Argumentation Semantics for Defeasible Logics

G. Governatori[1], M.J. Maher[2], G. Antoniou[2], and D. Billington[2]

[1] School of Information Systems, Queensland University of Technology,
GPO Box 2434 Brisbane, QLD 4001, Australia
[2] School of Computing and Information Technology, Griffith University,
Nathan, QLD 4111, Australia
{guido,mjm,ga,db}@cit.gu.edu.au

Abstract. Defeasible logic is a simple but efficient rule-based non-monotonic logic. It has powerful implementations and shows promise to be applied in the areas of legal reasoning and the modelling of business rules. So far defeasible logic has been defined only proof-theoretically. Argumentation-based semantics have become popular in the area of logic programming. In this paper we give an argumentation-based semantics for defeasible logic.
Recently it has been shown that a family of approaches can be built around defeasible logic, in which different intuitions can be followed. In this paper we present an argumentation-based semantics for an ambiguity propagating logic, too. Further defeasible logics can be characterised in a similar way.

1 Introduction

Defeasible logic is a practical nonmonotonic logic. This logic, and similar logics, have been proposed as the appropriate language for executable regulations [3], contracts [22], and business rules [13]. Unlike other nonmonotonic approaches, defeasible logic was designed to be easily implementable. In fact, recently very powerful implementations of defeasible logic became available, capable of handling 100,000s of defeasible rules [4]. Moreover, in [2] we have shown how to "tune" defeasible logic in order to deal with several nonmonotonic phenomena described in the literature.

Dung [9, 10] presented an abstract argumentation framework, and [7] shown that several well-known nonmonotonic reasoning systems are concrete instances of the abstract framework. Although defeasible logic can be described informally in terms of arguments, the logic has been formalized in a proof-theoretic setting in which arguments play no role. In this paper we will provide an argumentation-theoretic semantics for defeasible logic.

In addition to innovations we make in argumentation theory, the resulting argumentation-theoretic semantics will be advantageous for defeasible logic. The logic currently has no model theory, and the proof theory is clumsy. The semantics we provide is considerably more elegant. It will prove useful in the intended applications of defeasible logic mentioned above, where arguments are a natural feature of the problem domain.

This work is part of our ongoing effort to establish close connections between defeasible reasoning and theories of argumentation. Such connections usually lead to better

understanding, and cross-fertilisation. Also it is worth noting that usually argumentation is studied theoretically, while not so much emphasis is 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 abstract 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 logic. In this short paper there is no room for full details; for those we refer the reader to [17, 2]. We then provide our argumentation-theoretic semantics for defeasible logic and an ambiguity propagating variant in Section 3.

2   Overview of Defeasible Logics

We begin by presenting the basic ingredients of defeasible logic. A defeasible theory contains five different kinds of knowledge: facts, 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 variable-free instances.

Facts are indisputable statements, for example, "Tweety is an emu". In the logic, this might be expressed as $emu(tweety)$.

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: $emu(X) \rightarrow bird(X)$.

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: $heavy(X) \rightsquigarrow \neg flies(X)$. 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 that the animal *may* not be able to fly. 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 defeasible rules

$$r : \qquad bird \Rightarrow flies$$
$$r' : brokenWing \Rightarrow \neg flies$$

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.

It is not possible in this short paper to give a complete formal description of the logic. However, we hope to give enough information about the logic to make the discussion intelligible. We refer the reader to [19, 6, 17, 2] for more thorough treatments.

A *rule r* consists of its *antecedent* (or *body*) $A(r)$ which is a finite set of literals, an arrow, and its *consequent* (or *head*) $C(r)$ which is a literal. 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 triple $(F, R, >)$ where F is a finite set of facts, R a finite set of rules, and $>$ a superiority relation on 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 facts and strict rules).

$-\Delta q$ which is intended to mean that we have 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 we have proved that q is not defeasibly provable in D.

Provability is based on the concept of a *derivation* (or *proof*) in $D = (F, R, >)$. 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). Here we briefly state the conditions for positive defeasible conclusions [6]. The structure of the inference rules for negative literals is the same as that for the corresponding positive one, but the conditions are negated in some sense. The purpose of the $-\Delta$ and $-\partial$ inference rules is to establish that it is not possible to prove a corresponding positive tagged literal. These rules are defined in such a way that all the possibilities for proving $+\partial q$ (for example) are explored and shown to fail before $-\partial q$ can be concluded. Thus conclusions with these tags are the outcome of a constructive proof that the corresponding positive conclusion cannot be obtained.

In this paper we present the inference rules in a simplified form instead of the general one. In particular we do not consider the superiority relation. In fact, in [1], we proved that the superiority relation can be simulated in terms of the other elements of defeasible logic, and we provide an effective translation to transform a defeasible theory in an equivalent one with an empty superiority relation. The use of the simplified conditions will make our formal considerations much simpler.

In the following $P(1..i)$ denotes the initial part of the sequence P of length i.

