TLP **6** (6): 703–735, 2006. © 2006 Cambridge University Press
doi:10.1017/S1471068406002778 Printed in the United Kingdom

Embedding defeasible logic into logic programming

GRIGORIS ANTONIOU
Institute of Computer Science, FORTH, Greece
(*e-mail:* antoniou@ics.forth.gr)

DAVID BILLINGTON
School of ICT, Griffith University, Australia
(*e-mail:* d.billington@griffith.edu.au)

GUIDO GOVERNATORI
School of ITEE, University of Queensland, Australia
(*e-mail:* guido@itee.uq.edu.au)

MICHAEL J. MAHER
National ICT Australia, c/o UNSW, Australia
(*e-mail:* mmaher@cse.unsw.edu.au)

submitted 20 September 2004; revised 12 October 2005; accepted 8 November 2005

Abstract

Defeasible reasoning is a simple but efficient approach to nonmonotonic reasoning that has recently attracted considerable interest and that has found various applications. Defeasible logic and its variants are an important family of defeasible reasoning methods. So far no relationship has been established between defeasible logic and mainstream nonmonotonic reasoning approaches. In this paper we establish close links to known semantics of logic programs. In particular, we give a translation of a defeasible theory D into a meta-program $P(D)$. We show that under a condition of decisiveness, the defeasible consequences of D correspond exactly to the sceptical conclusions of $P(D)$ under the stable model semantics. Without decisiveness, the result holds only in one direction (all defeasible consequences of D are included in all stable models of $P(D)$). If we wish a complete embedding for the general case, we need to use the Kunen semantics of $P(D)$, instead.

KEYWORDS: Defeasible logic, stable semantics, Kunen semantics, non-monotonic logic

1 Introduction

Defeasible reasoning is a nonmonotonic reasoning (Marek and Truszczynski 1993) approach in which the gaps due to incomplete information are closed through the use of defeasible rules that are usually appropriate. Defeasible logics were introduced and developed by Nute (1994) over several years. These logics perform defeasible

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

reasoning, where a conclusion supported by a rule might be overturned by the effect of another rule. Roughly, a proposition p can be defeasibly proved ($+\partial p$) only when a rule supports it, and it has been demonstrated that no applicable rule supports $\neg p$; this demonstration makes use of statements $-\partial q$ which mean intuitively that an attempt to prove q defeasibly has failed finitely. These logics also have a monotonic reasoning component, and a priority on rules. One advantage of Nute's design was that it was aimed at supporting efficient reasoning, and in our work we follow that philosophy.

Defeasible reasoning has recently attracted considerable interest. Its use in various application domains has been advocated, including the modelling of regulations and business rules (Morgensten 1998; Antoniou *et al.* 1999), modelling of contracts (Grosof *et al.* 1999; Grosof 2004; Governatori 2005), legal reasoning (Prakken 1997; Governatori *et al.* 2005), agent negotiations (Governatori *et al.* 2001), modelling of agents and agent societies (Governatori and Rotolo 2004; Governatori and Padmanabhan 2003; Governatori *et al.* 2004), and applications to the Semantic Web (Bassiliades *et al.* 2004; Antoniou 2002). In fact, defeasible reasoning (in the form of courteous logic programs (Grosof 1997; Grosof *et al.* 1999)) provides a foundation for IBM's Business Rules Markup Language and for current W3C activities on rules (Grosof and Poon 2003; Grosof *et al.* 2002). In addition, defeasible theories, describing policies of business activities, can be mined efficiently from appropriate datasets (Johnston and Governatori 2003). Therefore defeasible reasoning is a promising subarea in nonmonotonic reasoning as far as applications and integration to mainstream IT is concerned.

Recent theoretical work on defeasible logics has: (i) established some relationships to other logic programming approaches without negation as failure (Antoniou *et al.* 2000); (ii) analysed the formal properties of these logics (Antoniou *et al.* 2001; Maher 2000; Maher 2001), as well as formal semantics for them in form of model theoretic semantics (Maher 2002) and argumentation semantics (Governatori *et al.* 2004), and (iii) has delivered efficient implementations (Maher *et al.* 2001).

However the problem remains that defeasible logic is not firmly linked to the mainstream of nonmonotonic reasoning, in particular the semantics of logic programs. This paper aims at resolving this problem. We use the translation of a defeasible theory D into a logic meta-program $\mathscr{M}$ proposed in Maher and Governatori (1999). For this translation we can show that, for the propositional case:

$$p \textit{ is defeasibly provable in } D \iff p \textit{ is included in all stable models of } \mathscr{M}. \quad (*)$$

However, this result can only be shown under the additional condition of *decisiveness*: the absence of cycles in the atom dependency graph.

If we wish to drop decisiveness, (*) holds only in one direction, from left to right. We show that if we wish the equivalence in the general case, we need to use another semantics for logic programs, namely Kunen semantics (Kunen 1987). In addition the possibility of cycles in the atom dependency graph of a defeasible theory prevents Defeasible Logic to be characterised by well-founded semantics (Gelder *et al.* 1991). It is possible to modify Defeasible Logic to accommodate well-founded semantics (Maher and Governatori 1999) even if this results in a more expensive computational model.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

The paper is organised as follows. Sections 2 and 3 present the basics of defeasible logic and logic programming semantics, respectively. Section 4 presents our translation of defeasible theories in logic programs, while Section 5 contains the main results.

2 Defeasible logic

2.1 A language for defeasible reasoning

A defeasible theory (a knowledge base in defeasible logic) consists of five different kinds of knowledge: facts, strict rules, defeasible rules, defeaters, and a superiority relation.

Facts are literals that are treated as known knowledge (given or observed facts of a case).

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, not inferior, 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 does not 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 defeasible rules

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

which contradict one another, no conclusive decision can be made about whether a bird with broken wings can fly. But if we introduce a superiority relation $>$ with $r' > r$, with the intended meaning that r' is strictly stronger than r, then we can indeed conclude that the bird cannot fly.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

It is worth noting that, in defeasible logic, priorities are *local* in the following sense: Two rules are considered to be competing with one another only if they have complementary heads. Thus, since the superiority relation is used to resolve conflicts among competing rules, it is only used to compare rules with complementary heads; the information $r > r'$ for rules r, r' without complementary heads may be part of the superiority relation, but has no effect on the proof theory.

2.2 Formal definition

In this paper we restrict attention to essentially propositional defeasible logic. Rules with free variables are interpreted as rule schemas, that is, as the set of all ground instances; in such cases we assume that the Herbrand universe is finite. We assume that the reader is familiar with the notation and basic notions of propositional logic. 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).

Rules are defined over a *language* (or *signature*) Σ, the set of propositions (atoms) and labels that may be used in the rule.

A *rule* $r : A(r) \hookrightarrow C(r)$ consists of its unique *label* r, its *antecedent* $A(r)$ ($A(r)$ may be omitted if it is the empty set) which is a finite set of literals, an arrow $\hookrightarrow$ (which is a placeholder for concrete arrows to be introduced in a moment), and its *head* (or *consequent*) $C(r)$ which is a literal. In writing rules we omit set notation for antecedents and sometimes we omit the label when it is not relevant for the context. There are three kinds of rules, each represented by a different arrow. Strict rules use $\rightarrow$, defeasible rules use $\Rightarrow$, and defeaters use $\rightsquigarrow$.

Given a set R of rules, we denote the set of all strict rules in R by R_s, and the set of strict and defeasible rules in R by R_{sd}. $R[q]$ denotes the set of rules in R with consequent q.

A *superiority relation on* R is a relation $>$ on R. When $r_1 > r_2$, then r_1 is called *superior* to r_2, and r_2 *inferior* to r_1. Intuitively, $r_1 > r_2$ expresses that r_1 overrules r_2, should both rules be applicable. $>$ must be acyclic (that is, its transitive closure must be irreflexive).

A *defeasible theory* D is a triple $(F, R, >)$ where F is a finite set of facts, R a finite set of rules, and $>$ an acyclic superiority relation on R.

2.3 Proof theory

A *conclusion* of a defeasible theory D is a tagged literal. A conclusion has one of the following four forms:

- $+\Delta q$, which is intended to mean that the literal q is definitely provable, using only strict rules.
- $-\Delta q$, which is intended to mean that q is provably not definitely provable (finite failure).
- $+\partial q$, 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.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Provability is defined below. It 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 the following conditions. The conditions are essentially inference rules phrased as conditions on proofs. $P(1..i)$ denotes the initial part of the sequence P of length i.

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

That means, to prove $+\Delta q$ we need to establish a proof for q using facts and strict rules only. This is a deduction in the classical sense – no proofs for the negation of q need to be considered (in contrast to defeasible provability below, where opposing chains of reasoning must be taken into account, too).

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

To prove $-\Delta q$, i.e. that q is not definitely provable, q must not be a fact. In addition, we need to establish that every strict rule with head q is *known to be* inapplicable. Thus for every such rule r there must be at least one antecedent a for which we have established that a is not definitely provable ($-\Delta a$).

$+\partial$: 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 illustrate this definition. 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 above). 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 (i.e. superior to) s. Thus each attack on the conclusion q must be counterattacked by a stronger rule.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

$-\partial$: If $P(i+1) = -\partial q$ then
(1) $-\Delta q \in P(1..i)$ and
(2) (2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r) : -\partial a \in P(1..i)$ or
(2.2) $+\Delta{\sim}q \in P(1..i)$ or
(2.3) $\exists s \in R[{\sim}q]$ such that
(2.3.1) $\forall a \in A(s) : +\partial a \in P(1..i)$ and
(2.3.2) $\forall t \in R_{sd}[q]$ either
$\exists a \in A(t) : -\partial a \in P(1..i)$ or $t \not> s$

To prove that q is not defeasibly provable, we must first establish that it is not definitely provable. Then we must establish that it cannot be proven using the defeasible part of the theory. There are three possibilities to achieve this: either we have established that none of the (strict and defeasible) rules with head q can be applied (2.1); or ${\sim}q$ is definitely provable (2.2); or there must be an applicable rule s with head ${\sim}q$ such that no applicable rule t with head q is superior to s.

In general the inference conditions for a negative proof tag (i.e., $-\Delta$, $-\partial$) explore all the possibilities to derive a literal (with a given proof strength) before stating that the literal is not provable (with the same proof strength). Thus conclusions with these tags are the outcome of a constructive proof that the corresponding positive conclusion cannot be obtained. As a result, there is a close relationship between the inference rules for $+\partial$ and $-\partial$, (and also between those for $+\Delta$ and $-\Delta$). The structure of the inference rules is the same, but the conditions are negated in some sense. To be more precise the inference conditions for a negative proof tag are derived from the inference conditions for the corresponding positive proof tag by applying the Principle of Strong Negation introduced in (Antoniou *et al.* 2000). The strong negation of a formula is closely related to the function that simplifies a formula by moving all negations to an innermost position in the resulting formula and replaces the positive tags with the respective negative tags and vice-versa.

The elements of a derivation are called *lines* of the derivation. We say that a tagged literal L is *provable* in $D = (F, R, >)$, denoted by $D \vdash L$, iff there is a derivation P in D such that L is a line of P.

Defeasible logic is closely related to several non-monotonic logics (Antoniou *et al.* 1999). In particular, the "directly skeptical" semantics of non-monotonic inheritance networks (Horty *et al.* 1987) can be considered an instance of inference in DL once an appropriate superiority relation, derived from the topology of the network, is fixed (Billington *et al.* 1990).

A defeasible theory D is *coherent*[1] if for no literal p both $D \vdash +\Delta p$ and $D \vdash -\Delta p$, and $D \vdash +\partial p$ and $D \vdash -\partial p$; and *relatively consistent* if whenever $D \vdash +\partial p$ and $D \vdash +\partial{\sim}p$, for some p, then also $D \vdash +\Delta p$ and $D \vdash +\Delta{\sim}p$. Intuitively, coherence says that no literal is simultaneously provable and unprovable. Consistency says that a literal and its negation can both be defeasibly provable only when it and its negation are definitely provable; hence defeasible inference does not introduce

[1] Notice that here coherent has a different meaning than other works on logic programming for example (Alferes and Pereira 1993a; Alferes and Pereira 1993b).

