlin: Springer Verlag, 1997.

[Asher & Morreau, 1990] N. Asher & M. Morreau, Commonsense entailment: a modal theory of nonmonotonic reasoning. *Proceedings of the Second European Workshop on Logics in Artificial Intelligence (JELIA'90)*. Lecture notes in Artificial Intelligence 478, 1–30. Berlin: Springer Verlag, 1990.

[Baker & Ginsberg, 1989] A.B. Baker & M.L. Ginsberg, A theorem prover for prioritized circumscription. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 463–467, 1989.

[Benferhat *et al.*, 1993] S. Benferhat, D. Dubois & H. Prade, Argumentative inference in uncertain and inconsistent knowledge bases. *Proceedings of the 9th International Conference on Uncertainty in Artificial Intelligence*, 411–419. San Mateo, CA: Morgan Kaufman Publishers Inc, 1993.

[Benferhat *et al.*, 1995] S. Benferhat, D. Dubois & H. Prade, How to infer from inconsistent beliefs without revising? *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1449–1455, 1995.

[Bondarenko *et al.*, 1997] A. Bondarenko, P.M. Dung, R.A. Kowalski & F. Toni, An abstract argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93:63–101, 1997.

[Brewka, 1989] G. Brewka, Preferred subtheories: an extended logical framework for default reasoning. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1043–1048, 1989.

[Brewka, 1991] G. Brewka, *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge: Cambridge University Press, 1991.

[Brewka, 1994a] G. Brewka, Reasoning about priorities in default logic. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 940–945, 1994.

[Brewka, 1994b] G. Brewka, A logical reconstruction of Rescher's theory of formal disputation based on default logic. *Proceedings of the 11th European Conference on Artificial Intelligence*, 366–370, 1994.

[Brewka, 1996] G. Brewka, Well-founded semantics for extended logic programs with dynamic preferences. *Journal of Artificial Intelligence Research* 4:19–30, 1996.

[Cayrol, 1995] C. Cayrol, On the relation between argumentation and non-monotonic coherence-based entailment. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1443–1448, 1995.

[Chesñevar *et al.*, 1999] C.I. Chesñevar, A.G. Maguitman & R.P. Loui, Logical models of argument. *Submitted*.

[Clark, 1990] P. Clark, Representing knowledge as arguments: Applying expert system technology to judgemental problem-solving. In *Research and Development in Expert Systems VII*, eds. T. R. Addis and R. M. Muir, 147–159. Cambridge University Press, 1990.

[Das *et al.*, 1996] S. Das, J. Fox, & P. Krause, A unified framework for hypothetical and practical reasoning (1): theoretical foundations. *Proceedings of the International Conference on Formal and Applied Practical Reasoning (FAPR'96)*. Lecture Notes in Artificial Intelligence 1085, 58–72. Berlin: Springer Verlag, 1996.

[De Kleer, 1986] J. De Kleer, An assumption-based TMS. *Artificial Intelligence* 28:127-162, 1986.

[Delgrande, 1988] J. Delgrande, An approach to default reasoning based on a first-order conditional logic: revised report. *Artificial Intelligence* 36:63–90, 1988.

[Doyle, 1979] J. Doyle, Truth Maintenance Systems. *Artificial Intelligence* 12:231-272, 1979.

[Dung, 1993] P.M. Dung, An argumentation semantics for logic programming with explicit negation. *Proceedings of the Tenth Logic Programming Conference*, 616–630. Cambridge, MA: MIT Press, 1993.

[Dung, 1994] P.M. Dung, Logic programming as dialogue games. Report Division of Computer Science, Asian Institute of Technology, Bangkok, 1994.

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

[Dung *et al.*, 1996] P.M. Dung, R.A. Kowalski & F. Toni, Synthesis of proof procedures for default reasoning. *Proceedings International Workshop on Logic Program Synthesis and Transformation (LOPSTR'96)*, ed. J. Gallagher. Lecture Notes in Computer Science 1207, 313–324. Berlin: Springer Verlag, 1996.

[Dung *et al.*, 1997] P.M. Dung, R.A. Kowalski & F. Toni, Argumentation-theoretic proof procedures for default reasoning. Report Department of Computing, Imperial College London, 1997.

[Elvang-Gøransson & Hunter, 1995] M. Elvang-Gøransson & A. Hunter, Argumentative logics: reasoning with classically inconsistent information. *Data & Knowledge Engineering* 16:125–145, 1995.

[Freeman & Farley, 1996] K. Freeman & A.M. Farley, A model of argumentation and its application to legal reasoning. *Artificial Intelligence and Law* 4:163–197, 1996. Reprinted in [Prakken & Sartor, 1997a].

