Argumentation Semantics for Defeasible Logic

Guido Governatori
School of Information Technology and Electrical Engineering,
The University of Queensland,
Brisbane, QLD 4072, Australia
email: guido@itee.uq.edu.au

Micheal J. Maher
Department of Computer Science, Loyola University Chicago,
6525 N. Sheridan Road, Chicago, IL 60626 USA
email: mjm@cs.luc.edu

Grigoris Antoniou
Department of Computer Science, University of Crete,
P.O. Box 2208, Heraklion, Crete, GR-71409, Greece,
email: ga@csd.uoc.gr

David Billington
School of Computing and Information Technology,
Griffith University,
Nathan, QLD 4111, Australia,
email: db@cit.gu.edu.au

Abstract

Defeasible reasoning is a simple but efficient rule-based approach to nonmonotonic reasoning. It has powerful implementations and shows promise to be applied in the areas of legal reasoning and the modeling of business rules. This paper establishes significant links between defeasible reasoning and argumentation. In particular, Dung-like argumentation semantics is provided for two key defeasible logics, of which one is ambiguity propagating and the other ambiguity blocking.

There are several reasons for the significance of this work: (a) establishing links between formal systems leads to a better understanding and cross-fertilization, in particular our work sheds light on the argumentation-theoretic features of defeasible logic; (b) we provide the first ambiguity blocking Dung-like argumentation system; (c) defeasible reasoning may provide an efficient implementation platform for systems of argumentation; and (d) argumentation-based semantics support a deeper understanding of defeasible reasoning, especially in the context of the intended applications.

Journal of Logic and Computation, 14 (5): 675–702.

The original publication is available at doi: 10.1093/logcom/14.5.675.

1 Introduction

Defeasible reasoning [31, 32] supports rule-based reasoning where rules may be defeated by other rules that support a contrary conclusion. The concept of *defeat* lies at the heart of defeasible reasoning. Where conflicts between rules arise, priorities can be used to resolve these conflicts.

Defeasible reasoning was developed to support practical nonmonotonic reasoning. Recently it has been proposed as an appropriate language for executable regulations [4], contracts [36], business rules [19], e-commerce [13], automated negotiation [14], and policy-based intentions [17].

The starting point in our considerations is the classical Defeasible Logic of [31] in the formalization of [6]. Unlike other nonmonotonic approaches, Defeasible Logic was designed with implementation in mind. In fact, recently very powerful implementations of defeasible logic became available, capable of handling 100,000s of defeasible rules [30]. The logic has been shown to have linear complexity [26].

In previous work we developed a framework for the definition of variants of Defeasible Logic [29, 2]; this framework allows us to "tune" defeasible logics in order to obtain a logic with desired properties. The issue of whether non-monotonic logics and, in particular, inheritance networks should be ambiguity blocking or ambiguity propagating has been the subject of considerable discussion (see, for example, [39, 38]). The original defeasible logic is ambiguity blocking, but we can also define an ambiguity propagating defeasible logic [2]. Most of the logics described in [2], including this ambiguity propagating defeasible logic, have been implemented in the *Deimos* system.[1] These two defeasible logics will be the focus of this paper.

Argumentation has long been used to study defeasible reasoning [8], and recently abstract argumentation frameworks have been developed [12, 40] to support the characterization of non-monotonic reasoning in argumentation-theoretic terms. The basic elements of these frameworks are the notions of arguments and "acceptability" of an argument. Briefly, an argument is acceptable if it is possible to show that it is not possible to rebut it with stronger arguments. Although defeasible logics can be described informally in terms of arguments, the logics have been formalized in a proof-theoretic setting in which arguments play no role.

Dung [11, 12] presented an abstract argumentation framework, and [7] showed that several well-known nonmonotonic reasoning systems are concrete instances of the abstract framework. Unfortunately, so far only one sceptical argumentation semantics (called grounded semantics) has been put forward. In this paper we will adapt Dung's framework to provide argumentation semantics for the two defeasible logics we investigate. We show that Dung's grounded semantics characterizes the ambiguity propagating defeasible logic. For the original (ambiguity blocking) defeasible logic, we modify Dung's notion of acceptability to give an argumentation characterization of this logic.

This work is part of our ongoing effort to establish close connections between defeasible reasoning and other formulations of non-monotonic reasoning. Such connections usually lead to a better understanding of each area, and cross-fertilization. Moreover the elegance of the correspondence we establish in this instance suggests that defeasible reasoning and argumentation are conceptually closely linked.

The significance of this paper to defeasible reasoning lies in the elegant argumentation-theoretic semantics we develop. In comparison, the proof theory defining the defeasible

[1] www.cit.gu.edu.au/$\sim$arock/defeasible/Defeasible.cgi

logics is clumsy. The argumentation-theoretic semantics will prove useful in the intended applications of defeasible logic mentioned above, where arguments are a natural feature of the problem domain.

The study of argumentation will also benefit from our work. For one, our argumentation semantics of classical defeasible logic provides an ambiguity blocking argumentation system, to our knowledge the first one. In addition, we admit infinite chains of reasoning as arguments, whereas most argumentation systems permit only finite arguments. We also characterize an underlying Kunen semantics of failure-to-prove [24] in argumentation-theoretic terms. Technically, we admit infinite arguments in our argumentation framework to achieve this characterization.

Furthermore, usually argumentation is studied theoretically, with not so much emphasis placed on implementation. On the other hand, there are already very powerful systems of defeasible reasoning. Thus our research may lead to the implementation of argumentation systems on the basis of defeasible reasoning.

This paper is structured as follows. In the next section we provide a brief introduction to defeasible logics. We then provide our argumentation-theoretic semantics for defeasible logic and an ambiguity propagating variant with the appropriate soundness and completeness results in Section 3, which is the main part of this paper. Related work is discussed in Section 4. All proofs may be found in an appendix at the end of the paper.

2 Overview of Defeasible Logics

We begin by presenting the basic ingredients of defeasible logic. A defeasible theory contains four different kinds of knowledge: strict rules, defeasible rules, defeaters, and a superiority relation. We consider only essentially propositional rules. Rules containing free variables are interpreted as the set of their ground instances.

Strict rules are rules in the classical sense: whenever the premises are indisputable (e.g. facts) then so is the conclusion. An example of a strict rule is "Emus are birds". Written formally: $emu(X) \rightarrow bird(X)$. Strict rules with an empty body represent indisputable statements called facts. An example is "Tweety is an emu". Written formally: $\rightarrow emu(tweety)$.

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is "Birds typically fly"; written formally: $bird(X) \Rightarrow flies(X)$. The idea is that if we know that something is a bird, then we may conclude that it flies, *unless there is other evidence suggesting that it may not fly*.

Defeaters are rules that cannot be used to draw any conclusions. Their only use is to prevent some conclusions. In other words, they are used to defeat some defeasible rules by producing evidence to the contrary. An example is "If an animal is heavy then it might not be able to fly". Formally: $heavy(X) \leadsto \neg flies(X)$. The main point is that the information that an animal is heavy is not sufficient evidence to conclude that it doesn't fly. It is only evidence against the conclusion that a heavy animal flies. In other words, we don't wish to conclude $\neg flies$ if $heavy$, we simply want to prevent a conclusion $flies$.

The *superiority relation* among rules is used to define priorities among rules, that is, where one rule may override the conclusion of another rule. For example, given the facts

$$\rightarrow bird$$
$$\rightarrow brokenWing$$

and the defeasible rules

$$\begin{array}{rrl} r: & bird & \Rightarrow flies \\ r': & brokenWing & \Rightarrow \neg flies \end{array}$$

which contradict one another, no conclusive decision can be made about whether a bird with a broken wing can fly. But if we introduce a superiority relation $>$ with $r' > r$, then we can indeed conclude that the bird cannot fly. The superiority relation is required to be acyclic.

In this paper we disregard the superiority relation to keep the discussion and the technicalities simple. This restriction does not affect the generality of our approach: In [1] we gave a modular transformation that empties the superiority relation while maintaining the same conclusions in the language of the original theory. That result was proven for our original (ambiguity blocking) defeasible logic, but subsequent work proved its correctness also for the ambiguity propagating defeasible logic we will be considering. The previous example about birds (with the relation $r' > r$) is transformed into the following equivalent theory with an empty superiority relation:

$$\begin{array}{rlcrl} bird & \Rightarrow \neg inf(r) & & & \rightarrow bird \\ \neg inf(r) & \Rightarrow flies & & & \rightarrow brokenWing \\ brokenWing & \Rightarrow \neg inf(r') & & & \\ \neg inf(r') & \Rightarrow \neg flies & & \neg inf(r') & \Rightarrow inf(r) \end{array}$$

The compilation of priorities into a rule set is used in other non-monotonic reasoning approaches, too, for example in default logic [9]. The intuition behind the compilation used in defeasible logic is to split each defeasible rule r into two rules connected by an $\mathit{inf}(r)$ literal, where $\mathit{inf}(r)$ expresses the idea that rule r is overruled by a (strict or defeasible) superior rule. Accordingly $\neg \mathit{inf}(r)$ means that r is not inferior to any applicable (strict of defeasible) rule. Finally, for each pair of rules for which the superiority relation obtains, we introduce a defeasible rule whose interpretation is that if the stronger rule is applicable and not overruled then the weaker rule is overruled. For the full explanation of this and related transformations see [1].

Now we present the defeasible logics formally. A *rule* r consists of its *antecedents* (or *body*) $A(r)$ which is a finite set of literals, an arrow, and its *consequent* (or *head*) $C(r)$ which is a literal. There are three kinds of arrows, $\rightarrow$, $\Rightarrow$ and $\rightsquigarrow$ which correspond, respectively, to strict rules, defeasible rules and defeaters. Where the body of a rule is empty or consists of one formula only, set notation may be omitted in examples.

Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of strict and defeasible rules in R by R_{sd}, the set of defeasible rules in R by R_d, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q. If q is a literal, $\sim q$ denotes the complementary literal (if q is a positive literal p then $\sim q$ is $\neg p$; and if q is $\neg p$, then $\sim q$ is p).

A *defeasible theory* D is a finite set of rules R. A *conclusion* of D is a tagged literal; in our original defeasible logic there are two tags, ∂ and Δ, that may have positive or negative polarity (further tags for defeasible logic variants will be introduced shortly):

$+\Delta q$ which is intended to mean that q is definitely provable in D (i.e., using only strict rules).

$-\Delta q$ which is intended to mean that it is proved that q is not definitely provable in D.

$+\partial q$ which is intended to mean that q is defeasibly provable in D.

$-\partial q$ which is intended to mean that it is proved that q is not defeasibly provable in D.

Provability is based on the concept of a *derivation* (or *proof*) in $D = R$. A derivation is a finite sequence $P = (P(1), \ldots, P(n))$ of tagged literals satisfying four conditions (which correspond to inference rules for each of the four kinds of conclusion). In the following $P(1..i)$ denotes the initial part of the sequence P of length i.

$+\Delta$:
If $P(i+1) = +\Delta q$ then
$\quad \exists r \in R_s[q]$
$\quad\quad \forall a \in A(r) : +\Delta a \in P(1..i)$

$-\Delta$:
If $P(i+1) = -\Delta q$ then
$\quad \forall r \in R_s[q]$
$\quad\quad \exists a \in A(r) : -\Delta a \in P(1..i)$

The definition of Δ describes just forward chaining of strict rules. For a literal q to be definitely provable we need to find a strict rule with head q, of which all antecedents have been definitely proved previously. And to establish that q cannot be proven definitely we must establish that for every strict rule with head q there is at least one antecedent which has been shown to be non-provable.

Now we turn to the more complex case of defeasible provability.

