An Argumentation-Theoretic Characterization of Defeasible Logic

G. Governatori and **M.J. Maher** [1]

Abstract. Defeasible logic is an efficient non-monotonic logic that is defined only proof-theoretically. It has potential application in some legal domains. We present here an argumentation semantics for defeasible logic that will be useful in these applications. Our development differs at several points from existing argumentation frameworks since there are several features of defeasible logic that have not been addressed in the literature.

1 Introduction

Defeasible logic (DL) is a practical non-monotonic logic. This logic, and similar logics, have been proposed as the appropriate language for executable regulations [4], contracts [22], and business rules [13]. There are several implementations of DL, each of which is capable of handling 100,000's of rules [5].

Although DL 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 DL.

There are already several different abstract argumentation frameworks [10, 8, 15, 20, 23, 24]. However, DL provides several challenges that have not yet been addressed by this work:

(1) DL has a "directly sceptical" semantics, in the sense of Horty [14], also called "conservative" in Wagner's classification [25]. Most argumentation-theoretic approaches provide sceptical semantics as the common part of credulous semantics, and so do not address this sort of scepticism. (2) DL provides three different kinds of rule, including a rule that cannot support an argument, only defeat one. Most argumentation-theoretic works have addressed a single kind of rule. (3) In DL, positive conclusions (that a proposition can be proved) are not the only consideration; negative conclusions (that a proposition cannot be proved) are of equal significance. (4) DL exhibits "team defeat" [12], in which one collection of arguments may defeat another, although no single argument defeats every argument in the other collection.

Technically, the main modifications we make to conventional argumentation-theoretic frameworks are: the explicit introduction of infinite arguments, the treatment of teams of arguments, rather than considering each argument only individually, and an iterative definition of rejected arguments.

In addition to innovations we make in argument theory, the resulting argumentation-theoretic semantics will be advantageous for DL. 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 DL mentioned above, where arguments are a natural feature of the problem domain.

This paper is structured as follows. In the next section we provide a brief introduction to DL. In this short paper there is no room for full details; for those we refer the reader to [17]. We then provide our argumentation-theoretic semantics for DL in Section 3. We conclude with a discussion of related work.

2 Overview of Defeasible Logic

We begin by presenting the basic ingredients of DL. 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:

$$heavy(X) \leadsto \neg flies(X).$$

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 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

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

[1] School of Computing and Information Technology, Griffith University, Nathan, QLD 4111, Australia. {guido,mjm}@cit.gu.edu.au

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 turns out that we only need to define the superiority relation over rules with contradictory conclusions.

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, 7, 17] 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 and can have one of the following four forms:

$+\Delta q$, 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 condition for positive defeasible conclusions [7]. ($P(1..i)$ denotes the initial part of the sequence P of length i):

$+\partial$: If $P(i+1) = +\partial q$ then either
(1) $+\Delta q \in P(1..i)$ or
(2) (2.1) $\exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P(1..i)$ and
(2.2) $-\Delta \sim q \in P(1..i)$ and
(2.3) $\forall s \in R[\sim q]$ either
(2.3.1) $\exists a \in A(s) : -\partial a \in P(1..i)$ or
(2.3.2) $\exists t \in R_{sd}[q]$ such that
$\forall a \in A(t) : +\partial a \in P(1..i)$ and $t > s$

Let us work through this condition. 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). Also (2.3) we must consider the set of all rules which are not known to be inapplicable and which have head $\sim q$ (note that here we consider defeaters, too, whereas they could not be used to support the conclusion q; this is in line with the motivation of defeaters given earlier). Essentially each such rule s attacks the conclusion q. For q to be provable, each such rule s must be counterattacked by a rule t with head q with the following properties: (i) t must be applicable at this point, and (ii) t must be stronger than s. Thus each attack on the conclusion q must be counterattacked by a stronger rule. In other words, r and the rules t form a team (for q) that defeats the rules s.

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 DL. Obviously, the underlying logical language we use is the language of DL; however, we consider facts to be strict rules with empty bodies.

Arguments are often defined to be either proof trees or monotonic derivations in the underlying logic. However, DL requires a more refined definition: as we have seen in the previous section, rules form teams to support conclusions. Thus we extend the simpler notion of argument and we allow arguments to be sets of proof trees (see example 2 for a more detailed explanation). DL also 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.

The arcs in a proof tree is labelled by the rules used to obtain them.

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*. As we shall see shortly, proof trees are only indirectly related to DL derivations.