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

inconsistency. A logic is coherent (relatively consistent) if the meaning of each theory of the logic, when expressed as an extension (i.e., when we consider the set of all the consequences of the theory), is coherent (relatively consistent), that is it is not possible to prove a formula and its negation unless the monotonic part of the theory proves them.

Proposition 2.1 (*Billington 1993*)
Defeasible logic is coherent and relatively consistent.

Consistency and coherence address the issue whether and how it is possible to derive "conflicts" from defeasible theories. On the other side we can ask under which conditions defeasible theories are *complete*, in the sense that for every literal in the theory it is possible to decide whether the literal is provable/non provable from the theory.

In the rest of this section we will study conditions under which it is possible to guarantee completeness of a defeasible theory.

Given a defeasible theory $D = (F, R, >)$ a literal q is *strictly unknowable in D* iff $D \nvdash +\Delta q$ and $D \nvdash -\Delta q$. A literal is *defeasibly unknowable in D* iff $D \nvdash +\partial q$ and $D \nvdash -\partial q$. A literal is *unknowable* in D iff it is either strictly unknowable in D or defeasibly unknowable in D.

The *dependency graph* of D, $DG(D)$, is the directed graph defined as follows: the set of points of $DG(D)$ is $\{\{q, \sim q\} : q \text{ is a literal in } D\}$. The set of arcs of $DG(D)$ is $\{(\{b, \sim b\}, \{a, \sim a\}) | \exists r \in R_s[b] : a \in A(r)\}$. Let $U(D)$ be the subgraph of $DG(D)$ restricted to the literals that are unknowable in D, i.e., the set of points of $U(D)$ is $\{\{q, \sim q\} : q \text{ is unknowable in } D\}$.

The next Lemmata show the mutual relationships among unknowable literals in a Defeasible theory, and general properties of graphs of unknowable literals.

Lemma 2.1
Let $D = (F, R, >)$ be a Defeasible Theory. The out-degree of $U(D)$ is at least 1.

Proof
We have to consider two cases: strictly unknowable literals and defeasibly unknowable literals.

Let q be a strictly unknowable literal, then there is a point $\{q, \sim q\}$ in $U(D)$. Then, by definition, $D \nvdash +\Delta q$ and $D \nvdash -\Delta q$.[2] Hence $q \notin F$ and $\forall r \in R_s[q] \exists a \in A(r)$ such that $D \nvdash +\Delta a$. Since $D \nvdash -\Delta q$ and $q \notin F$, then $\exists r \in R_s[q]$ such that $\forall a \in A(r)$ $D \nvdash -\Delta a$. So $\exists r \in R_s[q]$ $\exists a \in A(r)$ such that $D \nvdash +\Delta a$ and $D \nvdash -\Delta a$. Thus $(\{q, \sim q\}, \{a, \sim a\})$ is an arc in $U(D)$.

If q is a defeasibly unknowable literal then we reason as follows: let $\{q, \sim q\}$ be the point in $U(D)$ corresponding to q. Then $D \nvdash +\partial q$ and $D \nvdash -\partial q$.

Since $D \nvdash +\partial q$ we have the following:

a1) $D \nvdash +\Delta q$ and
a2) one of the following three holds:

[2] The case of $\sim q$ is identical.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

a2.1) $\forall r \in R_{sd}[p] \exists a \in A(r) : D \nvdash +\partial a$; or
a2.2) $D \nvdash -\Delta \sim q$; or
a2.3) $\exists s \in R[\sim q]$ such that
  a2.3.1) $\forall a \in A(s)$ $D \nvdash -\partial a$, and
  a2.3.2) $\forall t \in R_{sd}[q]$ either
    $\exists a \in A(t) : D \nvdash +\partial a$ or $t \not> s$.

Since $D \nvdash -\partial q$ we have the following.

b1) $D \nvdash -\Delta q$; or
b2) All of the following three hold:
  b2.1) $\exists r \in R_{sd}[q] \forall a \in A(r)$ $D \nvdash -\partial a$, and
  b2.2) $D \nvdash +\Delta \sim q$, and
  b2.3) $\exists s \in R[\sim q]$ either
    b2.3.1) $\exists a \in A(s), D \nvdash +\partial a$; or
    b2.3.2) $\exists t \in R_{sd}[q]$ such that
      $\forall a \in A(t), D \nvdash -\partial a$ and $t > s$.

By a1, $D \nvdash +\Delta q$. If b1 holds then q is strictly unknowable and we can repeat the first part of the proof. So suppose that b2 holds.

If a2.1 holds then by b2.1 $\exists r \in R_{sd}[p] \exists a \in A(r)$ such that a is defeasibly unknowable in D. Thus $(\{q, \sim q\}, \{a, \sim a\})$ is an arc in $U(D)$. If a2.2 holds then by b2.2 $\sim p$ is a strictly unknowable literal in D, and we have already proved the property in this case.

So suppose that a2.3 holds. If b2.3 holds then $\exists s \in R[\sim p] \exists a \in A(s)$ such that a is a defeasibly unknowable literal in D. Thus $(\{q, \sim q\}, \{a, \sim a\})$ is an arc in $U(D)$. So suppose that b2.3.2 holds. Then $t > s$ and so a2.3.2 holds. Hence $\exists t \in R_{sd}[q] \exists a \in A(t)$ such that a is defeasibly unknowable in D. Thus $(\{q, \sim q\}, \{a, \sim a\})$ is an arc in $U(D)$.

Therefore in all cases the out-degree of $\{q, \sim q\}$ is at least 1. □

Given a graph, a *walk* is an alternating sequence of vertices and edges, with each edge being incident to the vertices immediately preceeding and succeeding it in the sequence.

The set of points in a walk W in the dependency graph of a theory D is denoted by $Points(W)$. A walk W *ends in a cycle* iff W is finite and the last point of W occurs at least twice. A walk is *complete* iff either

1. $Points(W)$ is infinite; or
2. a point in W has out-degree zero; or
3. W ends in a cycle.

Lemma 2.2
Let $D = (F, R, >)$ be a defeasibly theory. Then the following are equivalent.

1. There is a literal which is unknowable in D.
2. $U(D)$ is not empty.
3. There is a walk in $U(D)$.
4. There is a complete walk in $U(D)$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

5. There is a complete walk in $U(D)$ and whenever W is a complete walk in $U(D)$ then either $Points(W)$ is infinite or W ends in a cycle.

Proof
1 and 2 are clearly equivalent. If 2 holds then by Lemma 2.1, 3 holds. If 3 holds, then 4 holds since every walk can be extended to a complete walk. By Lemma 2.1, every point in $U(D)$ has out-degree of at least 1, and so if 4 holds, then 5 holds. If 5 holds since walks are not empty 2 holds. □

A defeasible theory D is called *decisive* iff the dependency graph of D is acyclic.

The following proposition provides a sufficient condition to determine completeness of a defeasible theory.

Theorem 2.1
If D is decisive, then for each literal p:

(a) either $D \vdash +\Delta p$ or $D \vdash -\Delta p$
(b) either $D \vdash +\partial p$ or $D \vdash -\partial p$.

Proof
We prove the contrapositive, i.e., suppose there are unknowable literals. If there are unknowable literals then by Lemma 2.2, there is a walk in $U(D)$ such that $Points(W)$ is infinite or W ends in a cycle. Since there are only finitely many rules in D, $Points(W)$ is finite, thus W ends in a cycle. Thus $U(D)$ has a cycle and $U(D)$ is a subgraph of $DG(D)$, thus D is not decisive. □

Not every defeasible theory satisfies this property. For example, in the theory consisting of the single rule

$$r_1 : p \Rightarrow p$$

neither $-\partial p$ nor $+\partial p$ is provable.

Notice, however, that there are complete non decisive theories. If we extend the above theory with the rule

$$r_2 : \Rightarrow \neg p$$

and the superiority relation $>$ is defined as $r_2 > r_1$ then we can prove both $-\partial p$ and $+\partial \neg p$, thus the resulting theory is not decisive, but complete.

2.4 A bottom-up characterization of defeasible logic

The proof theory provides the basis for a top-down (backward-chaining) implementation of the logic. In fact the *Deimos* system (Maher *et al.* 2001) is based directly on the proof theory described above. However, there are advantages to a bottom-up (forward-chaining) implementation. In particular this presentation of Defeasible Logic provides a both a set theoretic and a declarative computational model of the logic compared to the procedural characterisation of the top-down definitions. This allows us to describe a defeasible theory as an extension (i.e., set of all conclusions provable from it) and to deal with finite as well as infinite theories.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

This is not possible in the other approach since derivations are required to be finite sequences of (tagged) literals and they are described in term of combinatorial constructions. Furthermore, a bottom-up definition of the logic provides a bridge to later considerations. For these reasons we now provide a bottom-up definition of Defeasible Logic.

We associate with D an operator $\mathscr{T}_D$ which works on 4-tuples of sets of literals. We call such 4-tuples an *extension.*

$$\mathscr{T}_D(+\Delta, -\Delta, +\partial, -\partial) = (+\Delta', -\Delta', +\partial', -\partial') \text{ where}$$

$$+\Delta' = F \cup \{q \mid \exists r \in R_s[q]\ A(r) \subseteq +\Delta\}$$

$$-\Delta' = -\Delta \cup (\{q \mid \forall r \in R_s[q]\ A(r) \cap -\Delta \neq \emptyset\} - F)$$

$$\begin{aligned} +\partial' = +\Delta \cup \{q \mid\ & \exists r \in R_{sd}[q]\ A(r) \subseteq +\partial, \\ & \sim q \in -\Delta, \text{ and} \\ & \forall s \in R[\sim q] \text{ either} \\ & \quad A(s) \cap -\partial \neq \emptyset, \text{ or} \\ & \quad \exists t \in R[q] \text{ such that} \\ & \quad\quad A(t) \subseteq +\partial \text{ and } t > s\} \end{aligned}$$

$$\begin{aligned} -\partial' = \{q \in -\Delta \mid\ & \forall r \in R_{sd}[q]\ A(r) \cap -\partial \neq \emptyset, \text{ or} \\ & \sim q \in +\Delta, \text{ or} \\ & \exists s \in R[\sim q] \text{ such that } A(s) \subseteq +\partial \text{ and} \\ & \forall t \in R[q] \text{ either} \\ & \quad A(t) \cap -\partial \neq \emptyset, \text{ or} \\ & \quad t \not> s\} \end{aligned}$$

The set of extensions forms a complete lattice under the pointwise containment ordering[3], with $\perp = (\emptyset, \emptyset, \emptyset, \emptyset)$ as its least element. The least upper bound operation is the pointwise union[4], which is represented by $\cup$.

The sequence of repeated applications of $\mathscr{T}_D$ to $\perp$, called the *Kleene sequence* of $\mathscr{T}_D$, is defined as follows:

- $\mathscr{T}_D \uparrow 0 = \perp$;
- $\mathscr{T}_D \uparrow (\alpha + 1) = \mathscr{T}_D(\mathscr{T}_D \uparrow \alpha)$;
- $\mathscr{T}_D \uparrow \alpha = \bigcup_{\beta<\alpha} \mathscr{T}_D \uparrow \beta$ if α is a limit ordinal.

Proposition 2.2
$\mathscr{T}_D$ is monotonic and the Kleene sequence from $\perp$ is increasing. Thus the limit $F = (+\Delta_F, -\Delta_F, +\partial_F, -\partial_F)$ of all finite elements in the sequence exists, and $\mathscr{T}_D$ has a least fixpoint $L = (+\Delta_L, -\Delta_L, +\partial_L, -\partial_L)$. When D is a finite propositional defeasible theory $F = L$.