$+\partial$:
If $P(i+1) = +\partial q$ then either
(1) $+\Delta q \in P(1..i)$ or
$\quad$ (2.1) $\exists r \in R_{sd}[q] \forall a \in A(r)$
$\quad\quad\quad +\partial a \in P(1..i)$ and
$\quad$ (2.2) $-\Delta{\sim}q \in P(1..i)$ and
$\quad$ (2.3) $\forall s \in R[{\sim}q]$
$\quad\quad\quad \exists a \in A(s) : -\partial a \in P(1..i)$

$-\partial$:
If $P(i+1) = -\partial q$ then
(1) $-\Delta q \in P(1..i)$ and
$\quad$ (2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r)$:
$\quad\quad\quad -\partial a \in P(1..i)$ or
$\quad$ (2.2) $+\Delta{\sim}q \in P(1..i)$ or
$\quad$ (2.3) $\exists s \in R[{\sim}q]$ such that
$\quad\quad\quad \forall a \in A(s) : +\partial a \in P(1..i)$

Let us work through the condition for $+\partial$. To show that q is provable defeasibly we have two choices: (1) We show that q is already definitely provable; or (2) we need to argue using the defeasible part of D as well. In particular, we require that there must be a strict or defeasible rule with head q which can be applied (2.1). But now we need to consider possible "attacks", that is, reasoning chains in support of ${\sim}q$. To be more specific: to prove q defeasibly we must show that ${\sim}q$ is not definitely provable (2.2). And finally (2.3), we need to show that all rules with head ${\sim}q$ are inapplicable.

In [2] we presented a framework for defeasible logic, where we showed how to tune defeasible logic in order to define variants able to deal with different nonmonotonic phenomena. In particular, we proposed different ways in which conclusions can be obtained. One of the properties most discussed in the literature is whether ambiguities should be propagated or blocked (see, for example, [39, 38]).

We illustrate the notion of ambiguity with the following example

Example 1 Consider the following defeasible theory D.

$$\Rightarrow a \qquad\qquad \Rightarrow b$$
$$\Rightarrow \neg a \qquad\qquad a \Rightarrow \neg b$$

Here a is ambiguous since we have applicable rules for both a and $\neg a$, and we have no means to decide between them. In a setting where the ambiguity is blocked, b is not ambiguous because we have an applicable rule for b and, at the same time, the rule for $\neg b$ is not applicable since we cannot prove its antecedent. On the other hand, in an ambiguity propagating setting, b is ambiguous because there are rules for both b and $\neg b$ antecedent of the rule for $\neg b$ is ambiguous, and hence the ambiguity is propagated to b. We have proofs in this theory for $-\partial a$, $-\partial\neg a$, $+\partial b$, and $-\partial\neg b$, thus showing the ambiguity blocking behavior of Defeasible Logic.

In the logic above ambiguities are blocked. In the following we introduce an ambiguity propagating variant. The result of [1] has been extended to this variant; thus, once again, the appropriate inference rules will be presented in simplified form without reference to the superiority relation.

The first step is to determine when a literal is "supported" in a defeasible theory D. Support for a literal p ($+\Sigma p$) consists of a monotonic chain of reasoning that would lead us to conclude p in the absence of conflicts. This leads to the following inference conditions:

$+\Sigma$:	$-\Sigma$:
If $P(i+1) = +\Sigma p$ then	If $P(i+1) = -\Sigma p$ then
$\exists r \in R_{sd}[p]$:	$\forall r \in R_{sd}[p]$:
$\forall a \in A(r) : +\Sigma a \in P(1..i)$	$\exists a \in A(r) : -\Sigma a \in P(1..i)$

A literal that is defeasibly provable is supported, but a literal may be supported even though it is not defeasibly provable. Thus support is a weaker notion than defeasible provability. For example, given two rules $\Rightarrow p$ and $\Rightarrow \neg p$, both p and $\neg p$ are supported, but neither is defeasibly provable. We say that p is ambiguous. In general, a literal is *ambiguous* if there is a chain of reasoning that supports a conclusion that p is true, and another that supports that $\sim p$ is true.

We can achieve ambiguity propagation behavior by making a minor change to the inference condition for $+\partial$: instead of requiring that every attack on p be inapplicable in the sense of $-\partial$, now we require that the rule for $\sim p$ be inapplicable because one of its antecedents cannot be *supported*. By making attack easier we are imposing a stronger condition for proving a literal defeasibly. Here is the formal definition:

$+\partial_{ap}$:	$-\partial_{ap}$:
If $P(i+1) = +\partial_{ap} q$ then either	If $P(i+1) = -\partial_{ap} q$ then
(1) $+\Delta q \in P(1..i)$ or	(1) $-\Delta q \in P(1..i)$ and
(2.1) $\exists r \in R_{sd}[q] \forall a \in A(r)$: $+\partial_{ap} a \in P(1..i)$ and	(2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r)$: $-\partial_{ap} a \in P(1..i)$ or
(2.2) $-\Delta\sim q \in P(1..i)$ and	(2.2) $+\Delta\sim q \in P(1..i)$ or
(2.3) $\forall s \in R[\sim q]$ $\exists a \in A(s) : -\Sigma a \in P(1..i)$	(2.3) $\exists s \in R[\sim q]$ such that $\forall a \in A(s) : +\Sigma a \in P(1..i)$

EXAMPLE 1 (continued)
We consider the defeasible theory of Example 1, but this time we compute the consequences using the conditions given above; we have $+\Sigma a$, $+\Sigma\neg a$, $+\Sigma b$ and $+\Sigma\neg b$ showing that there are chains of reasoning supporting a, $\neg a$, b and $\neg b$. Moreover we can derive $-\partial_{ap} a$, $-\partial_{ap}\neg a$, $-\partial_{ap} b$ and $-\partial_{ap}\neg b$ showing that the resulting logic exhibits an ambiguity propagating behavior. In fact b is now ambiguous, and its ambiguity depends on the ambiguity of a.

We present now an hypothetical scenario based on a legal proceeding –and whose structure frequently occurs in legal reasoning– where an ambiguity blocking argumentation framework seems more appropriate that the corresponding ambiguity propagation one[2].

Example 2 Let us suposse that a piece of evidence A suggests that the defendant is responsible while a second piece of evidence (let us call it B) indicates that he/she is not responsible; moreover the sources are equally reliable. According to the underlying legal system a defendant is presumed innocent (i.e., not guilty) unless responsibility has been proved.

The above scenario is encoded in the following defeasible theory:

$$\begin{array}{rclcrcl} & & & & & \Rightarrow & \neg guilty \\ evidenceA & \Rightarrow & responsible & \qquad & responsible & \Rightarrow & guilty \\ evidenceB & \Rightarrow & \neg responsible & & & & \end{array}$$

Given both $evidenceA$ and $evidenceB$, the literal $responsible$ is ambiguous. If we propagate ambiguity then the literals $guilty$ and $\neg guilty$ are ambiguous; thus an undisputed conclusion cannot be drawn. On the other hand, if we assume an ambiguity blocking stance, the literal $\neg guilty$ is not ambiguous and a definite verdict can be reached.

The above example shows that in domains where arguments are part of larger arguments and definite conclusions must be reached, which is often the case in the legal domain, ambiguity blocking systems offer more natural and intuitive representations than the corresponding ambiguity propagation ones. For a thorough discussion of various types of arguments, their applications and motivations see [35].

To conclude this section we notice that the inference conditions for defeasible logic closely resemble the inference mechanism of Prolog. In [29, 2] we have introduced a family of meta-programs for various variants of defeasible logic and we proved that defeasible logic is characterized by Kunen semantics. The meta-programs corresponding to the variants discussed in this paper are given in Appendix B.

3 Argumentation for Defeasible Logics

In this section we give the formal definition of an argumentation framework, and we describe in details two variants; the first capturing ambiguity propagation and the second ambiguity blocking. Moreover we prove that the two variants of defeasible logic presented in the previous section are sound and complete w.r.t the appropriate version of the semantics (Theorems 14 and 17).

3.1 Arguments

Argumentation systems usually contain the following basic elements: an underlying *logical language*, and the definitions of: *argument*, *conflict between arguments*, and the *status of arguments*. The latter elements are often used to define a consequence relation. In what follows we present an argumentation system containing the above elements in a way appropriate for defeasible logic.

Obviously, the underlying logical language we use is the language of defeasible logic.

[2]A similar structure is present in Example 3.11 of [21] about a medical procedure.

As usual, arguments are defined to be proof trees (or monotonic derivations). However, defeasible logic requires a more general notion of proof tree that admits infinite trees, so that the distinction is kept between an unrefuted, but infinite, chain of reasoning and a refuted chain.

An *argument* for a literal p based on a set of rules R is a (possibly infinite) tree with nodes labeled by literals such that the root is labeled by p and for every node with label h:

1. If $b_1, \ldots, b_n$ label the children of h then there is a rule in R with body $b_1, \ldots, b_n$ and head h.
2. If this rule is a defeater then h is the root of the argument.
3. The arcs in a proof tree are labeled by the rules used to obtain them.

Although condition 3 is required formally, to distinguish between rules with different arrows, we will not employ it in our examples since there is no chance of this confusion in our examples.

Condition 2 specifies that a defeater may only be used at the top of an argument; in particular, no chaining of defeaters is allowed. We illustrate this point by the following example.

Example 3 Consider the following defeasible theory D:

$$\begin{aligned} &\leadsto a \\ &a \leadsto b \\ &\Rightarrow \neg b \end{aligned}$$

Then $\leadsto a \leadsto b$ is not an argument (from now on we often use this linear, more compact representation of arguments that have one branch only, instead of a tree based representation). The reason is that, as we said before, defeaters are only used to prevent conclusions, but do not provide positive evidence. In our example, we have evidence *against* $\neg a$ (by the first defeater), but no evidence for a. Therefore the second defeater cannot be used since to do so we would need evidence for a. The proof theory of defeasible logic was defined in agreement with this reading, therefore $D \vdash +\partial\neg b$ and $D \vdash +\partial_{ap}\neg b$.

Given a defeasible theory D, the set of arguments that can be generated from D is denoted by $Args_D$.

Any literal labeling a node of an argument A is called a *conclusion* of A. However, when we refer to *the* conclusion of an argument, we refer to the literal labeling the root of the argument. A *(proper) subargument* of an argument A is a (proper) subtree of the proof tree associated to A.

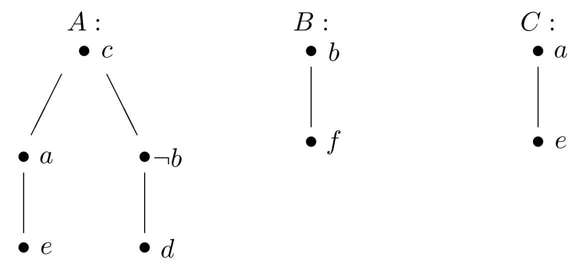
Sometimes we need to differentiate between arguments, depending on the rules used.

- A *supportive argument* is a finite argument in which no defeater is used.
- A *strict argument* is an argument in which only strict rules are used.
- An argument that is not strict is called *defeasible*.

Example 4 Consider the following defeasible theory D.

$$\begin{array}{l} \Rightarrow d \\ \rightarrow e \\ \Rightarrow f \end{array} \qquad\qquad \begin{array}{rl} \{a, \neg b\} & \Rightarrow c \\ e & \rightarrow a \\ f & \leadsto b \\ d & \Rightarrow \neg b \end{array}$$

Now we consider the following arguments:



Then A is a supportive argument for c, but not a strict argument. B is an argument for b that is not supportive. C is a strict supportive argument for a.

3.2 Arguments and Monotonic Proofs

At this stage we can characterize the definite conclusions of defeasible logic in argumentation-theoretic terms.

Proposition 5 Let D be a defeasible theory and p be a literal.