An *argument* for a literal p is a set of proof trees for p. We write $r \in A$ to denote that rule r is used in a proof tree in argument A. A *(proper) subargument* of an argument A is a subtree of a proof tree in A. We say that an argument A is *finite* if every proof tree in A is finite. An argument A is *strict* if every proof tree in A is strict. If an argument is not strict it is *defeasible*. An argument A for p is a *supportive argument* if every proof tree for p in it is supportive.

DL 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, but if we changed its definition, replacing "every" with "some", then we would have the following scenario: let A and B be, respectively, the arguments $\{r_1 : \Rightarrow p, r_2 : \rightsquigarrow p\}$ and $\{r_3 : \Rightarrow \neg p\}$ where $r_1 < r_3$ and $r_3 < r_2$. A would be a supportive argument, and its conclusion, p, derivable, but DL is not able to derive $+\partial p$.

An argument is based on an ordered theory $(R,<)$ if every rule in the argument is a ground instance of a rule in R. Clearly, a defeasible theory $(F,R,<)$ can be considered an ordered theory $(F \cup R,<)$. $Args_P$ is the set of arguments based on the ordered theory P.

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

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

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

This characterization is straightforward, since strict rules are the monotonic subset of DL. Characterizing defeasible provability requires more definitions.

An argument A *attacks* an argument B if a conclusion of A is the complement of a conclusion of B.

An argument A *defeats* a defeasible argument B at q if there exists $r_A \in A$ and $r_B \in B$ with conclusions $\sim q$ and q, respectively, such that $r_A \not< r_B$. A set of arguments S defeats a defeasible argument B if there is $A \in S$ that defeats B.

Example 1 Let D be a defeasible theory containing the rules

$$\begin{array}{llll} r_1: & a \Rightarrow p & \qquad r_3: & b \Rightarrow \neg p \\ r_2: & p \Rightarrow q & \qquad r_4: & \neg p \Rightarrow \neg q \end{array}$$

the facts a, b, and the superiority relation is $r_4 < r_2$. We consider the arguments

$$\begin{array}{lccccc} A: & a & \Rightarrow & p & \Rightarrow & q \\ B: & b & \Rightarrow & \neg p & \Rightarrow & \neg q \end{array}$$

A defeats B both at $\neg p$, because $r_4 < r_2$, and at $\neg q$, because there is no superiority relation between r_1 and r_3. B defeats A at p for the same reason A defeats B at $\neg p$. □

An argument A *team defeats* a defeasible argument B at q if for every $r_B \in B$ with conclusion q there exists a supportive rule $r_A \in A$ with conclusion $\sim q$ such that $r_B < r_A$.

Example 2 Let D be a defeasible theory containing the rules

$$\begin{array}{llll} r_1: & a_1 \Rightarrow p & \qquad r_3: & b_1 \Rightarrow \neg p \\ r_2: & a_2 \Rightarrow p & \qquad r_4: & b_2 \Rightarrow \neg p \end{array}$$

the facts a_1, a_2, b_1, b_2, and the superiority relation is $r_3 < r_1$, $r_4 < r_2$. Consider the argument A_p

$$\begin{array}{lcl} a_1 & & \\ & \searrow\!\!\!\searrow & \\ & & p \\ & \nearrow\!\!\!\nearrow & \\ a_2 & & \end{array}$$

containing two proof trees. A_p team defeats:

- the argument $B_1: b_1 \Rightarrow \neg p$ since $r_3 < r_1$;
- the argument $B_2: b_2 \Rightarrow \neg p$ since $r_4 < r_2$;
- the argument $B_3:$ $\begin{array}{lcl} b_1 & & \\ & \searrow\!\!\!\searrow & \\ & & \neg p \\ & \nearrow\!\!\!\nearrow & \\ b_2 & & \end{array}$ since $r_3 < r_1$ and $r_4 < r_2$. □

Example 3 Some explanation is due to justify the exclusion of arguments ending with a defeaters from the notion of team defeat (see also the comment about supportive arguments above). First of all one of the main aims of such a notion is to help establishing conclusive arguments (that is, arguments that can be used to draw positive conclusions). Let us consider a defeasible theory D' obtained from the defeasible theory of example 2 by replacing the rule r_2 by the defeater $r_2 : a_2 \rightsquigarrow p$. Let A be the argument

$$\begin{array}{lcl} a_1 & & \\ & \searrow\!\!\!\searrow & \\ & & p \\ & \nearrow & \\ a_2 & & \end{array}$$

Since A contains a defeater, it cannot team defeat the argument B_3 of the previous example. Let us compare this situation with the definition of $+\partial$. r_2 cannot be used to derive p: it is a defeater. On the other hand r_1 could be used to derive p if there is no applicable rule for $\neg p$. But, in this case, we have r_3 and r_4, and, when r_4 is applicable, we have a conflict between r_1 and r_4. However, p could be reinstated if there is an applicable supportive rule stronger than r_4 (2.3.2), but in this case the only rule stronger than r_4 is the defeater r_2, and so p cannot be concluded from r_1 and r_2. □

An argument A is *supported* by a set of arguments S if every conclusion in A is also the conclusion of a supportive argument in S.