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

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. In the logic above ambiguities are blocked. In the following we introduce an ambiguity propagating variant. The result of [1] can be easily extended to this variant; thus 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 chain of reasoning that would lead us to conclude p in the absence of conflicts. This leads to the following inference conditions:

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

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

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.

A literal is *ambiguous* if there is a chain of reasoning that supports a conclusion that p is true, and another that supports that $\neg p$ is true.

We can achieve ambiguity propagation behaviour by making a minor change to the inference condition for $+\partial$: instead or 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*. Thus we are imposing a stronger condition for proving a literal defeasibly. Here is the formal definition:

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

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

3 Argumentation for Defeasible Logic

Arumentation 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; however, we consider facts to be strict rules with empty bodies.

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.

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

- If $b_1, \ldots, b_n$ label the children of h then there is a ground instance of a rule in R with body $b_1, \ldots, b_n$ and head h.
- If, in addition, h is not the root of the tree then the rule must be a strict or defeasible rule.

If the rule at the root of a proof tree is strict or defeasible and the proof tree is finite we say it is a *supportive proof tree*. If all the rules in a proof tree are strict then we say that it is a *strict proof tree*.

An *argument* for a literal p is a proof tree for p. We say that an argument A is *finite* if the proof tree associated to A is finite. An argument A is *strict* if the proof tree associated to A is strict. If an argument is not strict it is *defeasible*. An argument A for p is a *supportive argument* if the proof tree for p associated to A is supportive.

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

Defeasible logic has three kinds of rules and only two of them can be used to support the derivation of a conclusion. Defeaters can only block derivations. Intuitively a supportive argument is an argument from which a conclusion can be drawn.

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

Proposition 1. *Let D be a defeasible theory and p be a literal.*

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

This characterization is straightforward, since strict rules are the monotonic subset of defeasible logic.

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 2. *Let D be a defeasible theory and p a literal.*

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

On the hand, characterizing defeasible provability requires more definitions.

A *(proper) subargument* of an argument A is a subtree of the proof tree associated to A.

An argument A *attacks* an argument B if a conclusion of A is the complement of a conclusion of B. A set of arguments S attacks a defeasible argument B if there is an argument A in S that attacks B.

An 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 (and thus preventing some conclusions), while supportive proof trees are finite and do not contain defeaters (cf. Proposition 2).

An argument A is *undercut* by a set of arguments S if S supports an argument B attacking a proper subargument of A.

It is worth noting that the above definitions concern only defeasible arguments; for strict arguments we stipulate that they cannot be undercut or attacked.

Example 1. We consider the defeasible theory D consisting of the following rules:

$$a \Rightarrow p \qquad b \Rightarrow \neg p \qquad p \Rightarrow q$$

Let $S = \{a, b\}$ be a set of arguments. The argument

$$A: \quad a \Rightarrow p \Rightarrow q$$

is undercut by S since the argument $B : b \Rightarrow \neg p$ attacks a subargument of A, and it is supported by S.

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.

The heart of argumentation semantics is the notion of acceptable argument. However, different definitions are possible and they characterise different variants of defeasible logic. Such a notion is used as a basis to define recursively the set of justified arguments. For the moment we leave it undefined (we shall propose later two different definitions: the first characterises the ambiguity propagating variant of defeasible logic – Definition 3 in Section 3.1–, and the second the ambiguity blocking variant – Definition 5 in Section 3.2), and we proceed to define the set of justified arguments.

Definition 1. *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 $JArg^D$.

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 conclusive argument 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 a justified 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 (the appropriate definitions will be given in Section 3.1 and Section 3.2).

Even in the case of rejected argument we need a recursive construction (see example 2 below for an explanation).

Definition 2. *We define R_i^D as follows.*

– $R_0^D = \emptyset$
– $R_{i+1}^D = \{a \in Args_D \mid a \text{ is rejected by } R_i^D\}$

The set of rejected arguments *in a defeasible theory D is $RArgs^D = \cup_{i=1}^{\infty} R_i^D$.*

A literal p is *rejected* if there is no argument in $Args_D - RArgs^D$ that ends with a supportive rule for p.

3.1 Grounded Semantics and Ambiguity Propagation

Dung [9, 10] 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 [10, 7].

In this section we show how to modify Dung's definition of acceptable argument in order to suit defeasible logic.

Definition 3. *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.*

As we have seen defeasible logic is more expressive, insofar as it is able to determine when a conclusion is demonstrably non provable; thus, before proving that grounded semantics characterises the ambiguity propagating variant of defeasible logic, we have to define the appropriate notion of rejected argument.

Definition 4. *An argument A is* rejected *by a set of arguments S when A is not strict, and either*

1. *a proper subargument of A is in S, or*
2. *it is attacked by a supportive argument.*

Using the notions of acceptable and rejected argument in definitions 1 and 2 enables us to prove the following theorem.

Theorem 1. *Let D be a defeasible theory and p be a literal.*

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

This theorem provides a characterization of defeasible provability in defeasible logic with ambiguity propagation.