[3] $(a_1, a_2, a_3, a_4) \leqslant (b_1, b_2, b_3, b_4)$ iff $a_i \subseteq b_i$ for $i = 1, 2, 3, 4$.
[4] Given two n-tuple of sets $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_2)$ the pointwise union of a and b is defined as follows: $a \cup b = (a_1 \cup b_1, \ldots, a_n \cup b_n)$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Proof
We prove by induction that $\mathscr{T}_D$ is pointwise monotonic. The other properties follow from standard and well-know set theoretic arguments.

The inductive base is trivial since the elements of $\perp$ are $\emptyset$.

For the inductive step we have four cases, where the inductive hypothesis amounts to: $+\Delta^{n-1} \subseteq +\Delta^n$, $-\Delta^{n-1} \subseteq -\Delta^n$, $+\partial^{n-1} \subseteq +\partial^n$ and $-\partial^{n-1} \subseteq -\partial^n$.

Case $+\Delta$. Let us investigate the reasons why $p \in +\Delta^n$: if $p \in F$, then, trivially, for all m $p \in +\Delta^m$; hence $p \in +\Delta^{n+1}$. Otherwise $\exists r \in R_s[p]$ such that $A(r) \subseteq +\Delta^{n-1}$. By inductive hypothesis $A(r) \subseteq +\Delta^n$. Therefore $p \in +\Delta^{n+1}$.

Case $-\Delta$. Trivial since $-\Delta^{n+1} = -\Delta^n \cup S$, for some set of literals S.

Case $+\partial$. If $p \in +\partial^n$ because $p \in +\Delta^{n-1}$, then by inductive hypothesis $p \in +\Delta^n$ and thus $p \in +\partial^{n+1}$. Otherwise $p \in +\partial^n$ if (i) $\exists r \in R_{sd}[p]$ such that $A(r) \subseteq +\partial^{n-1}$ and (ii) $\sim p \in -\Delta^{n-1}$ and $\forall s \in R[\sim p]$ either (iii) $A(s) \cap -\partial^{n-1} \neq \emptyset$ or (iv) $\exists t \in R[p]$ such that $A(t) \subseteq +\partial^{n-1}$ and $t > s$. By inductive hypothesis, if (i) then $A(r) \subseteq +\partial^n$, if (ii) then $\sim p \in -\Delta^n$, if (iii) then $A(s) \cap -\partial^n \neq \emptyset$, and if (iv) then $A(t) \subseteq +\partial^n$. Therefore every time the conditions for p being in $+\partial^n$ are satisfied, so are those for p being in $+\partial^{n+1}$.

Case $-\partial$. First of all, by definition, $-\partial^n \subseteq -\Delta^n$, then we have three possibilities to add a literal p to $-\partial^{n+1}$. (i) $\forall r \in R_{sd}[p] A(r) \cap -\partial^{n-1} \neq \emptyset$. by inductive hypothesis $-\partial^{n-1} \subseteq -\partial^n$, thus $A(r) \cap -\partial^n \neq \emptyset$, hence $p \in -\partial^{n+1}$. (ii) $\sim p \in +\Delta^{n-1}$, but by inductive hypothesis $+\Delta^{n-1} \subseteq +\Delta^n$, thus $p \in -\partial^{n+1}$. (iii) $\exists s \in R[\sim p]$ such that $A(s) \subseteq +\partial^{n-1}$ and $\forall t \in R[p]$ either $A(t) \cap -\partial^{n-1} \neq \emptyset$ or $t \not> s$. By inductive hypothesis $+\partial^{n-1} \subseteq +\partial^n$ and $-\partial^{n-1} \subseteq -\partial^n$. Therefore, also in this case, $p \in -\partial^{n+1}$. Hence $p \in -\partial^{n+1}$. □

The extension F captures exactly the inferences described in the proof theory.

Theorem 2.2
Let D be a finite propositional defeasible theory, q a literal and

$$F = (+\Delta_F, -\Delta_F, +\partial_F, -\partial_F)$$

is the limit of all finite elements of the Kleene sequence from $\perp$ via $\mathscr{T}_D$.

Then:

- $D \vdash +\Delta q$ iff $q \in +\Delta_F$
- $D \vdash -\Delta q$ iff $q \in -\Delta_F$
- $D \vdash +\partial q$ iff $q \in +\partial_F$
- $D \vdash -\partial q$ iff $q \in -\partial_F$

Proof
We prove the theorem by induction on the length of derivations in one direction and on the number of iterations of the operator $\mathscr{T}_D$ in the other.

Case $+\Delta$, *Inductive base* $\Rightarrow$. $P(1) = +\Delta q$. This means that either $q \in F$ or that there exists a rule $r \in R_s[q]$ such that $A(r) = \emptyset$. In both cases $q \in +\Delta_F^1$. In the first case by definition, in the second since $\emptyset \subseteq +\Delta^0 = \emptyset$. By the monotonicity of $\mathscr{T}_D$, $q \in +\Delta_F$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Inductive step. We assume that the theorem holds for proofs of length up to n, and $P(n+1) = +\Delta q$. Here we consider only the cases different from the inductive base. Thus there exists a rule $r \in R_s[q]$ such that $\forall a \in A(r)$, $+\Delta a \in P(1..n)$. By inductive hypothesis $a \in +\Delta_F$. Let $+\Delta^m$ be the minimal set of literals in the Kleene sequence defined by $\mathscr{T}_D$ containing all such as. Clearly $A(r) \subseteq \Delta^{m+1}$. Hence, by the monotonicity of $\mathscr{T}_D$, $q \in +\Delta_F$.

Inductive base $\Leftarrow$. If $q \in +\Delta^1$, then either $q \in F$ or $\exists r \in R_s[q] : A(r) \subseteq +\Delta^0$, that is $A(r) = \emptyset$. In both case we have that $+\Delta q$ is a single line proof of q.

Inductive step. We have that $q \in +\Delta^{n+1}$ and the property holds up to $+\Delta^n$. If $q \in +\Delta^{n+1}$ because $q \in F$, then, as in the previous case, $+\Delta q$ is a single line proof of q. Otherwise $\exists r \in R_s[q] : A(r) \subseteq +\Delta^n$. By inductive hypothesis $\forall a_i \in A(r)$, $D \vdash +\Delta a_i$. Let $a_1, \ldots, a_l$ be an enumeration of the literals in $A(r)$, and let $P(a_i)$ be a proof of a_i. We concatenate the $P(a_i)$s and we append $+\Delta q$ at the end. It is immediate to verify that the sequence thus obtained is a proof of $+\Delta q$.

Case $-\Delta$, *Inductive base* $\Rightarrow$. $P(1) = -\Delta q$ iff $p \notin F$ and $\neg\exists r \in R_s$ such that $C(r) = q$. On the other hand $-\Delta^0 = \emptyset$, so the set of literals satisfying $\forall s \in R_s[q] : A(s) \cap -\Delta^0 \neq \emptyset$ is the set of literals not appearing as the consequent in a strict rule in D. Moreover the definition of relative complement gives us

$$-\Delta^1 = \{p | \neg\exists r \in R_s \wedge p \notin F\}$$

Therefore $q \in -\Delta^1$, and by the monotonicity of $\mathscr{T}_D$, $q \in -\Delta_F$.

Inductive step. Let us assume that the property holds up to n and $P(n+1) = -\Delta p$. This implies 1) $p \notin F$ and 2) $\forall r \in R_s[p] \exists q \in A(r)$ such that $-\Delta q \in P(1..n)$. By inductive hypothesis, for some m, $q \in -\Delta^m$ and so $A(r) \cap -\Delta^m \neq \emptyset$, thus $p \in= \Delta^{m+1}$ and therefore by the monotonicity of $\mathscr{T}_D$ it is in $-\Delta_F$.

Inductive base $\Leftarrow$. As we have seen $-\Delta^1 = \{p | \neg\exists r \in R_s \wedge p \notin F\}$, thus, vacuously, we have a single line proof of $-\Delta q$.

Inductive step. Let us assume that the property holds up to n, and let us suppose that $q \notin -\Delta^n$, but $q \in -\Delta^{n+1}$. This implies that $q \notin F$ and $\forall r \in R_s[q] : A(r) \cap -\Delta^n \neq \emptyset$. This means that $\forall r \in R_s[q]$ $\exists a_r \in A(r)$ such that $D \vdash -\Delta a_r$. Let $P(a)$ be a derivation of a, and $P(a_1), \ldots, P(a_r)$ be the concatenation of the proofs of such as. We append $-\Delta q$ at the end and we obtain a proof of $-\Delta q$, thus $D \vdash -\Delta q$.

Cases $+\partial$ ***and*** $-\partial$. *Inductive base,* $\Rightarrow$. The tags $+\partial$ and $-\partial$, as well as the corresponding sets of literals, depend on each other, so we will carry out the proofs simultaneously. Moreover, since the tags $+\Delta$ and $-\Delta$ (and the sets of literal corresponding to them) are independent from $+\partial$ and $-\partial$ –and we have already proved the theorem for them– we assume, without any loss of generality, that derivations in defeasible logic consist only of defeasible tagged literals.

Case $P(1) = +\partial q$. This is possible if 1) $q \in +\Delta_F$, or 2) $\sim q \in -\Delta_F$ and $\exists r \in R_{sd}[q]$ such that $A(r) = \emptyset$ and there are no rules for $\sim p$. For 1) by the definition of $\mathscr{T}_D$ there exists an n such that $q \in +\Delta^n$, then $q \in +\partial^{n+1}$; by the monotonicity of

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

$\mathscr{T}_D\ q \in +\partial_F$. For 2) we have that $A(r)$ is included in every $+\partial^n$, and the condition beginning with $\forall r \in R[\sim q]$ is vacuously satisfied since $R[\sim q] = \emptyset$. Let $-\Delta^m$ be the minimal set in the Kleene sequence generated by $\mathscr{T}_D$ such that $\sim q \in -\Delta^m$. Then we can conclude that $q \in +\partial^{n+1}$; therefore $q \in +\partial_F$.

Case $P(1) = -\partial q$. First of all, let us consider the following decomposition of $-\partial^{n+1}$

$$-\partial^{n+1} = -\partial_{\cap}^{n+1} \cup -\partial_{\Delta}^{n+1} \cup -\partial_{>}^{n+1}$$

where

$$-\partial_{\cap}^{n+1} = \{p \in -\Delta^n | \forall r \in R_{sd}[p], A(r) \cap -\partial^n \neq \emptyset\}$$
$$-\partial_{\Delta}^{n+1} = \{p \in -\Delta^n | \sim p \in +\Delta^n\}$$
$$-\partial_{>}^{n+1} = \{p \in -\Delta^n | \forall r \in R_{sd}[p]\ \exists s \in R[\sim p] : A(s) \subseteq +\partial^n \text{ and } \forall t \in R_{sd}[p] :\ \text{either } A(t) \cap -\partial^n \neq \emptyset \text{ or } t \not> s\}$$

Now $P(1) = -\partial q$ is possible if $D \vdash -\Delta q$ and either

1. $D \vdash +\Delta q$ or
2. $R_{sd}[\sim q] = \emptyset$ or
3. $\exists s \in R[\sim q] : A(s) = \emptyset$ and $\neg \exists t \in R[q] : t > s$

From the previous cases we have $q \in -\Delta_F$ and for 1) $\sim q \in +\Delta_F$. Let n be the minimum number of iterations of $\mathscr{T}_D$ such that both $q \in -\Delta^n$ and $q \in +\Delta^n$; then by construction $q \in +\partial^{n+1}$. For 2) we obtain that $q \in -\partial_{\cap}^{n+1}$ and $q \in -\partial_{>}^{n+1}$ since the conditions are vacuously satisfied. Finally 3) implies that $A(s) \subseteq +\partial^n$ for any n and $\forall t \in R[q], t \not> s$; thus $q \in -\partial_{>}^{n+1}$ for some n such that $q \in +\Delta^n$.

Inductive step.