1. $D \vdash +\Delta p$ iff there is a strict supportive argument for p in $Args_D$
2. $D \vdash -\Delta p$ iff there is no (finite or infinite) strict argument for p in $Args_D$

At the same time we are ready to characterize the connection between the notion of support in defeasible logic and the existence of arguments.

Proposition 6 Let D be a defeasible theory and p a literal.

1. $D \vdash +\Sigma p$ iff there is a supportive argument for p in $Args_D$.
2. $D \vdash -\Sigma p$ iff there is no (finite or infinite) argument ending with a supportive rule for p in $Args_D$.

Both propositions are natural since strict provability in defeasible logic, support in defeasible logic, and arguments are monotonic proofs where no conflicting rules, respectively arguments, are considered.

EXAMPLE 4 (continued)
For the theory D in Example 4 we have the following:

$D \vdash +\Delta a$

$D \vdash +\Sigma c$

$D \vdash -\Delta f$ (there is no strict rule with head f)

$D \vdash -\Sigma b$ (there is no strict or defeasible rule with head b)

It is straightforward to see that these results are in agreement with the existence or otherwise of arguments, as specified by the propositions above. Arguments C and A provide the agreement in the first two cases, while the non-existence of an appropriate rule to place at the top of an argument provides the agreement in the last two cases.

3.3 Conflicting Arguments: Attack and Undercut

Next we begin to study the interaction between defeasible arguments. Obviously it is possible that arguments support contradictory conclusions. In Example 4 the arguments $\Rightarrow f \leadsto b$ and $\Rightarrow d \Rightarrow \neg b$ are conflicting.

An argument A *attacks* a defeasible argument B if a conclusion of A is the complement of a conclusion of B, and that conclusion of B is not part of a strict subargument of B. A set of arguments S attacks a defeasible argument B if there is an argument A in S that attacks B.

EXAMPLE 4 (continued)
The arguments A and B attack each other.

A defeasible argument A is *supported* by a set of arguments S if every proper subargument of A is in S.

Despite the similarity of name, this concept is not directly related to support in defeasible logic, nor to supportive arguments/proof trees. Essentially the notion of supported argument is meant to indicate when an argument may have an active role in proving or preventing the derivation of a conclusion. The main difference between the above notions is that infinite arguments and arguments ending with defeaters can be supported (thus preventing some conclusions), while supportive proof trees are finite and do not contain defeaters (cf. Proposition 6).

A defeasible argument A is *undercut* by a set of arguments S if S supports an argument B attacking a proper non-strict subargument of A. That an argument A is undercut by S means that we can show that some premises of A cannot be proved if we accept the arguments in S.

It is worth emphasizing that the above definitions concern only defeasible arguments and subarguments; for strict arguments we stipulate that they cannot be undercut or attacked. This is in line with definite provability in defeasible logic, where conflicts among rules are disregarded.

EXAMPLE 4 (continued)
The argument A is undercut by the set $S = \{\Rightarrow f\}$ (where f should be read as a tree consisting only of its root which is labeled by $\Rightarrow f$):

- S supports the argument B;
- B attacks a proper subargument of A: $\Rightarrow d \Rightarrow \neg b$.

3.4 The Status of Arguments

The heart of an argumentation semantics is the notion of an *acceptable argument*. Based on this concept it is possible to define *justified arguments* and *justified conclusions*, conclusions that may be drawn even taking conflicts into account. Intuitively, an argument A is acceptable w.r.t. a set of arguments S if, once we accept S as valid arguments, we feel compelled to accept A as valid.

The notion of acceptable argument can be defined in various ways – two such ways will be used later to characterise ambiguity propagating and ambiguity blocking defeasible logic. For the moment we leave this notion open, as a parameter that may be instantiated in different ways:

> Given an argument A and a set S of arguments (to be thought of as arguments that have already been demonstrated to be justified), we assume the existence of the concept: A is *acceptable w.r.t.* S.

Based on this concept we proceed to define justified arguments and justified literals.

Definition 7 Let D be a defeasible theory. We define J_i^D as follows.

- $J_0^D = \emptyset$;
- $J_{i+1}^D = \{a \in Args_D \mid a \text{ is acceptable w.r.t. } J_i^D\}$.

The set of *justified arguments* in a defeasible theory D is $JArgs^D = \cup_{i=1}^{\infty} J_i^D$.

A literal p is *justified* if it is the conclusion of a supportive argument in $JArgs^D$.

That an argument A is justified means that it resists every reasonable refutation. However, defeasible logic is more expressive since it is able to say when a conclusion is demonstrably non-provable ($-\partial$, $-\partial_{ap}$). Briefly, that a conclusion is demonstrably non-provable means that every possible argument not involving defeaters has been refuted. In the following we show how to capture this notion in our argumentation system by assigning the status *rejected* to arguments that are refuted. Roughly speaking, an argument is rejected if it has a rejected subargument or it cannot overcome an attack from another argument.

Again there are several possible definitions for the notion of rejected argument. Similarly to what we have done for the notion of acceptable argument we leave it temporarily undefined.

> Given an argument A, a set S of arguments (to be thought of as arguments that have already been rejected), and a set T of arguments (to be thought of as justified arguments that may be used to support attacks on A), we assume the existence of the concept: A is *rejected by* S *and* T.

Based on this concept we proceed to define rejected arguments and rejected literals.

Definition 8 Let D be a defeasible theory and T be a set of arguments. We define $R_i^D(T)$ as follows.

- $R_0^D(T) = \emptyset$;
- $R_{i+1}^D(T) = \{a \in Args_D \mid a \text{ is rejected by } R_i^D(T) \text{ and } T\}$.

The set of *rejected arguments* in a defeasible theory D w.r.t. T is $RArgs^D(T) = \cup_{i=1}^{\infty} R_i^D(T)$. We say that an argument is *rejected* if it is rejected w.r.t. $JArgs^D$.

A literal p is *rejected* by T if there is no argument in $Args_D - RArgs^D(T)$, the top rule of which is a strict or defeasible rule with head p. A literal is rejected if it is rejected by $JArgs^D$.

Note that a literal p is not necessarily rejected if there is no supportive argument for p in $Args_D - RArgs^D(T)$, because there may be an infinite argument for p in $Args_D - RArgs^D(T)$ without any defeaters (recall that supportive arguments must be finite). Thus it is possible for a literal to be neither justified nor rejected. The situation is similar to defeasible logic, where we may have both $D \not\vdash +\partial p$ and $D \not\vdash -\partial p$. A sufficient condition that prevents this situation is the acyclicity of the atom dependency graph (see [6]).

The different definitions of acceptable and rejected that we will introduce in the following sections satisfy similar technical properties, and consequently the two argumentation systems have some similar properties. Later, in Section 3.7, we will further investigate similarities – and establish some differences – between the two systems. Here we focus on some common technical properties of the argumentation systems.

Lemma 9 The sequences of sets of arguments J_i^D and $R_i^D(T)$ are monotonically increasing.

Lemma 10 Every subargument of a justified argument is justified.

Lemma 11 Let A be an argument.

1. A is acceptable w.r.t. $JArgs^D$ iff $A \in JArgs^D$.
2. A is rejected by $RArgs^D(T)$ and T iff $A \in RArgs^D(T)$.

We will see later, in Theorem 18, that, for the concepts of acceptability and rejected that we investigate, no argument or literal is both justified and rejected.

3.5 Grounded Semantics and Ambiguity Propagation

Dung [11, 12] proposed an abstract argumentation framework giving rise to several argumentation semantics, in particular to a skeptical semantics (called grounded semantics) which has been widely used to characterize several defeasible reasoning systems [12, 7, 34].

In this section we show how to modify Dung's definition of acceptable argument in order to suit defeasible logic. We begin by providing precise definition for the parameters left open in the previous section (acceptable argument w.r.t. S; argument rejected by S and T).

Definition 12 An argument A for p is *acceptable* w.r.t a set of arguments S if A is finite, and

1. A is strict, or
2. every argument attacking A is attacked by S.

The idea behind this definition is to provide a notion of "validity" of arguments w.r.t. a set of arguments that have already been assessed as valid. First of all, an argument to be valid must be finite (to avoid well-known fallacies such as circular argument and infinite regress). Secondly, as we have seen in the previous section, strict arguments are just monotonic proofs; thus they are *per se* valid. Finally, we consider to be valid those arguments whose counterarguments have been undermined by arguments that have already been assessed as valid.

Definition 13 An argument A is *rejected* by sets of arguments S and T when A is not strict, and either

1. a proper subargument of A is in S, or

2. it is attacked by a finite argument.

Note that T is not used in this definition.

An argument can be rejected for two reasons: (1) part of the argument has already been rejected and (2) there is a competing argument. The intuition behind (2) is that there is no superiority relation so, given two competing arguments, there is no way to decide between the two; thus, due to the sceptical nature of this semantics, we reject the two arguments.

EXAMPLE 4 (continued)
The argument A is acceptable w.r.t. $S = \{\Rightarrow d \Rightarrow \neg b\}$ because S attacks B, the only argument attacking A.

The argument $\Rightarrow d \Rightarrow \neg b$ is rejected by any sets S and T because it is attacked by the argument B.

Using the notions of acceptable and rejected argument in Definitions 7 and 8 enables us to provide a characterization of defeasible provability in ambiguity propagating defeasible logic.

Theorem 14 Let D be a defeasible theory, p be a literal, and T be a set of arguments.

1. $D \vdash +\partial_{ap} p$ iff p is justified.
2. $D \vdash -\partial_{ap} p$ iff p is rejected by T.

In a situation where there are no strict arguments, and only finite arguments, Definition 13 reduces to Dung's definition of acceptability [12]. When combined with Definition 7, it becomes apparent that $JArgs^D$ is Dung's grounded semantics under these circumstances. For this reason, we refer to this semantics of our argumentation systems as grounded semantics.

The following examples demonstrate the concepts defined in this and the previous section.

EXAMPLE 1 (continued)
We calculate the following:

$$J_0^D = \emptyset;$$
$$J_1^D = \emptyset = JArgs^D.$$

$$R_0^D(T) = \emptyset;$$
$$R_1^D(T) = \{\Rightarrow a,\ \Rightarrow \neg a,\ \Rightarrow b,\ \Rightarrow a \Rightarrow \neg b\};$$
$$R_2^D(T) = R_1^D(T) = RArgs^D(T).$$

All arguments in $R_1^D(T)$ are supportive arguments and each is attacked by at least another one. As a result, there are no justified literals and four rejected literals. This outcome agrees with the ambiguity propagating defeasible logic where $-\partial_{ap} a$, $-\partial_{ap} \neg a$, $-\partial_{ap} b$, $-\partial_{ap} \neg b$ can be derived.

EXAMPLE 4 (continued)
We have:

$$J_0^D = \emptyset;$$
$$J_1^D = \{\rightarrow e,\ \rightarrow e \rightarrow a,\ \Rightarrow f,\ \Rightarrow d\};$$
$$J_2^D = J_1^D = JArgs^D.$$

Thus a, e, d, f are the justified literals. This corresponds to the derivability results $D \vdash \partial_{ap} a$, $D \vdash \partial_{ap} e$, $D \vdash \partial_{ap} d$, $D \vdash \partial_{ap} f$ which follow easily using the proof theory of section 2.

$R_0^D(T) = \emptyset$;
$R_1^D(T) = \{A, B, \Rightarrow d \Rightarrow \neg b\}$;
$R_2^D(T) = R_1^D(T) = RArgs^D(T)$.

The arguments for b and $\neg b$ attack each other; since these are both finite arguments, both are rejected. The literals $b, \neg b$ and c are rejected because the only arguments for them are rejected. The literals $\neg a, \neg c, \neg d, \neg e, \neg f$ are rejected, since there is no argument for them. Again this outcome corresponds to the non-derivability results $D \vdash -\partial_{ap} \neg a$, $D \vdash -\partial_{ap} b$, etc.