[Gabbay, 1985] D.M. Gabbay, Theoretical Foundations for Non-monotonic Reasoning in Expert Systems, in: *Logics and Models of Concurrent Systems*, ed. K.R. Apt, 439–457. Berlin, Springer-Verlag, 1985.

[Gabbay *et al.*, 1994] D.M. Gabbay, C.J. Hogger & J.A. Robinson, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford: Oxford University Press, 1994.

[Garcia *et al.*, 1998] A.J. Garcia, G.R. Simari & C.I. Chesñevar, An argumentative framework for reasoning with inconsistent and incomplete information. *Proceedings of the ECAI'98 Workshop on Practical Reasoning and Rationality*, Brighton, UK, 1998.

[Geffner, 1991] H. Geffner, Beyond negation as failure. *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, 218–229. San Mateo, CA: Morgan Kaufmann Publishers Inc., 1991.

[Geffner & Pearl, 1992] H. Geffner & J. Pearl, Conditional entailment: bridging two approaches to default reasoning. *Artificial Intelligence* 53:209–244, 1992.

[Goodman, 1954] N. Goodman, *Fact, Fiction, and Forecast.* Cambridge, MA: Harvard University Press, 1954.

[Gordon, 1995] T.F. Gordon, *The Pleadings Game. An Artificial Intelligence Model of Procedural Justice*. Dordrecht etc.: Kluwer Academic Publishers, 1995.

[Gordon & Karacapilidis, 1997] T.F. Gordon & N. Karacapilidis, The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, 10–18. New York: ACM Press, 1997.

[Grosof, 1993] B.N. Grosof, Prioritizing multiple, contradictory sources in common-sense learning by being told; or, advice-taker meets bureaucracy. *Proceedings Common Sense '93: The Second Symposium on Logical formalisations of Common-Sense Reasoning*, Austin, Texas, 1993.

[Hage, 1997] J.C. Hage, *Reasoning With Rules. An Essay on Legal Reasoning and Its Underlying Logic*. Dordrecht etc.: Kluwer Law and Philosophy Library, 1997.

[Hage *et al.*, 1994] J.C. Hage, R. Leenes & A.R. Lodder, Hard cases: a procedural approach. *Artificial Intelligence and Law* 2:113–166, 1994.

[Hamblin, 1971] C.L. Hamblin, Mathematical models of dialogue. *Theoria* 37:130–155, 1971.

[Hanks & McDermott, 1987] S. Hanks & D. McDermott, Nonmonotonic Logic and Temporal Projection. *Artificial Intelligence* 33:379–412, 1987.

[Hart, 1949] H.L.A. Hart, The ascription of responsibility and rights. *Proceedings of the Aristotelean Society*, n.s. 49 (1948-9), 171–194. Reprinted in *Logic and Language. First Series*, ed. A.G.N. Flew, 145–166. Oxford: Basil Blackwell, 1951.

[Horty *et al.*, 1990] J.F. Horty, R.H. Thomasson & D.S. Touretzky, A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence* 42:311–348, 1990.

[Hunter, 1993] A. Hunter, Using priorities in non-monotonic proof theory. Report Department of Computing, Imperial College London, 1993.

[Jakobovits & Vermeir, 1999] H. Jakobovits, & D. Vermeir, Robust Semantics for Argumentation Frameworks. *Journal of Logic and Computation* 9:215–262, 1999.

[Kakas *et al.*, 1994] A.C. Kakas, P. Mancarella & P.M. Dung, The acceptability semantics for logic programs. *Proceedings of the Eleventh International Conference on Logic Programming*, 509–514. Cambridge, MA: MIT Press, 1994.

[Kakas & Toni, 1999] A.C. Kakas & F. Toni, Computing argumentation in logic programming. *Journal of Logic and Computation* 9:515–562, 1999.

[Konolige, 1988] K. Konolige, Defeasible argumentation in reasoning about events. In *Methodologies for Intelligent Systems*, eds. Z.W. Ras and L. Saitta, 380–390. Amsterdam: Elsevier, 1988.

[Kowalski & Toni, 1996] R.A. Kowalski & F. Toni, Abstract argumentation. *Artificial Intelligence and Law* 4:275–296, 1996. Reprinted in [Prakken & Sartor, 1997a].

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann & M. Magidor, Nonmonotonic reasoning, preferential models, and cumulative logics. *Artificial Intelligence* 44:167–207, 1990.

[Krause *et al.*, 1995] P. Krause, S.J. Ambler, M. Elvang-Gøransson & J. Fox, A logic of argumentation for uncertain reasoning. *Computational Intelligence* 11:113–131, 1995.

[Lewis, 1973] D.K. Lewis, *Counterfactuals*. Cambridge, MA: Harvard University Press, 1973.