Case $P(n+1) = +\partial q$. Let us assume that the inductive hypothesis holds for derivations of length up to n. We only show the cases different from the inductive base. This means that we have cases corresponding to clause 2) of the proof conditions for $+\partial$. From the inductive hypothesis we have $A(r) \subseteq +\partial_F$ for some $r \in R_{sd}[q]$ (clause 2.2), $\forall s \in R[\sim q]$ either $A(s) \cap -\partial_F \neq \emptyset$ (clause 2.3.1) or $\exists t \in R_{sd}[q] : t > s$ and $A(t) \subseteq +\partial_F$ (clause 2.3.2). By the monotonicity of $\mathscr{T}_D$ we get that a minimum n such that $+\partial^n$ and $-\partial^n$ that satisfy the above condition exists. Therefore $q \in +\partial^{n+1}$ and consequently $q \in +\partial_F$.

Case $P(n+1) = -\partial q$. Here we have 1) $q \in -\Delta_F$, and either 2) $\sim q \in +\Delta_F$ or 3) the clause 2.1 of the proof condition for $-\partial$ is not vacuously satisfied: in such a case $\forall r \in R_{sd}[p]$:

2.1 $\exists p \in A(r)$ such that $-\partial q \in P(1..n)$; by inductive hypothesis $p \in -\partial_F$, so for some m, $A(r) \cap -\partial^n \neq \emptyset$.

2.3 $\exists s \in R[\sim q]$ such that

2.3.1 $\forall a_i \in A(s) + \partial a_i \in P(1..n)$; by inductive hypothesis each $q_i \in +\partial_F$, therefore for some m, $A(s) \subseteq +\partial^m$; or

2.3.2 $\forall t \in R[q]$ either $t \not> s$, or $\exists a \in A(t)$ such that $-\partial q \in P(1..n)$. By inductive hypothesis $a \in -\partial_F$, then, for some m $A(t) \cap -\partial^m \neq \emptyset$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Also in this case it is immediate to see that each condition has been reduced to the correspondent condition required for the construction of $-\partial^{m+1}$, therefore, for the smallest m satisfying the above three conditions we can conclude $q \in -\partial^{m+1}$, and, by the monotonicity of $\mathscr{T}_D$, $q \in -\partial_F$.

Inductive base, $\Leftarrow$.

Case $q \in +\partial^1$. If $q \in +\partial^1$ because it is in $+\Delta_F$, then we have that $D \vdash +\Delta q$. Let P be a proof of $+\Delta q$; we append $+\partial q$ at the end of P obtaining a proof of $+\partial q$.

For the other case we have

$$\exists r \in R_{sd}[q] : A(r) = \emptyset \text{ and } \forall s \in R[\sim q]\ \exists t \in R[q] : A(t) = \emptyset, t > s$$

In this case it is easy to verify that $+\partial q$ is a single line proof for q.

Case $q \in -\partial^1$. Here we have that $q \in -\Delta_F$ and the following three cases:

1. $\sim q \in +\Delta_F$
2. $R_{sd}[q] = \emptyset$
3. $\exists r \in R[\sim q] : A(r) = \emptyset$ and $\forall t \in R[q], t \not> r$

In the first case, we have already proved that $D \vdash +\Delta q$ and $D \vdash -\Delta q$. So let P the concatenation of the proofs of $+\Delta q$ and $-\Delta q$. We append $-\partial q$ at the end of P and we obtain a derivation of $-\partial q$. For 2) the sequence $-\Delta q, -\partial q$ is vacuously a derivation of $-\partial q$, and for 3) the concatenation of a proof of $-\Delta q$ and $-\partial q$ satisfies condition 3 of the definition of $-\partial$.

Inductive step. We assume that the theorem holds up to $+\partial^n$ and $-\partial^n$. Furthermore it is worth noting that the construction of derivation of defeasible literals in Defeasible Logic and the construction of the corresponding sets of literals are invertible, i.e., each time the top-down (procedural) condition requires something to be provable, the bottom-up (set-theoretic) condition satisfies an appropriate set-theoretic condition. But we have seen that, granted the inductive hypothesis the proof-theoretic condition and the set-theoretic one coincide.

Case $q \in +\partial^{n+1}$. The difference with the inductive base is that now we replace $A(r) = \emptyset$, $A(t) = \emptyset$ with $A(r) \subseteq +\partial^n$, $A(t) \subseteq +\partial^n$, and $A(s) \cap -\partial^n \neq \emptyset$. By the inductive hypothesis we have $\forall a_r \in A(r) : D \vdash +\partial a_r$, $\forall a_t \in A(t) : D \vdash +\partial a_t$, and $\exists a_s \in A(s) : D \vdash -\partial a_s$. Let P be a concatenation of the derivations of the just mentioned tagged literals. We append $+\partial q$ at the end of P and we obtain a derivation of $+\partial q$.

Case $q \in -\partial^{n+1}$. First of all we can use an argument similar to the previous case. Here the main difference with the inductive base is that we have to consider that the set of supportive rules for q is not empty and we have to consider the case where $\forall r \in R_{sd}[q], A(r) \cap -\partial^n \neq \emptyset$. By inductive hypothesis we have that $\forall r \in R_{sd}[q] \exists a_r \in A(r) : D \vdash -\partial a_r$. At this stage we can concatenate the derivations of such a_rs with the other required derivations and we append $-\partial q$ at the end. Again we can verify that the resulting sequence is a proof for $-\partial q$. □

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

2.5 Beyond propositional defeasible logic

The restriction of Theorem 2.2 to finite propositional theories derives from the formulation of the proof theory; proofs are guaranteed to be finite under this restriction. Defeasible Logic has a constructive proof theory that guarantees the explainability of conclusions (i.e., for every conclusion we derive, it is possible to provide a full proof with justification of the essential steps). In addition the proof theory defines the classical defeasible logic of Antoniou *et al.* (2001) and Billington *et al.* (1990).

On the other hand, the bottom-up semantics does not need this restriction, and so can be used in conjunction with predicate defeasible logic rules that represent infinitely many propositional rules. This also means that under this characterisation defeasible logic can be supplemented with function symbols. In the following we will take advantage of this opportunity and provide a more elegant and general characterisation under the Kunen semantics (Section 3.2 and 5.3), while in the remaining sections we stick to the original definition.

3 Semantics of logic programs

Now that we have presented defeasible logic and have developed the technical properties that will be needed later on, we turn to the theme of this paper: establish formal connections between defeasible logic and logic programming semantics. This section presents the basics of the semantics that will be used in subsequent sections

A *logic program* P is a finite set of program clauses. A *program clause* r has the form

$$A \leftarrow B_1, \ldots, B_n, not\ C_1, \ldots, not\ C_m$$

where $A, B_1, \ldots B_n, C_1, \ldots, C_m$ are positive literals. A program clause with variables is considered to represent the set of its ground instances.

In this paper we will make use of two well-known logic programming semantics: stable model semantics (Gelfond and Lifschitz 1988) and Kunen semantics (Kunen 1987). In the following we present them briefly for the sake of completeness.

3.1 Stable model semantics

Let M be a subset of the Herbrand base. We call a ground program clause

$$A \leftarrow B_1, \ldots, B_n, not\ C_1, \ldots, not\ C_m$$

irrelevant w.r.t. M if at least one C_i is included in M. Given a logic program P, $ground(P)$ is the set of ground instances of the logic program P, and we define the *reduct of* P *w.r.t.* M, denoted by P^M, to be the logic program obtained from $ground(P)$ by

1. removing all clauses that are irrelevant w.r.t. M, and
2. removing all premises *not* C_i from all remaining program clauses.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Note that the reduct P^M is a definite logic program, and we are no longer faced with the problem of assigning semantics to negation, but can use the least Herbrand model instead.

M is a *stable model* of P iff $M = \mathscr{M}_{P^M}$, where $\mathscr{M}_{P^M}$ is the least Herbrand model of the reduct of P.

3.2 Kunen semantics

Kunen semantics (Kunen 1987) is a 3-valued semantics for logic programs. A *partial interpretation* is a mapping from ground atoms to one of the three truth values **t**, **f** and **u**, which denote true, false and unknown, respectively. This mapping can be extended to arbitrary formulas using Kleene's 3-valued logic.

Kleene's truth tables can be summarized as follows. If φ is a boolean combination of atoms with truth value one of **t**, **f** and **u**, its truth value is **t** iff all possible ways of putting **t** or **f** for the various **u**-values lead to a value **t** being computed in ordinary (2-valued) logic; φ gets the value **f** iff $\neg\varphi$ gets the value **t**; and φ gets the value **u** otherwise. These truth values can be extended in the obvious way to predicate logic, thinking of the quantifiers as infinite conjunctions or disjunctions.

The Kunen semantics of a program P is obtained from a sequence $\{I_n\}$ of partial interpretations, defined as follows:

1. $I_0(a) = \mathbf{u}$ for every atom a.
2. $I_{n+1}(a) = \mathbf{t}$ iff there is a program clause

$$A \leftarrow B_1, \ldots, B_n, not\ C_1, \ldots, not\ C_m$$

and a ground substitution σ such that $a = A\sigma$ and that

$$I_n((B_1 \wedge \ldots \wedge B_n \wedge \neg C_1 \wedge \ldots \wedge \neg C_m)\sigma) = \mathbf{t}.$$

3. $I_{n+1}(a) = \mathbf{f}$ iff for all clauses

$$A \leftarrow B_1, \ldots, B_n, not\ C_1, \ldots, not\ C_m$$

in the program, and all ground substitutions σ, if $a = A\sigma$ then

$$I_n((B_1 \wedge \ldots \wedge B_n \wedge \neg C_1 \wedge \ldots \wedge \neg C_m)\sigma) = \mathbf{f}.$$

4. $I_{n+1}(a) = \mathbf{u}$ if neither 2. nor 3. applies.

We shall say that the Kunen semantics of P supports a formula φ, written $P \models_K \varphi$, iff there is an interpretation I_n, for some finite n, such that $I_n(\varphi) = \mathbf{t}$.

4 A translation of defeasible theories into logic programs

In this section we describe a meta-program $\mathscr{M}$ in a logic programming form that expresses the essence of the defeasible reasoning embedded in defeasible logic first introduced in Maher and Governatori (1999). $\mathscr{M}$ consists of the following clauses. We first introduce the predicates defining classes of rules, namely

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

```
supportive_rule(Name, Head, Body):-
    strict(Name, Head, Body).

supportive_rule(Name, Head, Body):-
    defeasible(Name, Head, Body).

rule(Name, Head, Body):-
    supportive_rule(Name, Head, Body).

rule(Name, Head, Body):-
    defeater(Name, Head, Body).
```

Next we introduce the clauses defining the predicates corresponding to $+\Delta$, $-\Delta$, $+\partial$, and $-\partial$. These clauses specify the structure of defeasible reasoning in defeasible logic. Arguably they convey the conceptual simplicity of defeasible logic more clearly than the proof theory.

c1
```
definitely(X):-
    fact(X).
```

c2
```
definitely(X):-
    strict(R, X, [Y1, ..., Yl]),
    definitely(Y1), ..., definitely(Yn).
```

c3
```
defeasibly(X):-
    definitely(X).
```

c4
```
defeasibly(X):-
    not definitely(~ X),
    supportive_rule(R, X, [Y1, ..., Yl]),
    defeasibly(Y1), ..., defeasibly(Yn),
    not overruled(R, X).
```

c5
```
overruled(R, X):-
    rule(S, ~ X, [U1, ..., Ul]),
    defeasibly(U1), ..., defeasibly(Un),
    not defeated(S, ~ X).
```

c6
```
defeated(S, ~ X):-
    sup(T, S),
    supportive_rule(T, X, [V1, ..., Vl]),
    defeasibly(V1), ..., defeasibly(Vn).
```

The first two clauses address definite provability, while the remainder address defeasible provability. The clauses specify if and how a rule can be overridden by another, and which rules can be used to defeat an overriding rule, among other aspects of the structure of defeasible reasoning in defeasible logic.