3.6 Defeasible Semantics and Ambiguity Blocking

In the previous section we gave an argumentation theoretic characterization of defeasible logic with ambiguity propagation. In this section we see how to modify the notions of acceptable and rejected argument in order to capture defeasible provability in defeasible logic with ambiguity blocking (our original defeasible logic).

Definition 15 An argument A for p is *acceptable* w.r.t. a set of arguments S if A is finite, and

1. A is strict, or
2. every argument attacking A is undercut by S.

Here a defeasible argument is assessed as valid if we can show that the premises of all arguments attacking it cannot be proved if we consider valid the arguments in S.

Definition 16 An argument A is *rejected* by sets of arguments S and T when A is not strict and

1. a proper subargument of A is in S, or
2. it is attacked by an argument supported by T.

The simple existence of a competing argument is not enough to state that an argument is rejected. The attacking argument must be supported by the set of justified arguments.

Now we are ready to provide a characterization of defeasible logic.

Theorem 17 Let D be a defeasible theory and p be a literal.

1. $D \vdash +\partial p$ iff p is justified.
2. $D \vdash -\partial p$ iff p is rejected by $JArgs^D$.

We refer to the semantics of argumentation systems defined in this subsection as defeasible semantics because of the above characterization of the original defeasible logic.

EXAMPLE 1 (continued)
We calculate the following:

$J_0^D = \emptyset$;
$J_1^D = \{\Rightarrow b\}$;
$J_2^D = J_1^D = \{\Rightarrow b\} = JArgs^D$.

$R_0^D(JArgs^D) = \emptyset$;
$R_1^D(JArgs^D) = \{\Rightarrow a,\ \Rightarrow \neg a,\ \Rightarrow a \Rightarrow \neg b\}$;
$R_2^D(JArgs^D) = R_1^D(JArgs^D(\emptyset) = RArgs^D(JArgs^D)$.

Note that $J_0^D = \emptyset$ undercuts the argument $\Rightarrow a \Rightarrow \neg b$ because $\emptyset$ trivially supports the argument $\Rightarrow \neg a$ which attacks $\Rightarrow a \Rightarrow \neg b$.

As a result of our calculations b is justified while $a, \neg a, \neg b$ are rejected. This outcome is consistent with the way ambiguity blocking defeasible logic works: there is evidence for b and the evidence against b cannot be used because its antecedent a is not defeasible provable. Thus the ambiguity of a is not propagated to b, instead it is used directly to allow the derivation of b.

3.7 Grounded Semantics versus Defeasible Semantics

It is worthwhile elucidating the differences between defeasible semantics and grounded semantics as defined in the previous subsections. In both cases the set of justified arguments is defined by Definition 7, but with different notions of acceptable. Under the grounded semantics, any argument attacking an acceptable argument A must be countered by an attack from S. Under the defeasible semantics the kind of counter required is different: the counter-argument must attack a subargument, not the conclusion, and the counter-argument need only be supported by S, not be a member of S as in the grounded semantics.

There are similar differences in the definitions of rejected arguments. Under the grounded semantics, an argument is rejected if it is attacked by any finite argument. Under the defeasible semantics, an argument is rejected if it is attacked by a (possibly infinite) argument supported by T. In the important case when T is $JArgs^D$, the class of arguments rejected under the defeasible semantics is smaller than under the grounded semantics, as we will see.

Despite these definitional differences, the two semantics share many common properties. We have already seen some of these properties in Section 3.4. Here we will first present some deeper common properties, before addressing the differences between the semantics.

The following common property of the two semantics represents a consistency condition: no argument is both "believed" and "disbelieved".

Theorem 18 For every defeasible theory:

- No argument is both justified and rejected.
- No literal is both justified and rejected.

The following lemma is a consequence of Theorem 18.

Lemma 19 If $JArgs^D$ contains two arguments with conflicting conclusions then both arguments are strict.

This means that inconsistent conclusions can be reached only when the strict part of the theory is inconsistent. According to Definition 7 an argument is justified if it is acceptable and Definitions 12 and 15 stipulate that strict arguments are always accepted. Hence as a corollary to Theorem 18 we can show that the set of justified arguments is harmonious in the following sense.

Corollary 20 No justified argument is attacked by a justified argument.

These properties demonstrate the proper behavior of the proposed semantics. They show that the formal concepts behave in accord with our intuitions in some important respects.

In the case of grounded semantics, we can establish a further property, permitting a simplification of the semantics and a simpler notion of justified argument.

Theorem 21 Let D be a defeasible theory. Under the grounded semantics:

1. $JArgs^D = J_1^D$.
2. An argument is justified iff no argument attacks it.

The meaning of this theorem is that, under the grounded semantics, we do not have to construct the set of accepted arguments recursively.

Let us consider a literal p to be *ambiguous* in D if there is a finite argument for each of p and $\sim p$. As a consequence of this theorem, no ambiguous literal can be justified under the grounded semantics. Indeed, as a consequence of Theorem 23, every ambiguous literal is rejected under the grounded semantics.

Unfortunately this simplification (or a similar one) is not possible for the defeasible semantics. In fact, the next example shows that Theorem 21 does not hold for that semantics.

Example 22 The following theory shows why the set $JArgs^D$ has to be built recursively under the defeasible semantics. There are the following rules, for $i = 1, \ldots, n$

$$\begin{array}{lcl} & \Rightarrow & \neg a_i \\ \neg a_i & \Rightarrow & \neg b_i \\ b_i & \Rightarrow & a_{i+1} \\ & \Rightarrow & b_i \end{array}$$

and the rule $\Rightarrow a_1$.

In this theory we have the following conclusions $-\partial a_i$, $-\partial \neg a_i$, $+\partial b_i$, $-\partial \neg b_i$, for $i = 1, \ldots, n$.

For each $i > 0$, consider the arguments

$$\begin{array}{lcccc} A_i : & & & \Rightarrow & b_i \\ B_i : & \Rightarrow & \neg a_i & \Rightarrow & \neg b_i \\ C_i : & \Rightarrow & b_{i-1} & \Rightarrow & a_i \end{array}$$

Notice that

- each A_i is attacked by B_i;
- each C_i attacks a proper subargument of B_i;

- each A_i supports C_{i+1};

and, consequently,

- each B_i is undercut by $\{A_i\}$.

It is immediate to see that the argument A_1 is acceptable w.r.t. J_0^D since no argument attacks it, so A_1 is in J_1^D. At this point C_2 is supported by J_1^D, and therefore B_2 is undercut by J_1^D; hence A_2 is acceptable w.r.t. J_1^D. We can repeat this argument to show that each A_i is in J_i^D.

However, we must first establish that A_i is justified before we can establish that A_{i+1} is justified. By Definitions 7 and 15, if $A_{i+1} \in J_{i+1}^D$, then B_i is undercut by J_i^D. But the only argument that undercuts B_i is A_i. Thus $A_{i+1} \in J_{i+1}^D$ implies $A_i \in J_i^D$, for $i = 1, \ldots, n$. It follows that

$$J_0^D \subset J_1^D \subset \cdots \subset J_{n+1}^D.$$

In comparison, it is clear that all literals in the theory are ambiguous. Thus, no literal is justified under the grounded semantics.

The ambiguity propagating defeasible logic is conceptually simpler than the ambiguity blocking defeasible logic. Consequently, the differentiation between these two logics provided by Theorem 21 and Example 22 is not a complete surprise. We might expect that a similar differentiation applies when considering rejected arguments, especially since the definition of $RArgs^D(T)$ is independent of T under the grounded semantics, but that is not so. In fact, under both semantics we have simplifications of the definition of $RArgs^D(T)$ and simpler notions of rejected argument.

Theorem 23 Let D be a defeasible theory, and T be a set of arguments. Under both the grounded and defeasible semantics:

- $RArgs^D(T) = R_1^D(T)$.

Moreover, for any argument A,

1. A is rejected by T under the grounded semantics iff A is attacked by a finite argument.
2. A is rejected by T under the defeasible semantics iff A is attacked by an argument supported by T.

The meaning of this theorem is that we do not have to recursively construct $RArgs^D(T)$, the set of arguments rejected by T, if we are given T. This result contradicts speculation in [15] that, under the defeasible semantics, $RArgs^D(JArgs^D)$ would require an iterative (or recursive) definition, even when $JArgs^D$ is given. However, when $JArgs^D$ is not given, an iterative definition of $RArgs^D(JArgs^D)$ is required, as the next example shows.

EXAMPLE 22 (continued)
Clearly the arguments

$$\begin{array}{lcl} A_i: & \Rightarrow & b_i \\ D_i: & \Rightarrow & \neg a_i \end{array}$$

are supported by $JArgs^D$. Thus B_i and C_i are rejected, using the above theorem. Furthermore, C_i is supported by $JArgs^D$, and so D_i is rejected. However, notice that D_i cannot be rejected until C_i is supported, that is, until A_i is justified. Thus, under the defeasible semantics, calculation of the rejected arguments is dependent on the justified arguments, in contrast to the situation under grounded semantics.

Under the grounded semantics, A_i, B_i, C_i and D_i are rejected, since each is attacked by a finite argument. Clearly, identifying the justified arguments is unnecessary when determining the rejected arguments.

Our final result provides a comparison of the inferential power of the grounded and defeasible semantics. It shows that the defeasible semantics justifies more arguments, but rejects fewer arguments, than the grounded semantics. Thus, although both semantics are fundamentally sceptical, the defeasible semantics can be considered more credulous than the grounded semantics. Parts 3 and 4 were originally proved in [2].

Theorem 24 Fix a defeasible theory D. Let A be an argument, and p be a literal.

1. If A is justified under the grounded semantics then A is justified under the defeasible semantics.
2. If A is rejected under the defeasible semantics then A is rejected under the grounded semantics.
3. If p is justified under the grounded semantics then p is justified under the defeasible semantics.
4. If p is rejected under the defeasible semantics then p is rejected under the grounded semantics.

We conclude this section with examples demonstrating how two traditionally problematic features of argumentation are handled by the two semantics.

Example 25 (Self-defeating arguments) In this example we show how our framework deals with the so-called self-defeating arguments. Consider the defeasible theory with the following rules:

$$\begin{array}{rl} true & \Rightarrow p \\ p & \Rightarrow \neg p \end{array}$$

This defeasible theory produces the following conclusion $-\partial\neg p$. The arguments that can be built from the theory are:

$$\begin{array}{lcccc} A_1: & & & \Rightarrow & p \\ A_2: & \Rightarrow & p & \Rightarrow & \neg p \end{array}$$

Here A_2 is a self-defeating argument.

Under the ambiguity blocking, defeasible semantics, the argument A_1, although supported by J_0^D, is not acceptable w.r.t. J_0^D since there is an attacking argument, A_2, which is not undercut by J_0^D: no proper subargument of A_2 is defeated by an argument supported by J_0^D. For the same reason A_2 is not acceptable w.r.t. J_0^D. Consequently $J_1^D = J_0^D$, and therefore $JArgs^D$ is empty. Furthermore, $A_2 \in RArgs^D$. The reason why A_2 is rejected is the following: although A_1 is not justified, it is supported by $JArgs^D$, and so it can be used to stop the validity of another argument, since we have no means of deciding which one is to be preferred. On the other hand, A_1 cannot be rejected since the argument attacking it (A_2) is not supported by $JArgs^D$: as we have already seen $\Rightarrow p$ is not a justified argument.

Under the ambiguity propagating, grounded semantics, the argument A_1 is, again, not acceptable w.r.t. J_0^D since it is attacked by A_2, and A_2 is not attacked by J_0^D. Similarly, A_2 is not acceptable w.r.t. J_0^D and hence $JArgs^D$ is empty. Both A_1 and A_2 are rejected w.r.t. $R_0^D(T)$, since each is attacked by the other, and hence $RArgs^D(T) = \emptyset$. Thus the ambiguity propagating semantics differs from the ambiguity blocking semantics in that it rejects A_1 whereas the ambiguity blocking semantics does not.