In an ordered theory P, let $strong_P(S)$ be the set of arguments of P, all of whose proper subarguments are supported by S. Obviously $S \subseteq strong_P(S)$. Also note that, if $A_1, A_2 \in strong_P(S)$ are arguments for a literal q, then $A_1 \cup A_2 \in strong_P(S)$. Thus there is a maximal argument for q in $strong_P(S)$, which we denote by $max(q,S)$. A defeasible argument A is *undercut* by a set of arguments S if there is a literal q such that $strong_P(S)$ defeats a proper subargument of A at q, and A does not team defeat $max(\neg q, S)$ at $\neg q$.

Example 4 We consider again the defeasible theory D of example 1. 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 $B : b \Rightarrow \neg p$ is in $strong_D(S)$ and it is the maximal argument for $\neg p$. Moreover B defeats a proper subargument of A at p, but it is not team defeated by A at p. □

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 next example explains the reason for the use of team defeat in the definition of undercut.

Example 5 Let D' be the defeasible theory obtained from the defeasible theory of example 2 by adding the rule $p \Rightarrow q$. And let $S = \{a_1, a_2, b_1, b_2\}$ be a set of arguments. Let A_q be the argument

$$\begin{array}{lcl} a_1 & & \\ & \searrow\!\!\!\searrow & \\ & & p \Rightarrow q \\ & \nearrow\!\!\!\nearrow & \\ a_2 & & \end{array}$$

Notice that each of the arguments B_1, B_2, and B_3 of example 2 defeats A at p, but A is not undercut by S at p since the argument A team defeats the $max(\neg p, S)$ in $strong_{D'}(S)$. Here $max(\neg p, S)$ is the argument B_3 of example 2. Thus team defeat in the definition of undercut is necessary to be consistent with the use of team defeat at the top level of arguments. □

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

An argument A for p is *acceptable* w.r.t a set of arguments S if

1 A is strict, or
2a every proper subargument of A is supported by S, and
2b every argument attacking A is either undercut by S or team defeated by A.

Let P be an ordered theory. We define J_i^P as follows.

- $J_0^P = \emptyset$
- $J_{i+1}^P = \{a \in Args_P \mid a$ is acceptable w.r.t. $J_i^P\}$

The set of *justified arguments* in an ordered theory P is $JArgs^P = \cup_{i=1}^{\infty} J_i^P$. A literal p is *justified* if it is the conclusion of a supportive argument in $JArg^P$

Theorem 2 *Let P be a defeasible theory. Let p be a literal.* $P \vdash +\partial p$ *iff p is justified.*

This theorem provides a characterization of positive defeasible conclusions in DL by means of justified arguments.

Example 6 Given the theory D' of example 5, $J_1^{D'} = \{a_1, a_2, b_1, b_2\}$, and the argument A_p of example 2 is in $J_2^{D'}$, since it is acceptable w.r.t. $J_1^{D'}$: every proper subargument is supported, and the attacking arguments are team defeated. At this point it is immediate to see that the argument A_q of example 5 is in $J_3^{D'}$. Moreover $JArgs_{D'} = J_3^{D'}$. □

That an argument A is justified means that it resists every reasonable refutation. However, DL is more expressive since it is able to say when a conclusion is demonstrably non provable ($-\partial$). 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.

An argument A is *rejected* by sets of arguments S and T when

1 A is not strict, and either
2a a proper subargument of A is in S, or
2b there exists an argument B attacking A, such that: B is supported by T, and A does not team defeat B.

We define R_i^P as follows.

- $R_0^P = \emptyset$
- $R_{i+1}^P = \{a \in Args_P \mid a \text{ is rejected by } R_i^P \text{ and } JArgs^P\}$

The set of *rejected arguments* in an ordered theory P is $RArgs^P = \cup_{i=1}^{\infty} R_i^P$. A literal p is *rejected* if there is no argument in $Args_P - RArgs^P$ that ends with a supportive rule for p.

Theorem 3 *Let P be a defeasible theory. Let p be a literal.* $P \vdash -\partial p$ *iff p is rejected.*

Example 7 The following DL theory illustrates why $RArgs^P$ needs to be constructed iteratively, even after all the justified literals have been identified.

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

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

and the fact b_0. The superiority relation is empty.