We have permitted ourselves some syntactic flexibility in presenting the meta-program. However, there is no technical difficulty in using conventional logic programming syntax to represent this program.

Finally, for a defeasible theory $D = (F, R, >)$, we add facts according to the following guidelines:

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

1. `fact`(p). for each $p \in F$
2. `strict`$(r_i, p, [q_1, \ldots, q_n])$. for each rule $r_i : q_1, \ldots, q_n \rightarrow p \in R$
3. `defeasible`$(r_i, p, [q_1, \ldots, q_n])$. for each rule $r_i : q_1, \ldots, q_n \Rightarrow p \in R$
4. `defeater`$(r_i, p, [q_1, \ldots, q_n])$. for each rule $r_i : q_1, \ldots, q_n \rightsquigarrow p \in R$
5. `sup`(r_i, r_j). for each pair of rules such that $r_i > r_j$

5 Properties of the translation

5.1 Embedding under stable model semantics

We establish relationships between D and its translation $\mathscr{M}$. To do so we must select appropriate logic program semantics to interpret *not* in clauses c4 and c5. First we consider stable model semantics.

The aim of this section is twofold. On one hand it established a relationship between Defeasible Logic and stable semantics. This connection is obtained via the representation of the meta-program for Defeasible Logic as a Default Theory. In this way we are able to use the well-know and well-understood link between stable semantics and Default Logic to simplify the proofs of our results and, at the same time, it opens the way to investigations on the similarities and differences over the two non-monotonic formalisms. Furthermore it will enable further studies on relationships between semantics for default logic and defeasible logic.

To this end we briefly rehearse the basic definitions of default logic.

5.1.1 Basics of default logic

A *default* δ has the form $\frac{\varphi : \psi_1, \ldots, \psi_n}{\chi}$ with closed formulas φ, $\psi_1, \ldots, \psi_n$, χ. φ is the *prerequisite* $pre(\delta)$, $\psi_1, \ldots, \psi_n$ the *justifications* $just(\delta)$, and χ the *consequent* $cons(\delta)$ of δ.

A *default theory* T is a pair (W, Def) consisting of a set of formulas W and a countable set Def of defaults.

Let $\delta = \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi}$ be a default, and E a deductively closed set of formulas. We say that δ *is applicable to* E iff $\varphi \in E$, and $\neg\psi_1, \ldots, \neg\psi_n \notin E$.

Let $\Pi = (\delta_0, \delta_1, \delta_2, \ldots)$ be a finite or infinite sequence of defaults from Def without multiple occurrences (modelling an application order of defaults from Def). We denote by $\Pi[k]$ the initial segment of Π of length k, provided the length of Π is at least k.

- $In(\Pi) = Th(W \cup \{cons(\delta) \mid \delta \text{ occurs in } \Pi\})$, where Th denotes the deductive closure.
- $Out(\Pi) = \{\neg\psi \mid \psi \in just(\delta), \delta \text{ occurs in } \Pi\}$.

Π is called a *process of* T iff δ_k is applicable to $In(\Pi[k])$, for every k such that δ_k occurs in Π. Π is *successful* iff $In(\Pi) \cap Out(\Pi) = \emptyset$, otherwise it is *failed*. Π is *closed* iff every default that is applicable to $In(\Pi)$ already occurs in Π.

Antoniou (1997) showed that Reiter's (1980) original definition of extensions is equivalent to the following one: A set of formulas E is an *extension* of a default theory T iff there is a closed and successful process Π of T such that $E = In(\Pi)$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

5.1.2 Stable models and default logic

The Default Logic interpretation of a ground program clause Cl

$$A \leftarrow B_1, \ldots, B_n, not\ C_1, \ldots, not\ C_m$$

is given by the default

$$df(Cl) = \frac{B_1 \wedge \ldots \wedge B_n : \neg\ C_1, \ldots, \neg\ C_m}{A}.$$

We define $df(P)$, the *default logic interpretation* of the logic program P, to be the default theory (W, D) with $W = \emptyset$ and $D = \{df(Cl) \mid Cl \in ground(P)\}$.

Theorem 5.1 (*Antoniou 1997*)
Let P be a logic program, and M a subset of the Herbrand base. M is a stable model of P iff $Th(M)$ is an extension of $df(P)$.

5.2 The meta-program as a default theory

According to the previous section, the meta-program of section 4 can be equivalently viewed as the following default theory $T(D) = (W(D), Def(D))$.

The predicate logic part $W(D)$ contains:

1. $\texttt{fact}(p)$ — for each $p \in F$
2. $\texttt{strict}(r_i, p, [q_1, \ldots, q_n])$. — for each rule $r_i : q_1, \ldots, q_n \rightarrow p \in R$
3. $\texttt{defeasible}(r_i, p, [q_1, \ldots, q_n])$. — for each rule $r_i : q_1, \ldots, q_n \Rightarrow p \in R$
4. $\texttt{defeater}(r_i, p, [q_1, \ldots, q_n])$. — for each rule $r_i : q_1, \ldots, q_n \rightsquigarrow p \in R$
5. $\texttt{sup}(r_i, r_j)$. — for each pair of rules such that $r_i > r_j$

And the following defaults (actually default schemas) are elements of $Def(D)$:

$$d_1 = \frac{\texttt{fact}(X):}{\texttt{definitely}(X)}$$

$$d_2 = \frac{\texttt{strict}(R,X,[Y_1,\ldots,Y_n]) \wedge \texttt{definitely}(Y_1) \wedge \ldots \wedge \texttt{definitely}(Y_n):}{\texttt{definitely}(X)}$$

$$d_3 = \frac{\texttt{definitely}(X):}{\texttt{defeasibly}(X)}$$

$$d_4 = \frac{\texttt{supportive_rule}(R,X,[Y_1,\ldots,Y_n]) \wedge \texttt{defeasibly}(Y_1) \wedge \ldots \wedge \texttt{defeasibly}(Y_n): \quad \neg\,\texttt{definitely}(\sim X), \neg\,\texttt{overruled}(R,X)}{\texttt{defeasibly}(X)}$$

$$d_5 = \frac{\texttt{rule}(S,\sim X,[U_1,\ldots,U_n]) \wedge \texttt{defeasibly}(U_1) \wedge \ldots \wedge \texttt{defeasibly}(U_n) : \neg\,\texttt{defeated}(S,\sim X)}{\texttt{overruled}(R,X)}$$

$$d_6 = \frac{\texttt{sup}(T,S) \wedge \texttt{supportive_rule}(T,X,[V_1,\ldots,V_n]) \wedge \texttt{defeasibly}(V_1) \wedge \ldots \wedge \texttt{defeasibly}(V_n):}{\texttt{defeated}(S,X)}$$

$$d_7 = \frac{\texttt{strict}(Name,Head,Body):}{\texttt{supportive_rule}(Name,Head,Body)}$$

$$d_8 = \frac{\texttt{defeasible}(Name,Head,Body):}{\texttt{supportive_rule}(Name,Head,Body)}$$

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

$$d_9 = \frac{\texttt{supportive_rule}(Name,Head,Body):}{\texttt{rule}(Name,Head,Body)}$$

$$d_{10} = \frac{\texttt{defeater}(Name,Head,Body):}{\texttt{rule}(Name,Head,Body)}$$

Now let us prove a technical result on this default theory. It provides a condition on D under which $T(D)$ has at least one extension.

Lemma 5.1
Let D be a decisive defeasible theory. Then $T(D)$ has at least one extension.

Proof
If the atom dependency graph of D is acyclic we can define an arbitrary total order $\gg$ on atoms which respects the dependency graph. We proceed to construct a closed and successful process Π as follows:

1. First apply instantiations of d_1 and d_2 in any order. These defaults have no justification, so success cannot be jeopardized.
2. Then proceed to prove $\texttt{defeasibly}(p)$ or $\texttt{defeasibly}(\neg p)$ using the remaining defaults, in the order of $\gg$.
3. For each atom p, try to apply first defaults d_3, then d_6, then d_5 and finally d_4. If the defaults can be applied in this order only, the process Π cannot fail.
4. Π is closed when we have carried out step 3 for all atoms p.

The question is whether the order specified in 3 can always be respected. This is the case because Y_i, U_i and V_i use atoms appearing before X in the total order.

The argument in more detail: we analyze the situation where an instantiation of d_4 with p has impact on an earlier application of d_5 with instantiation $\sim q$. Suppose $\texttt{defeasibly}(p)$ is derived using default d_4 at stage k, and suppose it is then used to prove $\texttt{overruled}(r,q)$ later in the process Π, where $\texttt{defeasibly}(q)$ was derived in Π at a stage $l < k$. But then p occurs in the body of a rule with head predicate q, so q depends on p, so it must appear *after* p in $\gg$. So, according to 2 above, d_4 with instantiation q cannot have been applied before d_4 with instantiation p, so we have a contradiction.

A similar argument applies to the interplay between defaults d_5 and d_6. □

In general, $T(D)$ may not have any extension if the condition of the Lemma is not satisfied. For example consider D consisting of the rules:

$r_1 : \Rightarrow p$
$r_2 : p \Rightarrow q$
$r_3 : q \Rightarrow \neg p$

Let us now analyze the application of defaults in $T(D)$.

Because of r_1 we derive immediately $\texttt{overruled}(r_3)$ using d_5 with instantiation r_3 for R and r_1 for S. r_1 cannot be defeated using d_6 because there is no stronger rule. No interaction with other defaults can prevent this application of d_5, so it can appear at the beginning of any process, without loss of generality.

Then we can apply d_4 to derive $\texttt{defeasibly}(p)$, assuming $\neg\, \texttt{overruled}(r_1, p)$ (*).

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Now that $\texttt{defeasibly}(p)$ is derived, we can apply d_4 to derive $\texttt{defeasibly}(q)$, assuming $\neg\,\texttt{overruled}(r_2, q)$.

We apply d_5 with instantiation r_3 for S and r_1 for R to derive $\texttt{overruled}(r_1, p)$. This contradicts the previous assumption (*), so the process is failed.

There is no way any defaults along the process above can be blocked by applying another default instead. So there can be no extension.

We are now able to prove the main results, namely the relationships between Defeasible Logic and the stable semantics interpretation of the meta-program describing provability in Defeasible Logic.

Theorem 5.2

(a) If $D \vdash +\Delta p$ then $\texttt{definitely}(p)$ is included in all stable models of $\mathscr{M}$.

(b) If $D \vdash -\Delta p$ then $\texttt{definitely}(p)$ is not included in any stable model of $\mathscr{M}$.

(c) If D is decisive then the implications (a) and (b) are also true in the opposite direction.

Proof

(a): Proof by induction on the length of derivations P in D. Let the claim hold for $P(1..i)$, and let $P(i+1) = +\Delta p$. Let E be an extension of $T(D)$, and $E = In(\Pi)$ for a closed and successful process Π of $T(D)$.

Case 1: $p \in F$. Then $\texttt{fact}(p) \in W(D)$. Since d_1 is applicable to $In(\Pi)$ and Π is closed, we conclude $\texttt{definitely}(p) \in In(\Pi) = E$.

Case 2: There is $r \in R_s[p]$ such that $+\Delta a \in P(1..i)$ for all $a \in A(r)$. Since $r \in R_s[p]$ we have $\texttt{strict}(r, p, [q_1, \ldots, q_n]) \in Def(D)$. Since $+\Delta q_j \in P(1..i)$, for all $j = 1, \ldots, n$, we conclude with Induction Hypothesis that $\texttt{definitely}(q_j) \in E = In(\Pi)$. Thus d_2 is applicable to $In(\Pi)$. Π is closed so $\texttt{definitely}(p) \in In(\Pi) = E$.

(b): The proof goes by induction on the length of derivations P in D. Let the claim hold for $P(1..i)$ and let $P(i+1) = -\Delta p$. Further let E be an extension of $T(D)$. Then $E = In(\Pi)$ for a closed and successful process Π of $T(D)$.