Example 26 (Circular arguments) Very often circular arguments are not considered to be true arguments since they represent a very well known fallacy, and they are excluded from the set of arguments using syntactical definitions. Briefly an argument is circular if a conclusion depends on itself as a premise.

In our approach, circular arguments correspond to infinite arguments, and they are not justified. At the same time, however, they are not automatically rejected. Moreover, such an argument can be used to attack (and defeat) other arguments.

Let us first consider the defeasible theory D_1 consisting of the rules

$$\begin{array}{ll} p & \Rightarrow q \\ q & \Rightarrow p \end{array}$$

It is immediate to see that the only possible arguments here are the infinite arguments

$$\begin{array}{lllllllll} A_1 & \dots & p & \Rightarrow & q & \Rightarrow & p & \Rightarrow & q \\ A_2 & \dots & q & \Rightarrow & p & \Rightarrow & q & \Rightarrow & p \end{array}$$

They are not justified since no proper subargument is justified, and they are not rejected since no proper subargument is rejected and there is no argument attacking them. Thus both semantics agree on D_1.

The meaning of the theory at hand is that if something is p, then normally it is q, and if something is q, then normally it is p. Thus this amounts to say that normally p and q are equivalent properties.

We add to D_1 the following rules:

$$\begin{array}{ll} q & \Rightarrow r \\ & \Rightarrow \neg r \end{array}$$

obtaining the defeasible theory D_2. In this scenario, under the defeasible (respectively, grounded) semantics, the argument for r is infinite, circular, and rejected since there is a supported (respectively, finite) argument for $\neg r$. However, the argument $A_3 : \Rightarrow \neg r$ is not justified, since the argument for r attacks it and is not undercut (respectively, not attacked) by $JArgs^D$.

Finally, D_3 is obtained from D_2 by adding the rule $true \Rightarrow \neg p$. Now, under the defeasible semantics, A_3 becomes justified since, trivially, the argument $A_4 : \Rightarrow \neg p$ is supported by $J_0^{D_3}$, A_4 attacks A_2, and therefore the argument for r is undercut. Indeed, the argument for r is rejected. A_4 is not justified, but nor is it rejected. Under the grounded semantics, the argument for r is rejected, since it is attacked by A_4, but A_3 and A_4 are not rejected, since there is no finite argument attacking them. However A_3 and A_4 are not justified, since there is no argument in J_0^D that attacks the infinite arguments attacking them.

4 Related Work

[23] proposes an abstract defeasible reasoning framework that is achieved by mapping elements of defeasible reasoning into the default reasoning framework of [7]. While this framework is suitable for developing new defeasible reasoning languages, it is not appropriate for characterizing defeasible logic because:

- [7] does not address Kunen's semantics of logic programs which provides a characterization of failure-to-prove in defeasible logic [29].
- The correctness of the mapping needs to be established if [23] is to be applied to an existing language like defeasible logic. In fact the representation of priorities is inappropriate for defeasible logic.

In section 3.5 we have seen that Dung's grounded semantics can be used to provide an argumentation theoretic characterization of the ambiguity propagating variant of Defeasible Logic; however we have shown (Theorem 21) that when we have a specific symmetric notion of attack between argument instead of an abstract one the semantics can be simplified and there is no need for a recursive construction.

Two more systems characterized by Dung's grounded semantics, even though developed with different design choices and motivations, are those proposed by Simari and Loui [37] and Prakken and Sartor [34, 33]. Both are similar to the ambiguity propagating variant of defeasible logic, but their superiority relations are different: the first is argument based instead of rule based, while the second does not deal with teams of rules (see [2] for an explanation of the term "team defeat", which refers to the full defeasible logic with priorities).

[21] proposes a labeling system, in some way similar in intuition to the tags used in Defeasible Logic, to determine the status of arguments. Moreover they show that their minimal semantics, which is defined by the usual recursive definition of accepted argument, corresponds to Dung's grounded semantics. Therefore minimal semantics characterises the justified conclusions of the ambiguity propagating variant of Defeasible Logic. However they do not contemplate a sceptical ambiguity blocking semantics, even though they advocate the need for it.

The abstract argumentation framework of [40] addresses both strict and defeasible rules, but not defeaters. However, the treatment of strict rules in defeasible arguments is different from that of defeasible logic, and there is no concept of team defeat. There are structural similarities between the definitions of inductive warrant and warrant in [40] and J_i^D and $JArgs^D$, but they differ in that acceptability is monotonic in S whereas the corresponding definitions in [40] are antitone. The semantics that results is not sceptical, and more related to stable semantics than Kunen semantics. The framework does have a notion of *ultimately defeated argument* similar to our rejected arguments.

Among other contributions, [10] provides a sceptical argumentation theoretic semantics and shows that LPwNF – which is weaker, but very similar to defeasible logic [5] – is sound with respect to this semantics. However, both LPwNF and defeasible logic are not complete with respect to this semantics.

Governatori and Maher [15] have developed an argumentation theoretic semantics for ambiguity blocking defeasible logic with superiority relation. It is easy to see that the defeasible semantics presented here is a special case of that of [15] when the superiority relation is empty. However, as we have already alluded to, the superiority relation does not add to the formal expressive power of the variants of defeasible logic presented in this paper. Moreover the presence of the superiority relation makes the definitions of the notions involved much more complicated since they are strictly entangled together. Therefore we believe that the present semantics is simpler and more elegant than that of [15] enabling thus a better understanding of the basic mechanisms of defeasible reasoning.

Other semantic frameworks have been used recently to characterize Defeasible Logic. In [29, 2] we used a meta-programming approach to characterize Defeasible Logic. A denotational semantics for Defeasible Logic was presented in [25], and a model-theoretic semantics in [27].

5 Conclusion

Defeasible logic is a non-monotonic formalism able to capture many different facets of non-monotonic reasoning; moreover, it has been applied to several fields. As is usual with non-monotonic formalism many semantics have been devised for defeasible logic, but, despite their mathematical interest, they lack the intuitive appeal that argumentation semantics offers for non-monotonic systems.

To obviate this problem we have developed an argumentation framework for defeasible logic, and we have identified conditions corresponding to two important variants of defeasible logic: one enforcing ambiguity blocking and the other ambiguity propagation. We have shown that the ambiguity propagating variant is characterized by Dung's grounded semantics. On the other hand ambiguity blocking did not correspond to any existing argumentation semantics. In most argumentation frameworks arguments are considered as black boxes without any consideration of the internal structure of the argument. The analysis of the internal structure of arguments has enabled us to determine the relation between arguments needed for capturing ambiguity blocking. At the same time it has allowed us to give a better characterization of ambiguity propagating argumentation.

Finally, the close connection between defeasible logic and argumentation frameworks opens up the possibility of using existing efficient implementations of defeasible logic as a computational platform for argumentation.

Acknowledgements

We thank Alejandro Garcia for fruitful discussions on defeasible logic and argumentation. Much of this research was performed while the authors were employed by Griffith University. A preliminary versions of this paper [16] was presented at PRICAI 2000. This research was supported by the Australia Research Council under Large Grant No. A49803544.

References

[1] G. Antoniou, D. Billington, G. Governatori and M.J. Maher. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic* 2, 2 (2001) 255–287.

[2] G. Antoniou, D. Billington, G. Governatori and M.J. Maher. A Flexible Framework for Defeasible Logics. *Proc. American National Conference on Artificial Intelligence (AAAI-2000)*, 405–410.

[3] G. Antoniou, D. Billington and M.J. Maher. Normal Forms for Defeasible Logic. *Proc. Joint International Conference and Symposium on Logic Programming*, J. Jaffar (Ed.), 160–174. MIT Press, 1998.

[4] G. Antoniou, D. Billington and M.J. Maher. On the analysis of regulations using defeasible rules. *Proc. of the 32nd Annual Hawaii International Conference on System Sciences*. IEEE Press, 1999.

[5] G. Antoniou, M. Maher and D. Billington, Defeasible Logic versus Logic Programming without Negation as Failure, *Journal of Logic Programming*, 42, 47–57, 2000.

[6] D. Billington. Defeasible Logic is Stable. *Journal of Logic and Computation* 3 (1993): 370–400.

[7] A. Bondarenko, P.M. Dung, R. Kowalski, and F. Toni. An Abstract, Argumentation-Theoretic Framework for Default Reasoning. *Artificial Intelligence*, 93 (1997): 63–101.

[8] C.I. Chesñevar, A.G. Maguitman, R.P. Loui: Logical models of argument. *ACM Computing Surveys*, 32 (2000): 337–383.

[9] J.P. Delgrande and T. Schaub. Expressing preferences in default logic. *Artificial Intelligence* 124, 1 (2000): 41–87.

[10] Y. Dimopoulos and A. Kakas. Logic Programming without Negation as Failure. *Proc. ICLP-95*, MIT Press 1995, 369–384.

[11] P.M. Dung. An Argumentation Semantics for Logic Programming with Explicit Negation. *Proceedings of the Tenth Logic Programming Conference*. MIT Press, Cambridge: 616–630.

[12] P.M. Dung. On the Acceptability of Arguments and Its Fundamental Role in Non-monotonic Reasoning, Logic Programming, and n-person games. *Artificial Intelligence*, 77 (1995): 321–357.

[13] G. Governatori, A.H.M. ter Hofstede and P. Oaks. Defeasible Logic for Automated Negotiation. P.M. Swatman and P. Swatman (Eds.). *Proceedings of ColleCTeR*, Deakin University, 2000.

[14] G. Governatori, M. Dumas, A.H.M. ter Hofstede and P. Oaks. A Formal Approach To Protocols and Strategies for (Legal) Negotiation. *Proceedings of the 18th International Conference on Artificial Intelligence and Law*, 168–177, ACM Press, New York, 2001.

[15] G. Governatori and M.J. Maher. An Argumentation-Theoretic Characterization of Defeasible Logic. W. Horn (Ed.) *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, 469–474, IOS Press, Amsterdam, 2000.

[16] G. Governatori, M.J. Maher, G. Antoniou and D. Billington. Argumentation Semantics for Defeasible Logics. *Proc. Pacific Rim Conf. on Artificial Intelligence*, 2000.

[17] G. Governatori, V. Padmanabhan and A. Sattar. A Defeasible Logic of Policy-based Intention. *Proc. Australasian Workshop on Computational Logic 2002*, 9-20, 2002.

[18] B.N. Grosof. Prioritized Conflict Handling for Logic Programs. *Proc. Int. Logic Programming Symposium*, J. Maluszynski (Ed.), 197–211. MIT Press, 1997. 9-20

[19] B.N. Grosof, Y. Labrou, and H.Y. Chan. A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML, *Proceedings of the 1st ACM Conference on Electronic Commerce (EC-99),* ACM Press, 1999.

[20] J.F. Horty. Some Direct Theories of Nonmonotonic Inheritance. In D.M. Gabbay, C.J. Hogger and J.A. Robinson (Eds.): *Handbook of Logic in Artificial Intelligence and Logic Programming Vol. 3*, 111–187, Oxford University Press, 1994,

[21] H. Jakobovits and D. Vermeir. Robust Semantics for Argumentation Frameworks. *Journal of Logic and Computation*, Vol. 9, No. 2, 215–261, 1999.

[22] A. Kakas and F. Toni. Computing Arguments in Logic Programming. *Journal of Logic and Computation*, 9: 515-562 (1999).

[23] R. Kowalski and F. Toni. Abstract Argumentation. *Artificial Intelligence and Law* 4 (1996): 275–296.

[24] K. Kunen. Negation in Logic Programming. *Journal of Logic Programming* 4 (1987): 289–308.