[Lin & Shoham, 1989] F. Lin & Y. Shoham, Argument systems. A uniform basis for nonmonotonic reasoning. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 245–255. San Mateo, CA: Morgan Kaufmann Publishers Inc, 1989.

[Lodder, 1999] A.R. Lodder, *DiaLaw. On Legal Justification and Dialog Games*. To appear in Kluwer's Law and Philosophy Library, 1999.

[Loui, 1987] R.P. Loui, Defeat among arguments: a system of defeasible inference. *Computational Intelligence* 2:100–106, 1987.

[Loui, 1995] R.P. Loui, Hart's critics on defeasible concepts and ascriptivism. *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, 21–30. New York: ACM Press, 1995.

[Loui, 1998] R.P. Loui, Process and policy: resource-bounded non-demonstrative reasoning. *Computational Intelligence* 14:1–38, 1998.

[Loui *et al.*, 1993] R.P. Loui, J. Norman, J. Olson & A. Merrill, A design for reasoning with policies, precedents, and rationales. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, 202–211. New York: ACM Press, 1993.

[Loui & Norman, 1995] R.P. Loui & J. Norman, Rationales and argument moves. *Artificial Intelligence and Law* 3:159–189, 1995.

[Mackenzie, 1979] J.D. Mackenzie, Question-begging in non-cumulative systems. *Journal of Philosophical Logic* 8:117–133, 1979.

[Makinson, 1989] D. Makinson, General Theory of Cumulative Inference, *Proceedings of the 2nd Workshop on Nonmonotonic Reasoning*, eds. M. Reinfrank *et al.*, Lecture Notes in Artificial Intelligence 346, 1–18. Berlin: Springer Verlag, 1989.

[Makinson & Schlechta, 1991] D. Makinson & K. Schlechta, Floating conclusions and zombie paths: two deep difficulties in the 'directly sceptical' approach to inheritance nets. *Artificial Intelligence* 48:199–209, 1991.

[Marek *et al.*, 1990] W. Marek, A. Nerode & J. Remmel, A theory of non-monotonic rule systems I. *Annals of Mathematics and Artificial Intelligence* 1:241–273, 1990.

[Marek *et al.*, 1992] W. Marek, A. Nerode & J. Remmel, A theory of non-monotonic rule systems II. *Annals of Mathematics and Artificial Intelligence* 5:229–263, 1992.

[Martins & Reinfrank, 1991] J.P. Martins & M. Reinfrank (eds.), *Truth Maintenance Systems*. Springer Lecture Notes in Artificial Intelligence, 515, Berlin: Springer Verlag, 1991.

[McCarthy *et al.*, 1969] J. McCarthy & P.J. Hayes, Some Philosophical Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence* 4, eds. B. Meltzer *et al.*, 463–502. Edinburgh University Press, 1969.

[Nute, 1994] D.N. Nute, Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Nonmonotonic Reasoning and Uncertain Reasoning*, eds. D.M. Gabbay, C.J. Hogger & J.A. Robinson, 355–395. Oxford: Oxford University Press, 1994.

[Nute, 1997] D.N. Nute, Apparent obligation. In *Defeasible Deontic Logic*, ed. D.N. Nute, 287–315. Dordrecht etc.: Kluwer Synthese Library, 1997.

[Nute & Erk, 1995] D.N. Nute & K. Erk, Defeasible logic. Report AI Center, University of Georgia, Athens, GA, 1995.

[Parsons *et al.*, 1998] S. Parsons, C. Sierra & N.R. Jennings, Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8:261–292, 1998.

[Pollock, 1970] J.L. Pollock, The Structure of Epistemic Justification. *American Philosophical Quarterly*, monograph series, vol. 4, 62–78, 1970.

[Pollock, 1974] J.L. Pollock, *Knowledge and Justification*. Princeton: Princeton University Press, 1974.

[Pollock, 1987] J.L. Pollock, Defeasible reasoning. *Cognitive Science* 11:481–518, 1987.

[Pollock, 1991] J.L. Pollock, A theory of defeasible reasoning. *International Journal of Intelligent Systems* 6:33–54, 1991.

[Pollock, 1992] J.L. Pollock, How to reason defeasibly. *Artificial Intelligence* 57:1–42, 1992.

[Pollock, 1995] J.L. Pollock, *Cognitive Carpentry. A Blueprint for How to Build a Person*. Cambridge, MA: MIT Press, 1995.

[Poole, 1985] D.L. Poole, On the comparison of theories: Preferring the most specific explanation. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 144–147, 1985.

[Poole, 1988] D.L. Poole, A logical framework for default reasoning. *Artificial Intelligence* 36:27–47, 1988.