By the inference condition $(-\Delta)$ we know $p \notin F$, thus $\texttt{fact}(p) \notin W(D)$. (*)

Also we know $\forall r \in R_s[p] \exists a \in A(r) : -\Delta a \in P(1..i)$. By induction Hypothesis we conclude

$$\forall r \in R_s[p] \exists a \in A(r) : \texttt{definitely}(a) \notin In(\Pi) \qquad (**)$$

(*) and (**) show that neither d_1 nor d_2 can be used to derive $\texttt{definitely}(p)$ in Π. But these are, by construction of $T(D)$, the only possibilities. Thus $\texttt{definitely}(p) \notin In(\Pi) = E$.

(c): Let $\texttt{definitely}(p) \in E$ for an extension E of $T(D)$. Such an extension exists because D is decisive. Then, by part (b) we conclude $D \nvdash -\Delta p$. Therefore $D \vdash +\Delta p$ because D is decisive. Thus the opposite of (a) holds. The opposite of (b) is shown in an analogous way.

Consider $D = \{ \Rightarrow p,\ p \Rightarrow q,\ q \Rightarrow \neg p\}$. As we have shown in section A.3, $T(D)$ has no extension, so $\texttt{definitely}(p)$ is included in all extensions of $T(D)$. However $D \nvdash +\Delta p$. Thus the opposite of (a) does not hold, in general.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Consider the theory D consisting only of the strict rule $p \rightarrow p$. $T(D)$ has only one extension, $E = Th(\emptyset)$, and $\mathtt{definitely}(p) \notin E$. However $D \nvdash -\Delta p$. This shows that the opposite of (b) is not true, in general. □

Theorem 5.3

(a) If $D \vdash +\partial p$ then $\mathtt{defeasibly}(p)$ is included in all stable models of $\mathscr{M}$.
(b) If $D \vdash -\partial p$ then $\mathtt{defeasibly}(p)$ is not included in any stable model of $\mathscr{M}$.
(c) If D is decisive then the implications (a) and (b) are also true in the opposite direction.

Proof

(c): Let $\mathtt{defeasibly}(p) \in E$ for an extension E of $T(D)$. Such an extension exists because D is decisive. Then, by part (b) we conclude $D \nvdash -\partial p$. Therefore $D \vdash +\partial p$ because D is decisive. Thus the opposite of (a) holds. The opposite of (b) is shown in an analogous way.

Let D consist of the rules: $p \Rightarrow p$, $p \Rightarrow q$ and $\Rightarrow \neg q$. Then $\mathtt{defeasibly}(\neg q)$ is included in the only extension of $T(D)$ but $D \nvdash \neg q$. This shows that the opposite of (a) is not necessarily true if D is not decisive.

A counterexample for the opposite direction of (b) is the defeasible theory consisting only of the rule $p \Rightarrow p$. $\mathtt{defeasibly}(p)$ is not included in the only extension of $T(D)$, however $-\partial p$ cannot be derived from D.

Parts (a) and (b) are shown concurrently by induction on the length of a derivation P in D. Suppose (a) and (b) hold for $P(1..i)$ (Induction Hypothesis). Consider an extension E of $T(D)$, and let $E = In(\Pi)$ for a closed and successful process Π of $T(D)$.

Case $+\partial$: $P(i+1) = +\partial p$. By the inference condition $(+\partial)$ there are two cases. The first case is that $+\Delta p \in P(1..i)$. By Theorem 5.2, $\mathtt{definitely}(p) \in E = In(\Pi)$. Then the default d_3 (with instantiation p for X) is applicable to $In(\Pi)$. Since Π is closed we conclude $\mathtt{defeasibly}(p) \in In(\Pi) = E$.

The second case is as follows:

(1) $\exists r \in R_{sd}[q]\ \forall a \in A(r) : +\partial a \in P(1..i)$ and
(2) $-\Delta{\sim}q \in P(1..i)$ and
(3) $\forall s \in R[{\sim}q]$ either
  (3.1) $\exists a \in A(s) : -\partial a \in P(1..i)$ or
  (3.2) $\exists t \in R_{sd}[q]$ such that
    $\forall a \in A(t) : +\partial a \in P(1..i)$ and $t > s$

From (2) we conclude

$$\mathtt{definitely}({\sim}p) \notin In(\Pi) = E \qquad (*)$$

using Theorem A.2. From (1) we get

$$\mathtt{supportive_rule}(r, p, [q_1, \ldots, q_n]) \in W(D) \qquad (**)$$

From (1) and Induction Hypothesis we get

$$\mathtt{defeasibly}(q_i) \in In(\Pi), \text{ for all } i = 1, \ldots, n \qquad (***)$$

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

In the following we show that $\texttt{overruled}(r, p) \notin In(\Pi)$. Together with (*)-(* * *) it shows that the default d_4 (with instantiation p for X) is applicable to $In(\Pi)$. Since Π is closed we get $\texttt{defeasibly}(p) \in In(\Pi) = E$.

Consider $s \in R[\sim p]$. In case (3.1) holds we have $-\partial a \in P(1..i)$ for an $a \in A(s)$. By Induction Hypothesis we conclude $\texttt{defeasibly}(a) \notin In(\Pi) = E$. Thus default d_5 cannot be applied with s instantiated for S.

In case (3.2) holds, it is easily seen that default d_6 can be used for the derivation of $\texttt{defeated}(s, \sim p)$. Thus $\texttt{defeated}(s, \sim p) \in In(\Pi)$, and d_5 cannot be applied with s instantiated for S.

Overall we have shown that d_5 fails to derive $\texttt{overruled}(r, p)$.

Case $-\partial$: Let $P(i+1) = -\partial p$. From the $(-\partial)$ inference condition we know $-\Delta \in P(1..i)$. Therefore, by Theorem A.2, $\texttt{definitely}(p) \notin E = In(\Pi)$. Thus default d_3 cannot be used to derive $\texttt{defeasibly}(p)$ in Π.

Next we show that d_4 cannot be used, either, to derive $\texttt{defeasibly}(p)$ in Π. Then $\texttt{defeasibly}(p) \notin In(\Pi) = E$, and we are finished. By the $(-\partial)$ inference condition there are three cases.

Case 1: $\forall r \in R_{sd}[p] \exists a \in A(r) : -\partial a \in P(1..i)$. By Induction Hypothesis we conclude that for every strict or defeasible rule with head p there is at least one antecedent a such that $\texttt{defeasibly}(a) \notin In(\Pi)$. Therefore, for every possible instantiation of R in default d_4 the prerequisite of d_4 is not in $In(\Pi)$. Thus d_4 cannot be used to derive $\texttt{defeasibly}(p)$ in Π.

Case 2: $+\Delta \sim p \in P(1..i)$. Then no instantiation of d_4 where the consequent is $\texttt{defeasibly}(p)$ can be applied since $\sim p \in In(\Pi) = E$, by Theorem A.2 (and because Π is successful).

Case 3: There is $s \in R[\sim p]$ such that:

(1) $\forall a \in A(s) + \partial a \in P(1..i)$ and
(2) $\forall t \in R_{sd}[p] : t \not> s$ or $\exists a \in A(t) : -\partial a \in P(1..i)$

From (1) together with Induction Hypothesis we get:

$$\texttt{rule}(s, \sim p, [u_1, \ldots, u_n]) \wedge \texttt{defeasibly}(u_1) \wedge \ldots \wedge \texttt{defeasibly}(u_n) \in In(\Pi) \quad (*)$$

Let $t \in R_{sd}[p]$.

Case 3.1: $t \not> s$. Then $\texttt{sup}(t, s) \notin W(D)$. So, d_6 with instantiation t for T cannot be used to derive $\texttt{defeated}(s, \sim p)$ in Π.

Case 3.2: $\exists a \in A(t) : -\partial a \in P(1..i)$. Then, by inductive hypothesis, we have $\texttt{defeasibly}(a) \notin In(\Pi)$. Again, d_6 with instantiation t for T cannot be used to derive $\texttt{defeated}(s, \sim p)$ in Π.

Overall we have shown:

$$\texttt{defeated}(s, \sim p) \notin In(\Pi) \quad (**)$$

From (*) and (**) we get that d_5 with instantiation s for S and r for R can be applied to $In(\Pi)$. Since Π is closed, we conclude $\texttt{overruled}(r, p) \in In(\Pi)$. Since r was chosen arbitrarily, the default d_4 cannot be used to prove $\texttt{defeasibly}(p)$ in Π, thus $\texttt{defeasibly}(p) \notin E = In(\Pi)$. □

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

The above two theorems show that if D is decisive, then the stable model semantics of $\mathscr{M}$ corresponds to the provability in defeasible logic. However part (c) is not true in the general case, as the following example shows.

Example 5.1
Consider the defeasible theory

$r_1 : \Rightarrow \neg p$
$r_2 : p \Rightarrow p$

In defeasible logic, $+\partial\neg p$ *cannot be proven because we cannot derive* $-\partial p$. *However,* $\texttt{defeasibly}(\neg p)$ *is a sceptical conclusion of* $\mathscr{M}$ *under stable model semantics because it is included in the only stable model of* $\mathscr{M}$. □

5.3 Embedding under Kunen semantics

If we wish to have an equivalence result without the condition of decisiveness, then we must use a different logic programming semantics, namely Kunen semantics.

The domain of our interpretation is given by the set of the rule-names and the set of literals occurring in a defeasible theory D.

- $I(\alpha) = \mathbf{t}$ iff
  1. $\alpha = \texttt{fact}(p)$ and $p \in F$;
  2. $\alpha = \texttt{strict}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \rightarrow p \in R$;
  3. $\alpha = \texttt{defeasible}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \Rightarrow p \in R$;
  4. $\alpha = \texttt{defeater}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \rightsquigarrow p \in R$;
  5. $\alpha = \texttt{sup}(r_i, r_j)$ and $\langle r_i, r_j \rangle \in >$.
- $I(\alpha) = \mathbf{f}$ iff
  1. $\alpha = \texttt{fact}(p)$ and $p \notin F$;
  2. $\alpha = \texttt{strict}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \rightarrow p \notin R$;
  3. $\alpha = \texttt{defeasible}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \Rightarrow p \notin R$;
  4. $\alpha = \texttt{defeater}(r_i, p, [q_1, \ldots, q_n])$ and $r_i : q_1, \ldots, q_n \rightsquigarrow p \notin R$;
  5. $\alpha = \texttt{sup}(r_i, r_j)$ and $\langle r_i, r_j \rangle \notin >$.
- $I(\alpha) = \mathbf{u}$ otherwise.

The intuition behind this interpretation is that the predicates correspond to the elements of D.

It is immediate to see that

$$I(\texttt{supportive_rule}(r_i, p, [q_1, \ldots, q_l])) = \begin{cases} \mathbf{t} & r_i \in R_{sd}[p] \text{ and } q_1, \ldots, q_l = A(r_i) \\ \mathbf{f} & \text{otherwise} \end{cases}$$

Similarly

$$I(\texttt{rule}(r_i, p, [q_1, \ldots, q_l])) = \begin{cases} \mathbf{t} & r \in R[p] \text{ and } q_1, \ldots, q_l = A(r_i) \\ \mathbf{f} & \text{otherwise} \end{cases}$$

Theorem 5.4
(a) $D \vdash +\Delta p \Leftrightarrow \mathscr{M} \models_K \texttt{definitely}(p)$.
(b) $D \vdash -\Delta p \Leftrightarrow \mathscr{M} \models_K \neg\texttt{definitely}(p)$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

(c) $D \vdash +\partial p \Leftrightarrow \mathscr{M} \models_K \texttt{defeasibly}(p)$.
(b) $D \vdash -\partial p \Leftrightarrow \mathscr{M} \models_K \neg\,\texttt{defeasibly}(p)$.

Proof

Case 1, $\Rightarrow$. We prove it by induction on the construction of $+\Delta$.