[25] M. Maher, A Denotational Semantics for Defeasible Logic, *Proc. First International Conference on Computational Logic*, LNAI 1861, Springer, 2000, 209–222.

[26] M. Maher, Propositional Defeasible Logic Has Linear Complexity, *Theory and Practice of Logic Programming*, 1(6), 691–711, 2001.

[27] M. Maher, A Model-Theoretic Semantics for Defeasible Logic *Proc. Workshop on Paraconsistent Computational Logic*, 67– 80, 2002.

[28] M. Maher, G. Antoniou and D. Billington. A Study of Provability in Defeasible Logic. *Proc. Australian Joint Conference on Artificial Intelligence*, 215–226, LNAI 1502, Springer, 1998.

[29] M. Maher and G. Governatori. A Semantic Decomposition of Defeasible Logic. *Proc. American National Conference on Artificial Intelligence (AAAI-99)*, 299–305.

[30] M.J. Maher, A. Rock, G. Antoniou, D. Billington and T. Miller. Efficient Defeasible Reasoning Systems. *International Journal on Artificial Intelligence Tools*, 10(4), 483–501, 2001.

[31] D. Nute. Defeasible Reasoning. *Proc. 20th Hawaii International Conference on Systems Science*, IEEE Press 1987, 470–477.

[32] D. Nute. Defeasible Logic. In D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.): *Handbook of Logic in Artificial Intelligence and Logic Programming Vol. 3*, Oxford University Press 1994, 353–395.

[33] H. Prakken. *Logical Tools for Modelling Legal Argument: A Study of Defeasible Reasoning in Law.* Kluwer Academic Publishers, 1997.

[34] H. Prakken and G. Sartor. Argument-based Extended Logic Programming with Defeasible Priorities. *Journal of Applied and Non-Classical Logics* 7 (1997): 25–75.

[35] H. Prakken and G. Vreeswijk. Logics for Defeasible Argumentation. In D.M. Gabbay and F. Guenthner (eds.): *Handbook of Philosophical Logic, 2nd edition, Vol. 4*, Kluwer, 2001, 219–318.

[36] D.M. Reeves, B.N. Grosof, M.P. Wellman, and H.Y. Chan. Towards a Declarative Language for Negotiating Executable Contracts, *Proceedings of the AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce (AIEC-99)*, AAAI Press / MIT Press, 1999.

[37] G.R. Simari and R.P. Loui. A Mathematical Treatment of Argumentation and Its Implementation. *Artificial Intelligence*, 53 (1992): 125–157.

[38] L.A. Stein. Resolving Ambiguity in Nonmonotonic Inheritance Hierarchies. *Artificial Intelligence* 55(1992): 259–310.

[39] D.D. Touretzky, J.F. Horty and R.H. Thomason. A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems. *Proc. IJCAI-87*, 476–482, Morgan Kaufmann, 1987.

[40] G. Vreeswijk. Abstract Argumentation Systems. *Artificial Intelligence*, 90 (1997): 225–279.

A Proofs

Monotonic Proofs

PROPOSITION 5
Let D be a defeasible theory and p be a literal.

1. $D \vdash +\Delta p$ iff there is a strict supportive argument for p in $Args_D$.
2. $D \vdash -\Delta p$ iff there is no (finite or infinite) strict argument for p in $Args_D$.

Proof We prove the only if part by induction on the length of defeasible proofs, the if direction of Case 1 by induction on the height of finite arguments, and the if direction of Case 2 through the contrapositive.

Case 1 ($\Rightarrow$).

Inductive base. The proof has a single line and $P(1) = +\Delta p$. This means that there is a rule r for p with empty antecedent. Thus p itself is a strict proof tree for itself.

Inductive step. Let us suppose the proposition holds for proofs of length up to n, and $P(n+1) = +\Delta p$. This means that there is a strict rule r for p such that $\forall a \in A(r)$, $+\Delta p \in P(1..n)$. By inductive hypothesis we have strict arguments (i.e., strict proof trees) for each $a \in A(r)$. Let τ be the proof tree with root p and with children the proof trees for the $a \in A(r)$. It is immediate to verify that τ is a strict proof tree/argument for p.

($\Leftarrow$).

Inductive base. Let A be a strict argument for p of height 1, that is, A is a strict rule for p with empty antecedent; therefore it is immediate to see that $+\Delta p$.

Inductive step. Let us suppose the proposition holds for arguments of height less than n, and let A be a strict argument for p of height n. The root of A is a strict rule $A(r) \rightarrow p$. By construction, for each $q \in A(r)$ we a strict argument of height less than n, thus all such literals are justified, and, by inductive hypothesis, we have $+\Delta q$; therefore the conditions to derive $+\Delta p$ are satisfied.

Case 2 ($\Rightarrow$).

Inductive base. The proof consists of a single line and $P(1) = -\Delta p$. This is possible only if there are no strict rules for p. But if there are no strict rules for p a proof tree for p cannot exist.

Inductive step. Let us suppose the proposition holds for proofs of length up to n, and $P(n+1) = -\Delta p$. This means that for each strict rule r for p, there exists a literal a in $A(r)$ such that $-\Delta a \in P(1..n)$. By inductive hypothesis there is no strict tree (argument) for a, and then r cannot be used to construct a strict proof tree for p. However, this is true for each strict rule for p, therefore a strict proof tree (argument) for p cannot be built.

($\Leftarrow$).

Suppose $D \not\vdash -\Delta p$. We will construct the required argument. If $\forall r \in R_s[p]\ \exists a \in A(r)\ D \vdash -\Delta a$ then $D \vdash -\Delta p$, since we can concatenate all such proofs and then apply the $-\Delta$ inference rule. This contradict the original supposition. Hence $\exists r \in R_s[p]\ \forall a \in A(r)\ D \not\vdash -\Delta a$. We can construct a partial argument for p that begins with r and where every leaf a satisfies $D \not\vdash -\Delta a$. We can apply the same argument to each leaf to construct a deeper partial argument. Repeated applications will construct either a finite argument or, when carried on indefinitely, an infinite argument. This completes the proof.

PROPOSITION 6
Let D be a defeasible theory and p a literal.

1. $D \vdash +\Sigma p$ iff there is a supportive argument for p in $Args_D$.
2. $D \vdash -\Sigma p$ iff there is no (finite or infinite) argument ending with a supportive rule for p in $Args_D$.

Proof The proof of this proposition is analogous to the proof of Proposition 5; all we have to do is to replace the occurrences of "strict" with "strict or defeasible".

Common Properties

Although the notion of justified is different in the two semantics we consider, the proof of the following lemma is essentially the same for both semantics.

LEMMA 10
Every subargument of a justified argument is justified.

Proof An argument A is justified iff for some n, A is acceptable w.r.t. J_n^D. Consequently, it suffices to show that, for an arbitrary set of arguments S, if A is acceptable w.r.t. S then every subargument of A is acceptable w.r.t. S. Under the grounded semantics, A is acceptable w.r.t. S iff A is finite and strict, or A is finite and every argument attacking A is attacked by S.

Let A be acceptable w.r.t. S and let B be a subargument of A. Then B is finite. If A is strict then B must also be strict, and hence B is acceptable w.r.t. S. Every argument C attacking B must also attack A. Since A is acceptable w.r.t. S, C is attacked by S. Thus every argument attacking B is attacked by S, and hence B is acceptable w.r.t. S.

Under the defeasible semantics the proof is essentially the same, except that "attacked by" is replaced by "undercut by", reflecting the difference between the two definitions of acceptable.

Many of the common properties have similar proofs for the grounded and defeasible semantics, as in the previous lemma. In the next proof we exploit Theorem 21 to give a particularly simple proof in the grounded semantics case.

THEOREM 18
For every defeasible theory:

1. No argument is both justified and rejected.
2. No literal is both justified and rejected.

Proof Suppose there is an argument that is both justified and rejected. Let n be the smallest index such that, for some argument A, $A \in RArgs^D(JArgs^D)$ and $A \in J_n^D$.

For the grounded semantics: From the definitions, there is a finite argument B that attacks A, and B is attacked by J_{n-1}^D. By Theorem 21, $n = 1$, and thus B is attacked by $J_0^D = \emptyset$. But no argument can be attacked by $\emptyset$, and hence the original supposition is false.

For the defeasible semantics: From the definitions, there is an argument B, supported by $JArgs^D$, that attacks A, and B is undercut by J_{n-1}^D. Thus there is an argument C, supported by J_{n-1}^D, that attacks a proper subargument B' of B. Since $B' \in JArgs^D$, C is undercut by $JArgs^D$, that is, there is an argument E, supported by $JArgs^D$, that attacks a proper subargument C' of C. $C' \in J_{n-1}^D$ since C' is a proper subargument of an argument supported by J_{n-1}^D. Moreover, C' is rejected, since it is attacked by an argument (E) that is supported by $JArgs^D$. But this contradicts the assumed minimality of n. Hence the original supposition is false, and no argument is both justified and rejected.

The second part follws easily from the first: if p is justified there is a supportive argument for p in $JArgs^D$. From the first part, this argument is in $Args^D - RArgs^D(JArgs)$. Thus if p is justified then it is not rejected.

The following lemma is a consequence of Theorem 18.

LEMMA 19
If $JArgs^D$ contains two arguments with conflicting conclusions then both arguments are strict.

Proof Let the two arguments be A and B. Suppose B is strict. Then, for A to be acceptable w.r.t. any S, A must be strict (since B attacks A, and B cannot be attacked or undercut, because it is strict). Thus, by symmetry, either A and B are strict, or both are non-strict.

For the grounded semantics: Suppose both A and B are non-strict. Both must be finite arguments, from the definition of acceptability. A must be rejected because it is attacked by a finite argument (B), and is justified by assumption. By Theorem 18, this is not possible. Thus no two non-strict justified arguments have conflicting conclusions.

For the defeasible semantics: The same argument applies, replacing "attacked by a finite argument" by "attacked by an argument supported by $JArgs^D$".

The following corollary follows immediately from the previous lemma.

COROLLARY 20
No justified argument is attacked by a justified argument.

Proof Suppose one justified argument attacks another at p, say. Let A and B be the corresponding subarguments with roots p and $\sim p$. By Lemma 19, A and B are strict. Thus the original argument does not attack the other at p. From this contradiction, the original supposition is false.

LEMMA 11
Let A be an argument.

1. A is acceptable w.r.t. $JArgs^D$ iff $A \in JArgs^D$

2. A is rejected by $RArgs^D(T)$ and T iff $A \in RArgs^D(T)$.

Proof If $A \in JArgs^D$ then $A \in J^D_{n+1}$, for some n. Hence, A is acceptable w.r.t. J^D_n. Since the notion of acceptability is monotonic, A is acceptable w.r.t. $JArgs^D$. Conversely, since $JArgs^D = J^D_m$, for some m, if A is acceptable w.r.t. $JArgs^D$ then $A \in J^D_{m+1}$. Thus $A \in JArgs^D$.

If $A \in RArgs^D(T)$ then $A \in R^D_{n+1}(T)$, for some n. Hence, A is rejected by $R^D_n(T)$ and T. Since the notion of rejection by S and T is monotonic in T, A is rejected by $RArgs^D(T)$ and T. Conversely, since $RArgs^D(T) = R^D_m(T)$, for some m, if A is rejected by $RArgs^D(T)$ and T then $A \in R^D_{m+1}(T)$. Thus $A \in RArgs^D(T)$.

THEOREM 23
Let D be a defeasible theory, and T be a set of arguments. Under both the grounded and defeasible semantics:

- $RArgs^D(T) = R^D_1(T)$.