3.2 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 5. *An argument A for p is* acceptable *w.r.t to a set of argument S if A is finite, and*

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

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.

Definition 6. *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.*

To accommodate with the slightly different notion of rejected argument we have to modify the second point of Definition 2 as follow

- $R_{i+1}^D = \{a \in Args_D \mid a \text{ is rejected by } R_i^D \text{ and } JArgs^D\}$

Theorem 2. *Let D be a defeasible theory and p be a literal.*

- $D \vdash +\partial p$ *iff p is justified.*
- $D \vdash -\partial p$ *iff p is rejected.*

This theorem provides a characterization of defeasible conclusions in ambiguity blocking defeasible logic in terms of justified and rejected argument in defeasible argumentation semantics.

Governatori and Maher [11] have developed an argumentation theoretic semantics for ambiguity blocking defeasible logic with superiority relation. It is easy to see that the semantics presented here is a special case of that of [11] when the superiority relation is empty. However, as we have already alluded to, the superiority relation does not add anything to the expressive power of the variants of defeasible logic presented in this paper. Therefore we believe that the present semantics enables a better understanding of the basic mechanisms of defeasible reasoning.

Example 2. The following defeasible logic theory illustrates why $RArgs^D$ needs to be constructed iteratively, even after all the justified literals have been identified.

There are the following rules, for $i = 1, \ldots, n$:

$$\begin{aligned} true &\Rightarrow b_i \\ a_i &\Rightarrow \neg b_i \\ b_{i-1} &\Rightarrow a_i \\ true &\Rightarrow \neg a_i \end{aligned}$$

and the fact b_0.

This theory produces the following conclusions: $-\partial a_i, -\partial \neg a_i, +\partial b_i, -\partial \neg b_i$, for $i = 0, \ldots, n$.

For each i, consider the following arguments:

$$\begin{aligned} A_i : &\qquad\qquad\qquad true \Rightarrow \neg a_i \\ B_i : &\ true \Rightarrow b_{i-1} \Rightarrow a_i \quad \Rightarrow \neg b_i \end{aligned}$$

and their subarguments. Notice that

- each argument A_i is attacked by B_i at a_i.
- each argument B_i is attacked by B_{i-1} at b_{i-1}.

Eventually, both A_i and B_i will be rejected, since neither can defeat the other, but this cannot be done until the status of b_{i-1} is determined. As noted above, this depends on B_{i-1}. Thus the situation incorporates some sequentiality, where B_{i-1} must be resolved before resolving B_i, and this suggests that a characterization of $RArgs^D$ must be iterative, even after all the justified literals have been identified.

4   Related Work

[16] 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 [18].

- The correctness of the mapping needs to be established if [16] is to be applied to an existing language like defeasible logic. In fact the representation of priorities is inappropriate for defeasible logic.

Two more systems characterized by Dung's grounded semantics, even if developed with different design choices and motivations, are those proposed by Simari and Loui [23] and Prakken and Sartor [21, 20]. Both are similar to the ambiguity blocking 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.

The abstract argumentation framework of [24] 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 [24] and J_i^D and $JArgs_D$, but they differ in that acceptability is monotonic in S whereas the corresponding definitions in [24] 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, but the definition is not iterative, possibly because the framework does not have a directly sceptical semantics.

Among other contributions, [8] 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.

5 Conclusion

Defeasible logic is a simple but efficient rule-based nonmonotonic logic. So far defeasible logic has been defined only proof-theoretically. In this paper we presented an argumentation-theoretic semantics for defeasible logic and an ambiguity propagating variant. This paper is part of our ongoing effort to establish close connections between defeasible reasoning and theories of argumentation.

Acknowledgments

References

1. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. Representation Results for Defeasible Logic. Technical Report, CIT, Griffith University, 2000.
2. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. A Flexible Framework for Defeasible Logic. *Proc. American National Conference on Artificial Intelligence (AAAI-2000).*

3. G. Antoniou, D. Billington and M.J. Maher. On the analysis of regulations using defeasible rules. In R.H. Sprague (Ed.) *Proc. of the 32nd Annual Hawaii International Conference on System Sciences*. IEEE Press, 1999.

4. G. Antoniou, D. Billington, M.J. Maher, A. Rock, Efficient Defeasible Reasoning Systems, *Proc. Australian Workshop on Computational Logic*, 2000.

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. Y. Dimopoulos and A. Kakas. Logic Programming without Negation as Failure. In *Proc. ICLP-95*, MIT Press 1995.

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

10. 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.

11. G. Governatori and M.J. Maher. An Argumentation-Theoretic Characterization of Defeasible Logic. In W. Horn (ed.) *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, 2000.

12. B.N. Grosof. Prioritized Conflict Handling for Logic Programs. In *Proc. Int. Logic Programming Symposium*, J. Maluszynski (Ed.), 197–211. MIT Press, 1997.

13. 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.

14. 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,

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

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

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

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

19. 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.

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

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

22. 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.

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

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