Inductive base.
$p \in +\Delta^1$ iff $p \in F$ or $\rightarrow p \in R$, iff $I(\texttt{fact}(p)) = \mathbf{t}$ or $I(\texttt{strict}(r, p, [\,])) = \mathbf{t}$, so either one of the ground instance $\texttt{definitely}(p)$:- $\texttt{fact}(p)$ of the clause c1 or $\texttt{definitely}(p)$:- $\texttt{strict}(r, p, [\,])$ of the clause c2 implies $I_1(\texttt{definitely}(p)) = \mathbf{t}$.

Inductive step. Let us assume that the property holds up to n, and $p \in +\Delta^{n+1}$. This means that either

1. $p \in F$, for which we can repeat the same argument as the inductive base; or
2. $\exists r \in R_s[p]$, say r_i such that $A(r_i) \subseteq +\Delta^n$. This implies that, for some m

$$I_m(\texttt{strict}(r_i, p, [q_1, \ldots, q_l])) = \mathbf{t},$$

and, by inductive hypothesis, for each q_j $(1 \leqslant j \leqslant l)$

$$I_m(\texttt{definitely}(q_j)) = \mathbf{t}.$$

Thus, by clause c2, $I_{m+1}(\texttt{definitely}(p)) = \mathbf{t}$.

Case 1, $\Leftarrow$. We use induction on the steps on which $\texttt{definitely}(p)$ is supported.

Inductive base. In this case we have

$$I_1(\texttt{definitely}(p)) = \mathbf{t} \text{ iff } \begin{cases} \texttt{definitely}(p) \text{ :- } \texttt{fact}(p) \text{ and } I(\texttt{fact}(p)) = \mathbf{t} \\ \texttt{definitely}(p) \text{ :- } \texttt{strict}(r, p, [\,]) \text{ and } \\ \qquad\qquad\qquad\qquad I(\texttt{strict}(r, p, [\,])) = \mathbf{t} \end{cases}$$

The first case amounts to $p \in F$, but $F \subseteq \Delta^n$, for any n, therefore $p \in +\Delta_F$. In the second case we have a rule r such that $A(r) = \emptyset$ and so $A(r) \subseteq +\Delta^n$, for any n, hence $p \in +\Delta_F$.

Inductive step. Let us assume that the property holds up to n, and

$$I_{n+1}(\texttt{definitely}(p)) = \mathbf{t}.$$

This is possible if there is either a ground instance of clause c1 $\texttt{definitely}(p)$:- $\texttt{fact}(p)$, for which we repeat the argument of the inductive base, or a ground instance

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

of clause c2 such that

$$I_n(\texttt{strict}(r, p, [q_1, \ldots, q_l])) = \mathbf{t},$$

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

which implies $r \in R_s[p]$, and, for each q_i,

$$I_n(\texttt{definitely}(q_i)) = \mathbf{t}.$$

By inductive hypothesis, for some m, $q_i \in +\Delta^m$, so $A(r) \subseteq +\Delta^m$; therefore $p \in +\Delta^{m+1}$; hence $p \in +\Delta_F$.

Case 2, $\Rightarrow$. Also in this case we use induction on the construction of $-\Delta$.

Inductive base. If $p \in -\Delta$ then

1. $p \notin F$, but $p \notin F$ iff $I(\texttt{fact}(p)) = \mathbf{f}$; this implies that all instances of c1 fails; or
2. $\forall r \in R_s[p](A(r) \cap -\Delta^0 \neq \emptyset)$. Given $-\Delta^0 = \emptyset$, the sentence $A(r) \cap -\Delta^0 \neq \emptyset$ is always false, therefore the condition can be satisfied only vacuously. This means $R_s[p] = \emptyset$, which implies

$$I(\texttt{strict}(r, p, [q_1, \ldots, q_l])) = \mathbf{f},$$

from which it follows that all ground instances of c2 fail.

Since $\texttt{definitely}(p)$ fails in all the cases we conclude $I_1(\texttt{definitely}(p)) = \mathbf{f}$, and $I_1(\texttt{not definitely}(p)) = \mathbf{t}$.

Inductive step. Let us assume that the property holds up to n, $p \notin -\Delta^n$, and $p \in -\Delta^{n+1}$. This implies that $p \notin F$, so we can repeat the above argument for the first clause. If $\forall r \in R_s[p](A(r) \cap \Delta^n \neq \emptyset)$ is satisfied, we have two cases: if it is vacuously we use the argument of the inductive base, otherwise, for each rule r, $A(r) \cap -\Delta^n = q_i$. From this we know that $q_i \in -\Delta^n$, and we can apply the inductive hypothesis; thus for some m, $I_m(\texttt{not definitely}(q_i)) = \mathbf{t}$, then $I_m(\texttt{definitely}(q_i)) = \mathbf{f}$. This happens for each instance of clause c2, so $I_{m+1}(\texttt{definitely}(p)) = \mathbf{f}$, therefore $I_{n+1}(\texttt{not definitely}(p)) = \mathbf{t}$.

Case 2, $\Leftarrow$. The proof is by induction on the step on which $\texttt{not strictly}(p)$ is supported.

Inductive base: $n = 1$.

$$I_1(\texttt{not definitely}(p)) = \mathbf{t} \iff I_1(\texttt{definitely}(p)) = \mathbf{f}$$

$$\iff \begin{cases} 1)\ I(\texttt{fact}(p)) = \mathbf{f}, \text{ and} \\ 2)\ \forall q, r,\ I(\texttt{strict}(r, p, [q_1, \ldots, q_l]), \\ \qquad\qquad \texttt{definitely}(q_1), \ldots, \\ \qquad\qquad \texttt{definitely}(q_l)) = \mathbf{f} \end{cases}$$

1) implies $p \notin F$; for 2), since `definitely` is not a primitive predicate

$$I(\texttt{definitely}(p)) = \mathbf{u}$$

for every p, thus

$$I(\texttt{strict}(r, p, [q_1, \ldots, q_l])) = \mathbf{f}$$

for every q, r; this implies $R_s[p] = \emptyset$, therefore $p \in -\Delta^1$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Inductive step. Let us assume that it holds up to n. We can repeat the above reasoning, but we have to consider also the case where for each rule $r_i \in R_s[p]$ and some $q_j \in A(r)$, $I_n(\texttt{definitely}(q_j)) = \mathbf{f}$. So $I_n(\texttt{not definitely}(q_j)) = \mathbf{t}$; by inductive hypothesis, $q_j \in -\Delta_F$, then $A(r) \cap -\Delta_F \neq \emptyset$. Let $-\Delta^m$ be smallest set satisfying this property. By construction we have $p \in -\Delta^{m+1}$, hence $p \in -\Delta_F$.

Cases 3 and 4 ($\Rightarrow$): Inductive base. The inductive base is trivial since it is easy to verify that both $+\partial^1$ and $-\partial^1$ are empty.

Inductive step. We assume that the theorem holds up to $+\partial^n$ and $-\partial^n$.

Case $p \in +\partial^{n+1}$. If $p \in +\partial^{n+1}$ because $p \in +\Delta^n$, then we can use the following ground instance of clause c1

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

By the inductive hypothesis $I_m(\texttt{definitely}(p)) = \mathbf{t}$, for some appropriate m, hence $I_{m+1}(\texttt{defeasibly}(p)) = \mathbf{t}$.

Otherwise, there is a rule $r \in R_{sd}[p]$ such that (1) $A(r) \subseteq +\partial^n$ and $\sim p \in -\Delta^n$, and (2) for every rule $s \in R[\sim p]$ either (a) $A(s) \cap -\partial^n \neq \emptyset$ or (b) there is rule $t \in R_{sd}[p]$ such that $A(t) \subseteq +\partial^n$ and $t > s$.

Let us consider an appropriate ground instance of clause c4:

$$\begin{aligned}&\texttt{defeasibly}(p) \texttt{ :- supportive_rule}(r, p, [q_1, \ldots, q_l]),\\ &\quad \texttt{not defeasibly}(\sim p),\\ &\quad \texttt{defeasibly}(q_1), \ldots, \texttt{defeasibly}(q_l),\\ &\quad \texttt{not overruled}(r, p).\end{aligned}$$

By construction $I(\texttt{supportive_rule}(r, p, [q_1, \ldots, q_l])) = \mathbf{t}$, and by the inductive hypothesis and (1), for some appropriate m, $I_m(\texttt{not definitely}(p)) = \mathbf{t}$ (from $\sim p \in -\Delta^n$), $I_m(\texttt{defeasibly}(q_i)) = \mathbf{t}$ for all $q_i \in A(r)$ since $A(r) \subseteq +\partial^n$, Therefore we have to show that

$$I_m(\texttt{not overruled}(r, p)) = \mathbf{t}.$$

This is equivalent to $I_m(\texttt{overruled}(r, p)) = \mathbf{f}$, that means that all the bodies of the appropriate ground instances of clause c5 are false in I_m. By construction for all rules $r \in R[\sim p]$, $I(\texttt{rule}(s, \sim p, [q_1^s, \ldots, q_l^s])) = \mathbf{t}$. Thus we have to prove that either $I_m(\texttt{defeasibly}(q_i^s)) = \mathbf{f}$ or $I_m(\texttt{not defeated}(s, \sim p)) = \mathbf{f}$.

Let us partition $R[\sim p]$ in two sets $S' = \{s \in R[\sim p] : A(s) \cap -\partial^n = \emptyset\}$ and $S'' = R[\sim p] - S'$. By (2a) and the inductive hypothesis we have that for every $s \in S'$

$$I_m(\texttt{defeasibly}(q_i^s)) = \mathbf{f}$$

for some $q_i^s \in A(s)$. For the rules in S'', on the other hand, we have to prove that $I_m(\texttt{not defeated}(s, \sim p)) = \mathbf{f}$.

$$I_m(\texttt{not defeated}(s, \sim p)) = \mathbf{f} \text{ iff } I_m(\texttt{defeated}(s, \sim p)) = \mathbf{t}.$$

Hence we need a ground instance of clause c6 evaluated as true in I_m. By (2b) for every rule in S'' there is a rule $t \in R_{sd}[p]$ such that $A(t) \subseteq +\partial^n$ and $t > s$. By

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

construction we obtain

$$I(\texttt{supportive_rule}(t, p, [q_1^t, \ldots, q_l^t])) = \mathbf{t} \text{ and } I(\texttt{supportive_rule}(t, s)) = \mathbf{t},$$

and by the inductive hypothesis, for all $q_i^t \in A(t)$, $I_m(\texttt{defeasibly}(q_i^t)) = \mathbf{t}$.

Case $p \in -\partial^{n+1}$. We have to show that not defeasibly(p) is supported by the Kunen semantics. This means that for some m, $I_m(\texttt{not defeasibly}(p)) = \mathbf{t}$, which implies $I_m(\texttt{defeasibly}(p)) = \mathbf{f}$. Accordingly we have to show that for each instance of clauses c3 and c4 an element of the body is not supported by the semantics.

Let us examine clause c3:

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

But $-\partial^{n+1} \subseteq -\Delta^n$, therefore $p \in -\Delta^n$. By the inductive hypothesis, for some m, $I_m(\texttt{not definitely}(p)) = \mathbf{t}$ and $I_m(\texttt{definitely}(p)) = \mathbf{f}$.

For clause c4

$$\begin{aligned}&\texttt{defeasibly}(p) \text{ :-}\\ &\quad \texttt{supportive_rule}(r, p, [q_1, \ldots, q_l]),\\ &\quad \texttt{not definitely}(\sim p),\\ &\quad \texttt{defeasibly}(q_1), \ldots, \texttt{defeasibly}(q_l),\\ &\quad \texttt{not overruled}(r, p).\end{aligned}$$

Let us examine the cases that imply that $p \in -\partial^{n+1}$.

If $\sim p \in +\Delta^n$, then, by the inductive hypothesis $I_m(\texttt{definitely}(\sim p)) = \mathbf{t}$ and $I_m(\texttt{not definitely}(\sim p)) = \mathbf{f}$.