Moreover, for any argument A,

1. A is rejected w.r.t. T under the grounded semantics iff A is attacked by a finite argument.

2. A is rejected w.r.t. T under the defeasible semantics iff A is attacked by an argument supported by T.

Proof By definition, $RArgs^D(T)$ is the set of arguments rejected w.r.t. T. Using the definition of R^D_i and Definition 13 (respectively Definition 16), $R^D_1(T)$ is the set of arguments attacked by a finite argument (respectively, attacked by an argument supported by T). Thus, the equation $RArgs^D(T) = R^D_1(T)$ follows immediately from the numbered statements.

1. If A is rejected then $A \in R^D_{i+1}(T)$, for some i. That is, A is rejected by $R^D_i(T)$ and T. Hence, either a proper subargument A_1 of A is in $R^D_{i-1}(T)$ or A is attacked by a finite argument. If the latter is true then the result is established. Otherwise, consider $A_1 \in R^D_{i-1}(T)$. Applying this argument i times, we find that there is a subargument A_i of A for which either a subargument is in $R^D_0(T)$ or A_i is attacked by a finite argument. Since $R^D_0(T) = \emptyset$, the former is not possible, and hence A_i (and A) is attacked by a finite argument.

   Conversely, if A is attacked by a finite argument then, from the second condition of rejection by S and T, A is rejected.

2. The above proof applies, replacing "a finite argument" by "an argument supported by T".

Ambiguity Propagating

Let the notions of accepted argument and rejected argument be those given in Definition 12 and Definition 13. Thus we are employing the grounded semantics.

LEMMA 9 (Ambiguity Propagating)
The sequences of sets of arguments J^D_i and $R^D_i(T)$ are monotonically increasing.

Proof We prove the lemma by induction on i. The inductive base is trivial since $J^D_0 = \emptyset$ and $R^D_0(T) = \emptyset$, and therefore $J^D_0 \subseteq J^D_1$ and $R^D_0(T) \subseteq R^D_1(T)$.

Let us assume it holds up to J^D_n and $R^D_n(T)$.

By definition, strict arguments are acceptable w.r.t. every set of arguments; thus they are in every J^D_i.

Let A be an argument in J^D_n. By Definition 12 if there is an argument B attacking A, then B is attacked by J^D_{n-1}; by inductive hypothesis $J^D_{n-1} \subseteq J^D_n$, thus A is acceptable w.r.t. J^D_n, and therefore $A \in J^D_{n+1}$.

Let A be an argument in $R^D_n(T)$. By Definition 13 either there is a proper subargument B of A in $R^D_{n-1}(T)$ or A is attacked by a finite argument. By inductive hypothesis $R^D_{n-1}(T) \subseteq R^D_n(T)$ and thus, in the first case, B also is in $R^D_n(T)$, and therefore A is in $R^D_{n+1}(T)$. If A is attacked by a finite argument then A is in every $R^D_i(T)$.

Corollary 27

1. If $J^D_n = J^D_{n+1}$, then $\forall m, n < m\ J^D_n = J^D_m = JArgs^D$.
2. If $R^D_n(T) = R^D_{n+1}(T)$, then $\forall m, n < m\ R^D_n(T) = R^D_m(T) = RArgs^D(T)$

Proof The two properties are immediate consequences of the proof of Lemma 9, and the definitions of $JArgs^D$ and $RArgs^D$.

THEOREM 21
Let D be a defeasible theory. Under the grounded semantics:

1. $JArgs^D = J^D_1$.
2. an argument is justified iff no argument attacks it.

Proof By definition $JArgs^D = \bigcup_{i=1}^{\infty} J_i^D$. By Corollary 27, all we have to show is that $J_1^D = J_2^D$.

Let us suppose it is not true; therefore there is an argument A such that $A \in J_2^D$ but $A \notin J_1^D$. If no argument attacks A then $A \in J_1^D$. Otherwise, let B an argument that attacks A; since A is in J_2^D there must be an argument C in J_1^D that attacks B. If C attacks B, then B attacks C. Thus, since $C \in J_1^D$, there must be an argument E in J_0^D attacking B. However $J_0^D = \emptyset$, thus no argument in J_0^D attacks B, therefore C cannot be in J_1^D, hence $A \notin J_2^D$, and we have a contradiction.

THEOREM 14
Let D be a defeasible theory and p be a literal. Under the grounded semantics:

1. $D \vdash +\partial_{ap} p$ iff p is justified.
2. $D \vdash -\partial_{ap} p$ iff p is rejected.

Proof ($\Rightarrow$) We prove this direction by induction on the length of derivations in Defeasible Logic.

Inductive Base. Let P be a proof in Defeasible Logic of length 1. Thus P consists of a single line $\pm\partial_{ap} p$.

Case $P(1) = +\partial_{ap} p$. This means there is either a 1) strict rule r for p or 2) a defeasible rule r for p with empty antecedent and there is no rule for $\sim p$. In both cases r alone is a proof tree: for 1) the argument is strict and therefore is accepted, and for 2) there is no argument for $\sim p$. Therefore the argument p is acceptable w.r.t. J_0^D, and then, by the monotonicity of the sets of justified argument (Lemma 9), p is justified.

Case $P(1) = -\partial_{ap} p$. This means that there are no strict or defeasible rules for p. Therefore there is no argument for p, and thus there are no arguments for p in $JArgs^D - RArgs^D$.

Inductive step. We assume that the property holds for proofs of length up to n and $P(n+1) = \pm\partial_{ap} p$.

Case $P(n+1) = +\partial_{ap} p$. By definition this implies that there exists a rule r such that 1) $r \in R_{sd}[p]$, 2) $\forall q \in A(r), +\partial_{ap} q \in P(1..n)$, and 3) $\forall s \in R[\sim p] \exists a_s \in A(s)$ such that $-\Sigma a_s \in P(1..n)$.

From 2) and the inductive hypothesis we have that each q is justified, thus for each q we have a justified argument A_q; from this and 1) we have a supportive argument A for p. Let m be the smallest index such that J_m^D contains such A_qs and their subarguments.[3] Let us now consider the eventual arguments that might attack A. Let B be an argument attacking A at p. Suppose first that B is an argument for $\sim p$. So B is a proof tree whose root is labeled with a rule for $\sim p$. From 3) we know that for each rule for $\sim p$ there is a literal q in the antecedent of the rule such that $-\Sigma q$ is provable; by proposition 6 there is no supportive argument for q, and therefore there is no argument for $\sim p$. Thus B is not an argument. On the other hand if B attacks a proper subargument of A, let us say A_s, then B is attacked by an argument in J_m^D (namely A_s). Since all arguments attacking A are attacked by J_m^D, $A \in J_{m+1}^D$.

Case $P(n+1) = -\partial_{ap} p$. We have to consider two cases: 1) $\forall r \in R_{sd}[p] \exists q \in A(r)$ such that $-\partial_{ap} q \in P(1..n)$ and 2) $\exists s \in R[\sim p] \forall q \in A(s), +\Sigma q \in P(1..n)$. Let us consider an argument A whose ending rule is a supportive rule for p.

[3]As shown in Lemma 10, a subargument of a justified argument is itself justified.

For 1) we have that, by inductive hypothesis for any supportive rule for p there is a literal for which either all the arguments are rejected or there is no argument; in the first case a proper subargument of A is rejected and thus A is rejected too, and in the second case A is not an argument. Thus either all the supportive arguments for p are rejected or there are no such arguments.

For 2) by Proposition 6 we have a finite argument for $\sim p$, and therefore any argument A for p is rejected.

Case 1 ($\Leftarrow$). Here we prove that if a supportive argument for p is in J_1^D, then p is provable defeasibly. An argument A is in J_1^D if A is acceptable w.r.t. J_0^D, that is, the empty set. If p is justified according to J_1^D, then there is an argument A accepted w.r.t. the empty set. The argument A for p must be finite, and either 1) A is strict, or 2) every argument attacking A is attacked by J_0^D. 1) If A is strict then, by Proposition 5, $D \vdash +\Delta p$ and therefore $D \vdash +\partial_{ap} p$; otherwise we consider the height n of A.

If A has height 1, then the argument consists of a single rule with empty body, so the only possible form of attack is an attack to the head. This means there is an argument for $\sim p$. However, every argument for $\sim p$ must be attacked by an argument in J_0^D; but J_0^D is the empty set, thus there is no argument for $\sim p$. But if there are no argument for $\sim p$, then, according to Proposition 6, $D \vdash -\Sigma{\sim}p$. Therefore, in this case $D \vdash +\partial_{ap} p$.

We assume that the property holds for arguments in J_1^D with height up to n. Let B an argument attacking A. Now B can attack A in two ways: they have competing conclusions, or B and a proper subargument of A have competing conclusions. We use the same reasoning in both cases: an argument in J_1^D cannot be attacked by any argument, otherwise the attacking argument must be attacked by an argument in the empty set; thus $D \vdash -\Sigma{\sim}p$. Moreover, each subargument of A, has height less than n, and we can make use of the inductive hypothesis to conclude that there is a (strict or defeasible) rule r for p, and $\forall q \in A(r)$, $D \vdash +\partial_{ap} q$. Hence $D \vdash +\partial_{ap} p$.

Case 2 ($\Leftarrow$).

If $D \not\vdash -\partial_{ap} p$ then either

(1) $D \not\vdash -\Delta p$, or

(2) $\exists r \in R_{sd}[p]\ \forall a \in A(r)\ D \not\vdash -\partial_{ap} a$ and $\forall s \in R[\sim p]\ \exists a \in A(s)\ D \not\vdash +\Sigma a$

We construct an unrejected argument for p, starting from a partial tree containing only the unexpanded node p.

If (1) then, by Proposition 5, there is a finite or infinite strict argument B for p. Since B is strict, it is not rejected. Expanding the node p with B constructs the argument (and so p is not rejected).

If (2) then, for every rule s for $\sim p$, some $a \in A(s)$ satisfies $D \not\vdash +\Sigma a$. By Proposition 6, there is no supportive argument for a. Hence, there is no finite argument for $\sim p$.

In addition, there is a supportive rule r for p such that $\forall a \in A(r)\ D \not\vdash -\partial_{ap} a$. Thus we can expand p to have the unexpanded children $a \in A(r)$. We can repeat this construction for each $a \in A(r)$, and for each of their unexpanded children, and so on. Thus we can construct a finite or infinite argument C for p that does not use defeaters.

Furthermore, for every literal q in C, there is no finite argument for $\sim q$ attacking C: either (1) applied at this node and this subargument is strict (and thus cannot be attacked), or (2) applied and we established above that there is no finite argument for $\sim q$. Thus C, and every subargument of C, is not attacked by by a finite argument. Hence, by Theorem 23, C is not rejected.

Ambiguity Blocking

Let the notions of accepted argument and rejected argument be those given in Definition 15 and Definition 16. Thus we are employing the defeasible semantics.

LEMMA 9 (Ambiguity Blocking)
The sequences of sets of arguments J_i^D and $R_i^D(JArgs^D)$ are monotonically increasing.

Proof We prove the lemma by induction on i. The inductive base is trivial in both cases since $J_0^D = \emptyset$ and $R_0^D(T) = \emptyset$ and thus $J_0^D \subseteq J_1^D$ and $R_0^D(T) \subseteq R_1^D(T)$.

By definition strict arguments are acceptable w.r.t. every set of arguments thus they are in every J_i^D.

Let A be an argument in J_n^D, and let B an argument attacking A. By construction B is undercut by J_{n-1}^D, and, by inductive hypothesis $J_{n-1}^D \subseteq J_n^D$; hence B is undercut by J_n^D. Therefore $A \in J_{n+1}^D$.

We consider now the sequence of rejected arguments. Any argument attacked by an argument supported by T is in $R_i^D(T)$, for every i. Let A be an argument in $R_n^D(T)$. If a proper subargument B of A is in $R_{n-1}^D(T)$, then, by inductive hypothesis, $B \in R_n^D(T)$; therefore $A \in R_{n+1}^D(T)$.