[Prakken, 1993] H. Prakken, An argumentation framework in default logic. *Annals of Mathematics and Artificial Intelligence* 9:91–132, 1993.

[Prakken, 1995] H. Prakken, A semantic view on reasoning about priorities (extended abstract). *Proceedings of the Second Dutch/German Workshop on Nonmonotonic Reasoning*, Utrecht, 152–159, 1995.

[Prakken, 1997] H. Prakken, *Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoning in Law*. Dordrecht etc.: Kluwer Law and Philosophy Library, 1997.

[Prakken, 1999] H. Prakken, Dialectical proof theory for defeasible argumentation with defeasible priorities (preliminary report). To appear in *Proceedings of the 4th ModelAge Workshop 'Formal Models of Agents'*, Springer Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag, 1999.

[Prakken & Sartor, 1996] H. Prakken & G. Sartor, A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law* 4:331–368, 1996. Reprinted in [Prakken & Sartor, 1997a].

[Prakken & Sartor, 1997a] H. Prakken & G. Sartor, (eds.) 1997a. *Logical Models of Legal Argument.* Dordrecht etc.: Kluwer Academic Publishers, 1997. (reprint of *Artificial Intelligence and Law* 4, 1996).

[Prakken & Sartor, 1997b] H. Prakken & G. Sartor, Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7:25–75, 1997.

[Prakken & Sartor, 1998] H. Prakken & G. Sartor, Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law* 6:231–287, 1998.

[Reiter, 1980] R. Reiter, A logic for default reasoning. *Artificial Intelligence* 13:81–132, 1980.

[Rescher, 1976] N. Rescher, *Plausible Reasoning*. Assen: Van Gorcum, 1976.

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

[Ross, 1930] W.D. Ross, *The Right and the Good*. Oxford: Oxford University Press, 1930.

[Sartor, 1994] G. Sartor, A formal model of legal argumentation. *Ratio Juris* 7:212–226, 1994.

[Shoham, 1988] Y. Shoham, *Reasoning about Change. Time and Causation from the Standpoint of Artificial Intelligence*. Cambridge, MA: MIT Press, 1988.

[Simari *et al.*, 1994] G.R. Simari, C.I. Chesñevar & A.J. Garcia, The role of dialectics in defeasible argumentation. *Proceedings of the XIV International Conference of the Chilean Computer Science Society*, Concepción, Chile, 1994.

[Simari & Loui, 1992] G.R. Simari & R.P. Loui, A mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence* 53:125–157, 1992.

[Starmans, 1996] R.J.C.M. Starmans, *Logic, Argument, and Commonsense*. Doctoral Dissertation, Tilburg University, 1996.

[Thielscher, 1996] M. Thielscher, A nonmonotonic disputation-based semantics and proof procedure for logic programs. *Proceedings of the Joint International Conference and Symposium on Logic Programming*, 483–497. Cambridge, MA: MIT Press, 1996.

[Toulmin, 1958] S.E. Toulmin, *The Uses of Argument*. Cambridge: Cambridge University Press, 1958.

[Verheij, 1996] B. Verheij, *Rules, Reasons, Arguments. Formal Studies of Argumentation and Defeat*. Doctoral Dissertation, University of Maastricht, 1996.

[Vreeswijk, 1989] G.A.W. Vreeswijk, The Feasibility of Defeat in Defeasible Reasoning, *Proceedings of the Second International Conference on Knowledge Representation and Reasoning*, 526–534. San Mateo, CA: Morgan Kaufmann Publishers Inc., 1991. Also published in *Diamonds and Defaults*, Studies in Language, Logic, and Information, Vol. 1, 359–380. Dordrecht: Kluwer, 1993.

[Vreeswijk, 1993a] G.A.W. Vreeswijk, *Studies in Defeasible Argumentation*. Doctoral dissertation, Department of Computer Science, Free University Amsterdam, 1993.

[Vreeswijk, 1993b] G.A.W. Vreeswijk, Defeasible dialectics: a controversy-oriented approach towards defeasible argumentation. *Journal of Logic and Computation* 3:317–334, 1993.

[Vreeswijk, 1995] G.A.W. Vreeswijk, The computational value of debate in defeasible reasoning. *Argumentation* 9:305–342, 1995.

[Vreeswijk, 1997] G.A.W. Vreeswijk, Abstract argumentation systems. *Artificial Intelligence* 90:225–279, 1997.

[Vreeswijk, 1999] G.A.W. Vreeswijk, Representation of formal dispute with a standing order. To appear in *Artificial Intelligence and Law*, 1999.

[Walton & Krabbe, 1995] D.N. Walton & E.C.W. Krabbe, *Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning*. Albany, NY: State University of New York Press, 1995.