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

The arguments defined by this theory are, for each i :

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

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 team 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 $RArgs^P$ must be iterative, even after all the justified literals have been identified. □

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

Example 8 (Self-defeating arguments) In this example we show how our framework deals with the so called self-defeating arguments. Consider the defeasible theory with no facts, an empty superiority relation and the following rules:

$$true \Rightarrow p \qquad\qquad p \Rightarrow \neg p$$

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

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

Here A_2 is a self-defeating argument. Since the superiority relation is empty there is no team defeat. A_1, although supported by J_0^P, is not acceptable in J_0^P since there is an attacking argument, A_2, which is not undercut by J_0^P: no proper subargument of A_2 is defeated by an argument supported by J_0^P. For the same reason A_2 is not acceptable in J_0^P. Consequently $J_1^P = J_0^P$, and therefore $JArg_P$ is empty. Furthermore, $A_2 \in RArg^P$. The reason why A_2 is rejected is the following: although A_1 is not justified, it is supported by $JArgs^P$, 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 $JArgs^P$: as we have already seen $true \Rightarrow p$ is not a justified argument. □

Example 9 (Circular arguments) Here we examine 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

$$p \Rightarrow q \qquad\qquad q \Rightarrow p$$

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

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

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.

The meaning of the theory at hand is that if p then normally q, and if q then normally p. Thus this amounts to say that normally p and q are equivalent. We add to D_1 the following rules:

$$q \Rightarrow r \qquad\qquad true \Rightarrow \neg r$$

obtaining the defeasible theory D_2. In this scenario each argument for r is infinite, circular, and rejected since there is a supported argument for $\neg r$. However, the argument $A_3 : true \Rightarrow \neg r$ is not justified, since each argument for r attacks it and is not undercut (no argument attacks a proper subargument of an argument for r).

Finally D_3 is obtained from D_2 by adding the rule $true \Rightarrow \neg p$. Now A_3 becomes justified since, trivially, the argument $A_4 : true \Rightarrow p$ is supported by $J_0^{D_3}$, A_3 attacks A_2, and therefore each argument for r is undercut. □

4 Related Works

[16] proposes an abstract defeasible reasoning framework that is achieved by mapping elements of defeasible reasoning into the default reasoning framework of [8]. While this framework is suitable for developing new defeasible reasoning languages, it is not appropriate for characterizing DL because:

- [16, 8] do not address direct scepticism.
- [8] does not address Kunen's semantics of logic programs which provides a characterization of failure-to-prove in DL [18].
- The correctness of the mapping needs to be established if [16] is to be applied to an existing language like DL. In fact the representation of priorities is inappropriate for DL, although results of [3, 1] might be adapted to remedy this point.

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 DL, 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^P and $JArgs_P$, 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.

Prakken and Sartor [21, 20], motivated by legal reasoning, have proposed an argumentation system that combines the language of extended logic programming with a well-founded semantics. The use of this semantics makes Prakken and Sartor's system not directly sceptical. It is worth noting that our definition of defeat is the same as that of rebut in [21, 20], but the systems differ on the notion of acceptability of arguments. Moreover, Prakken and Sartor do not address the question of teams of rules.

On the other hand Simari and Loui's system [23] deals with teams of arguments/rules but it is characterized by Dung's grounded semantics, which corresponds to an ambiguity propagating variant of DL (see [2, 11]).

Among other contributions, [9] provides a sceptical argumentation theoretic semantics and shows that LPwNF – which is weaker, but very similar to DL [6] – is sound with respect to this semantics. However, both LPwNF and DL are not complete with respect to this semantics.

Acknowledgements

We thank Grigoris Antoniou, David Billington and Alejandro Garcia for fruitful discussions on DL and argumentation. 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. Technical Report, CIT, Griffith University, 2000.

[2] G. Antoniou, D. Billington, G. Governatori, M.J. Maher and A. Rock. A Family of Defeasible Reasoning Logics and its Implementation. In W. Horn (ed.): *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, 2000.

[3] G. Antoniou, D. Billington and M.J. Maher. Normal Forms for Defeasible Logic. In *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. In *Proc. of the 32nd Annual Hawaii International Conference on System Sciences*. IEEE Press, 1999.

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

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

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

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

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

[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, M.J. Maher, G. Antoniou and D. Billington. Argumentation Semantics for Defeasible Logics. In *Proceeding of PRICAI 2000*. Springer, 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*, 9(1999), 215-261.

[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.

[25] G. Wagner. Ex Contradictione Nihil Sequitur. In *Proc. IJCAI-91*, 538–546 , Morgan Kaufmann, 1991.