THEOREM 17
Let D be a defeasible theory and p be a literal. Under the defeasible semantics:

1. $D \vdash +\partial p$ iff p is justified.
2. $D \vdash -\partial p$ iff p is rejected by $JArgs^D$.

Proof ($\Rightarrow$). We prove the only if direction of the theorem by induction on the length of derivations in Defeasible Logic.

Inductive Base. Let P be a derivation in Defeasible Logic.

Case $P(1) = +\partial p$. This means there is supportive rule for p with empty antecedent. If the rule is strict the rule itself is a strict argument for p, and strict argument are acceptable w.r.t. any J_i^D. Therefore the argument is justified and so is p. If the rule for p is defeasible then the rule itself is a defeasible argument for p. Let us call this argument A. Moreover, condition 2.3 of $+\partial$ must be satisfied. This is possible only if there are no rules for $\sim p$, but if there are no rules for $\sim p$, there are no arguments for $\sim p$; the only way A can be attacked is by an argument for $\sim p$. We have seen that in such a case there are no arguments for $\sim p$, and therefore $A \in J_1^D$. By Lemma 9, $A \in JArgs$, and so p is justified

Case $P(1) = -\partial p$. This is possible only in the case where there are no supportive rules for p; then there are no supportive arguments for p in $Args^D$, so p is rejected.

Inductive Step. As usual we assume that the theorem holds for derivations whose length is less than or equal to n.

Case $P(n + 1) = +\partial p$. We consider only the cases different from those investigated in the inductive base. By definition there is a supportive rule $r \in R_{sd}[p]$ such that $\forall a_r \in A(r)$, $+\partial a_r \in P(1..n)$. By inductive hypothesis we have justified arguments for each a_r; this implies we have a supportive argument for p, let us call it A. Moreover $\forall s \in R[\sim p]$, $\exists a_s \in A(s)$ such that $-\partial a_s \in P(1..n)$. By inductive hypothesis such a_ss are rejected w.r.t.

$JArgs^D$; that is either there are no supportive arguments or such arguments are attacked by arguments supported by $JArgs^D$.[4]

Consider an argument B attacking A. If it attacks a proper subargument of A then it is undercut by $JArgs^D$, since the subarguments are justified. If B attacks A at $\sim p$, then the subargument at some child of $\sim p$ in B is rejected w.r.t. $JArgs^D$, as discussed above. Thus some proper subargument of B is attacked by an argument supported by $JArgs^D$. Hence, also in this case, B is undercut by $JArgs^D$. Consequently, every argument attacking A is undercut by $JArgs^D$. It follows that A is acceptable w.r.t. $JArgs^D$, and hence p is justified.

Case $P(n+1) = -\partial p$. By definition $\forall r \in R_{sd}[p]$ either (a) $\exists a_r \in A(r)$ such that $-\partial a_r \in P(1..n)$ or (b) $\exists s \in R[\sim p]$ such that $\forall a_s \in A(s)$ $+\partial a_s \in P(1..n)$.

For (a), by inductive hypothesis any supportive argument for p has a rejected proper subargument. Therefore all the supportive arguments for p are rejected, and hence p is rejected.

For (b), by inductive hypothesis, every argument for p is attacked by an argument supported by $JArgs^D$. Hence every argument for p is rejected, and also in this case p is rejected.

($\Leftarrow$). We prove the first part by induction on the stage of acceptability of arguments for p and by induction on the height of trees for p.

Case 1. To prove this case we have to use a double induction. The external induction on the stage of acceptability of arguments, and then induction on the height of arguments with the same stage of acceptability.

Case Inductive Base. we begin by considering supportive arguments acceptable w.r.t. J_0^D whose height is 1. Such arguments consist of a single supportive rule r for p with empty antecedents. If the argument (let us say A) is strict then r is strict and therefore we can prove $D \vdash +\Delta p$ and consequently $D \vdash +\partial p$. If the argument is defeasible then r is a defeasible rule, and in such a case we know that every attack on A is undercut by J_0^D, that is the empty set. Let B be an argument attacking A whose root is a rule s for $\sim p$. Obviously, B is an argument for $\sim p$. Since A is acceptable w.r.t. J_0^D, B is undercut by J_0^D; therefore there is an argument C attacking a subargument of B. The argument C is supported by J_0^D; thus C consists of a single rule for $\sim q$, for some q occurring in B. At this point it is immediate to verify that the conditions to prove $-\partial q$ are satisfied.

We now show that $\forall s \in R[\sim p]\ \exists a \in A(s) : D \vdash -\partial a$. Suppose, to obtain a contradiction, that $\exists s \in R[\sim p]\ \forall a \in A(s) : D \not\vdash -\partial a$. We construct a (possibly infinite) argument for $\sim p$ as follows.

Initially, the partial tree contains $\sim p$ at the root, with unexpanded children $a \in A(s)$.

Now, let a be an unexpanded node such that $D \not\vdash -\partial a$. From the inference rule $-\partial$, we know that if $D \not\vdash -\partial a$ then either

(1) $D \not\vdash -\Delta a$, or

(2) $\exists r \in R_{sd}[a]\ \forall a' \in A(r)\ D \not\vdash -\partial a'$

If (1) then, by Proposition 5, there is a finite or infinite strict argument B for a. Expanding the node a with B constructs the argument. Otherwise, (2) holds and there is a supportive rule r for a such that $\forall a' \in A(r)\ D \not\vdash -\partial a'$. Thus we can expand a to have the unexpanded children $a' \in A(r)$.

[4]If we unravel the derivation of $-\partial a_s$ we have that some rules are discarded because some antecedents are not provable (i.e., for some literals q we can prove $-\partial q$ and then we have to repeat the same reasoning). Otherwise we have a rule for $\sim a_s$ and we can attach the tag $+\partial$ to every literal in the antecedent of that rule; we can apply the inductive hypothesis for such literals, thus they are justified, and so an argument for $\sim a_s$ is supported by $JArgs^D$.

We can repeat this construction for each unexpanded node that appears in the partial tree. Thus we can construct a finite or infinite argument B for $\sim p$ where every literal a in B satisfies $D \not\vdash -\partial a$.

However, we have already noted that for any such argument B there is an argument C attacking a subargument of B at some q, and that $D \vdash -\partial q$. Thus we have a contradiction, and consequently $\forall s \in R[\sim p]\ \exists a \in A(s) : D \vdash -\partial a$. Combined with the finite argument A for p, we thus have $D \vdash +\partial p$.

To compete the proof of the inductive base we have to show that the property holds for arbitrary arguments in J_1^D. So, we assume, by induction, that the theorem is true for literals having arguments in J_1^D whose height is less than h. Let A be a supportive argument for p whose height is h. All the subarguments of A have height less than h, therefore, if r is the rule labeling the root of A, by induction we have $\forall a_r \in A(r)$, $D \vdash +\partial a_r$. At this point we can repeat the reasoning of the previous case.

Inductive step. We can repeat the proof of the inductive base noting that the undercutting arguments are supported by J_n^D, and any argument in J_n^D is justified; thus for any literal q in the antecedents of undercutting arguments we have $D \vdash +\partial q$.

Case 2. We prove the contrapositive. From the inference rule $-\partial$, we know that if $D \not\vdash -\partial p$ then either

(1) $D \not\vdash -\Delta p$, or

(2) $\exists r \in R_{sd}[p]\ \forall a \in A(r)\ D \not\vdash -\partial a$ and $\forall s \in R[\sim p]\ \exists a \in A(s)\ D \not\vdash +\partial a$

We construct an unrejected argument for p, starting from a partial tree containing only the unexpanded node p.

If (1) then, by Proposition 5, there is a finite or infinite strict argument B for p. Since B is strict, it is not rejected. Expanding the node p with B constructs the argument (and so p is not rejected).

If (2) then there is a supportive rule r for p such that $\forall a \in A(r)\ D \not\vdash -\partial a$. Thus we can expand p to have the unexpanded children $a \in A(r)$. We can repeat this construction for each $a \in A(r)$, and for each of their unexpanded children, and so on. Thus we can construct a finite or infinite argument C for p that does not use defeaters.

Consider an argument E attacking C at q. If this node satisfies (1) then it is part of a strict argument and cannot be attacked. Consequently the node satisfies (2). Hence, $\forall s \in R[\sim q]\ \exists a \in A(s)\ D \not\vdash +\partial a$. So whatever rule s is at the root of E, a child a satisfies $D \not\vdash +\partial a$. By the first part of this theorem, a is not justified. Thus E is not supported by $JArgs^D$. Consequently C is not undercut by an argument. Hence, using Theorem 23, C is not rejected.

Comparison

THEOREM 24
Fix a defeasible theory D. Let A be an argument, and p be a literal.

1. If A is justified under the grounded semantics then A is justified under the defeasible semantics.

2. If A is rejected under the defeasible semantics then A is rejected under the grounded semantics.

3. If p is justified under the grounded semantics then p is justified under the defeasible semantics.
4. If p is rejected under the defeasible semantics then p is rejected under the grounded semantics.

Proof By Theorem 21, if A is justified under the grounded semantics then it is not attacked by any argument. Consequently, $A \in J_1^D$ under the defeasible semantics.

By Theorem 23, if A is rejected by $JArgs^D$ under the defeasible semantics then A is attacked by an argument supported by $JArgs^D$. Since all justified arguments are finite, A is attacked by a finite argument. Thus, again by Theorem 23, A is rejected under the grounded semantics.

Parts 3 and 4 follow immediately from parts 1 and 2 and Definitions 7 and 8.

B Metaprogam

The meta-programs defined in this appendix are based on the family of meta-programs given in [2]. We have permitted ourselves some syntactic flexibility in presenting the meta-programs. However, there is no technical difficulty in using conventional logic programming syntax to represent these programs.

$c1$ $\texttt{definitely}(p)$:- $\texttt{fact}(p)$.

$c2$ $\texttt{definitely}(p)$:- $\texttt{strict\_rule}(r, p, [q_1, \ldots, q_n])$,
$\texttt{definitely}(q_1), \ldots, \texttt{definitely}(q_n)$.

$c3$ $\texttt{supported}(p)$:- $\texttt{fact}(p)$.

$c4$ $\texttt{supported}(p)$:- $\texttt{supportive\_rule}(r, p, [q_1, \ldots, q_n])$,
$\texttt{supported}(q_1), \ldots, \texttt{supported}(q_n)$.

$c5$ $\texttt{defeasibly}(p)$:- $\texttt{definitely}(p)$.

$c6$ $\texttt{defeasibly}(p)$:- not $\texttt{definitely}({\sim}p)$,
$\texttt{supportive\_rule}(r, p, [q_1, \ldots, q_n])$,
$\texttt{defeasibly}(q_1), \ldots, \texttt{defeasibly}(q_n)$,
not $\texttt{overruled}(r, p)$.

$c7$ $\texttt{overruled}(r, p)$:- $\texttt{rule}(s, {\sim}p, [q_1, \ldots, q_n])$,
$\texttt{defeasibly}(q_1), \ldots, \texttt{defeasibly}(q_n)$.

$c8$ $\texttt{overruled}(r, p)$:- $\texttt{rule}(s, {\sim}p, [q1, \ldots, qn])$,
$\texttt{supported}(q_1), \ldots, \texttt{supported}(q_n)$.

The interpretation of the predicates is straightforward. The first two clauses address definite provability, the second two describe the notion of support and the remainder address defeasible provability. The clauses specify if and how a rule in defeasible logic can be overridden by another, among other aspects of the structure of defeasible reasoning in defeasible logic.

Clauses $c1$–$c6$ are common to the two variants of defeasible logic. Clause $c7$ defines `overruled` in the ambiguity blocking variant of defeasible logic while clause $c8$ defines the same notion in the ambiguity propagtion version. Intuitivelly clauses $c7$ and $c8$ roughly corresponds to the notions of undercut and the attack among arguments.