From the basic interpretation it is obvious that we have to consider only the rules for p, since $I(\texttt{supportive_rule}(r, p, [q_1, \ldots, q_l])) = \mathbf{t}$ only if $r \in R_{sd}[p]$. Let us partition $R_{sd}[p]$ as follows: $R' = \{r \in R_{sd} : A(r) \cap -\partial^n \neq \emptyset\}$, and $R'' = R - R'$.

If $r \in R'$, then by the inductive hypothesis, for some $q_i \in A(r)$,

$$I_m(\texttt{not defeasibly}(q_i)) = \mathbf{t} \text{ and } I_m(\texttt{defeasibly}(q_i)) = \mathbf{f}.$$

If $r \in R''$, then we have to show that, for some m, $I_m(\texttt{not overruled}(r, p)) = \mathbf{f}$. This means that $I_m(\texttt{overruled}(r, p)) = \mathbf{t}$.

Let us examine again a relevant instance of clause c5:

$$\begin{aligned}&\texttt{overruled}(r, p) \text{ :-}\\ &\quad \texttt{rule}(s, \sim p, [q_1^s, \ldots, q_l^s]),\\ &\quad \texttt{defeasibly}(q_1^s), \ldots, \texttt{defeasibly}(q_l^s),\\ &\quad \texttt{not defeated}(s, \sim p).\end{aligned}$$

Since $p \in -\partial^{n+1}$, there is a rule s such that $s \in R[\sim p]$ and $A(s) \subseteq +\partial^n$. By construction $I(\texttt{rule}(s, \sim p, [q_1^s, \ldots, q_l^s])) = \mathbf{t}$, and by the inductive hypothesis, for every $q_i^s \in A(s)$:

$$I_m(\texttt{defeasibly}(q_i^s)) = \mathbf{t}.$$

Hence we have to show that $I(\texttt{not defeated}(s, \sim p)) = \mathbf{t}$, i.e. $I(\texttt{defeated}(s, \sim p)) = \mathbf{f}$, from which we know that all appropriate substitutions of clause c6 fail.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

$\texttt{defeated}(s, \sim p)$:-
$\texttt{supportive_rule}(t, p, [q_1^t, \ldots, q_l^t])$,
$\texttt{defeasibly}(q_1^t), \ldots, \texttt{defeasibly}(q_l^t)$,
$\texttt{sup}(t, s)$.

As we have seen $R[p] \neq \emptyset$, thus we consider two cases: i) $t \in R'$, and ii) $t \in R''$.

If $t \in R'$, then, by the inductive hypothesis, there exists a $q_i^t \in A(t)$ such that $I_m(\texttt{defeasibly}(q_i^t)) = \mathbf{f}$.

If $t \in R''$, since $p \in -\partial^{n+1}$, we have that $t \not> s$, thus $I(\texttt{sup}(t, s) = \mathbf{f})$. We have proved that $I_m(\texttt{defeated}(s, \sim p)) = \mathbf{f}$, which then imply $I_m(\texttt{overruled}(r, p)) = \mathbf{t}$. Thus, in all cases, $I_m(\texttt{defeasibly}(p)) = \mathbf{f}$ and $I_m(\texttt{not defeasibly}(p)) = \mathbf{t}$.

Inductive base, $\Leftarrow$. For the inductive base we consider the literals supported by I_1; however, according to clause c3 and clause c4, no literal $\texttt{defeasibly}(p)$ / $\texttt{not defeasibly}(p)$ is supported by I_1, thus we consider only $\texttt{definitely}(p)$ and $\texttt{not definitely}(p)$.

If $\texttt{definitely}(p)$ is supported by I_1 we have to analyse two cases, a successful instance of clause c1

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

or a successful instance of clause c2 with the following form:

$\texttt{definitely}(p)$:- $\texttt{strict}(r, p, [\,])$.

In the first case $I(\texttt{fact}(p)) = \mathbf{t}$ and so $p \in F$ and $F \subseteq +\Delta^m$, for every m. In the second case we a rule $r \in R_s[p]$ such that, trivially $A(r) \subseteq +\Delta^0$. Hence $p \in +\Delta_F$.

On the other hand $I_1(\texttt{not definitely}(p)) = \mathbf{t}$ iff $I_1(\texttt{definitely}(p)) = \mathbf{f}$. This means that all appropriate substitution of the clauses c1 and c2 fail. From clause c1 we obtain $I(\texttt{fact}(p)) = \mathbf{f}$ and so $p \notin F$. The only case when c2 fails wrt I is when $I(\texttt{strict}(r, p, [q_1, \ldots, q_l]))$ is false for all r, p, q. This implies that $R_s[p] = \emptyset$. It is now immediate to verify that the two conditions imply that $p \in -\Delta^1$, and by the monotonicity of $\mathscr{T}$, $p \in -\Delta_F$.

Inductive step. We assume that the property holds up to I_n, and we only show the cases different from the inductive base.

Case $I_{n+1}(\texttt{definitely}(p)) = \mathbf{t}$. Let us consider the following instance of clause c2:

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

Here $I_n(\texttt{strict}(r, p, [q_1, \ldots, q_l])) = \mathbf{t}$, and for all $q_i \in A(r)$, $I_n(\texttt{definitely}(q_i)) = \mathbf{t}$. By construction $r \in R_s[p]$, and by the inductive hypothesis $A(r) \subseteq +\Delta_F$.

Case $I_{n+1}(\texttt{not definitely}(p)) = \mathbf{f}$. Besides the cases of the inductive base we have to consider all ground substitutions of c2 where $I_n(\texttt{strict}(r, p, [q_1, \ldots, q_l])) = \mathbf{t}$. It follows that for some $q_i \in A(r)$, $I_n(\texttt{definitely}(q_i)) = \mathbf{f}$. By the inductive hypothesis $q_i \in -\Delta_F$, hence for some m, for all $r \in R_s[p]$, $A(r) \cap -\Delta^m \neq \emptyset$. Hence $p \in -\Delta_F$.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Case $I_{n+1}(\texttt{defeasibly}(p)) = \mathbf{t}$. We have to consider two cases. The first corresponds to the following clause:

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

Here $I_n(\texttt{definitely}(p)) = \mathbf{t}$, therefore, by the inductive hypothesis $p \in +\Delta_F$, thus there is an m such that $p \in +\Delta^m$, and consequently $p \in +\partial^m$, and $p \in +\partial_F$.

Otherwise we have

$\texttt{defeasibly}(p)$:- $\texttt{not definitely}(\sim p)$,
$\texttt{supportive_rule}(r, p, [q_1^r, \ldots, q_l^r])$,
$\texttt{defeasibly}(q_1^r), \ldots, \texttt{defeasibly}(q_l^r)$,
$\texttt{not overruled}(r, p)$.

where $I_n(\texttt{not definitely}(\sim p)) = \mathbf{t}$, $I_n(\texttt{supportive_rule}(r, p, [q_1^r, \ldots, q_l^r])) = \mathbf{t}$, and for all q_i^rs, $I_n(\texttt{defeasibly}(q_i^r)) = \mathbf{t}$. By construction and the inductive hypothesis $\sim p \in +\Delta_F$, $\exists r \in R_{sd}[p]$, and $A(r) \subseteq +\partial_F$. Moreover $I_n(\texttt{not overruled}(r, p)) = \mathbf{t}$, and therefore $I_n(\texttt{overruled}(r, p)) = \mathbf{f}$. This means that all substitutions of clause c5 are evaluated as false in I_n.

If $I_n(\texttt{rule}(s, \sim p, [q_1^s, \ldots, q_l^s])) = \mathbf{f}$, then $R[\sim p] = \emptyset$, and then trivially $p \in +\partial_F$.

Otherwise we consider the substitutions such that $I_n(\texttt{rule}(s, \sim p, [q_1^s, \ldots, q_l^s])) = \mathbf{t}$. For such substitutions either

$$I_n(\texttt{defeasibly}(q_i^s)) = \mathbf{f}, \text{ for some } q_i^s,$$

or

$$I_n(\texttt{not defeated}(s, \sim p)) = \mathbf{f}.$$

In the first case, by the inductive hypothesis, $q_i^s \in -\partial_F$. In the second case $I_n(\texttt{defeated}(s, \sim q)) = \mathbf{t}$. This implies $I_n(\texttt{supportive_rule}(t, p, [q_1^t, \ldots, q_l^t])) = \mathbf{t}$, $I_n(\texttt{defeasibly}(q_i^t)) = \mathbf{t}$ for all $q_i^t \in A(t)$, and $I_n(\texttt{supportive_rule}(t > s)) = \mathbf{t}$. Therefore, by construction and the inductive hypothesis, we obtain

$$\forall s \in R[\sim p] \text{ either } A(s) \cap -\partial_F \neq \emptyset \text{ or } \exists t \in R_{sd}[p] : A(t) \subseteq +\partial_F, t > s.$$

This implies that there is a suitable m where all those conditions are satisfied and hence $p \in +\partial^{m+1}$, and consequently $p \in +\partial_F$.

Case $I_{n+1}(\texttt{not defeasibly}(p)) = \mathbf{t}$. The proof of this case is analogous to the previous case. Indeed the proof of $\texttt{not defeasibly}(p)$ corresponds to the constructive negation of $\texttt{defeasibly}(p)$, and the conditions for $p \in -\partial^{n+1}$ are the negation of those for $p \in +\partial^{n+1}$. □

6 Conclusion

We motivated and presented a translation of defeasible theories into logic programs, such that the defeasible conclusions of the former correspond exactly with the sceptical conclusions of the latter under the stable model semantics, if a condition of decisiveness is satisfied. If decisiveness is not satisfied, we have to use Kunen semantics instead for a complete characterization.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

This paper closes an important gap in the theory of nonmonotonic reasoning, in that it relates defeasible logic with mainstream semantics of logic programming. This result is particularly important, since defeasible reasoning is one of the most successful nonmonotonic reasoning paradigms in applications.

Acknowledgements

Preliminary versions of the material included in this paper appeared at AAAI'99 (Maher and Governatori 1999) and ICLP'02 (Antoniou and Maher 2002).

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program.

References

Alferes, J. J. and Pereira, L. M. 1993a. Contradiction: When avoidance equals removal – Part I. In *Proc. 4th International Workshop on Extensions of Logic Programming*, R. Dyckhoff, Ed. LNCS, vol. 798. Springer, Berlin, 11–23.

Alferes, J. J. and Pereira, L. M. 1993b. Contradiction: When avoidance equals removal – Part II. In *Proc. 4th International Workshop on Extensions of Logic Programming*, R. Dyckhoff, Ed. LNCS, vol. 798. Springer, Berlin, 268–281.

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

Antoniou, G. 2002. Nonmonotonic rule system on top of ontology layer. In *ISWC 2002*, I. Horrocks and J. Hendler, Eds. LNCS, vol. 2432. Springer, Berlin, 394–398.

Antoniou, G., Billington, D., Governatori, G. and Maher, M. J. 2000. A flexible framework for defeasible logics. In *Proc. American National Conference on Artificial Intelligence (AAAI-2000)*. AAAI/MIT Press, Menlo Park, CA, 401–405.

Antoniou, G., Billington, D., Governatori, G. and Maher, M. J. 2001. Representation results for defeasible logic. *ACM Transactions on Computational Logic 2,* 2, 255–287.

Antoniou, G., Billington, D., and Maher, M. J. 1999. On the analysis of regulations using defeasible rules. In *Proc. 32nd Hawaii International Conference on Systems Science*. IEEE Press.

Antoniou, G. and Maher, M. J. 2002. Embedding defeasible logic into logic programs. In *Proc. International Conference on Logic Programming 2002*, P. Stuckey, Ed. LNCS, vol. 2401. Springer, Berlin, 393–404.

Antoniou, G., Maher, M. J. and Billington, D. 2000. Defeasible logic versus logic programming without negation as failure. *Journal of Logic Programming 41,* 1, 45–57.

Bassiliades, N., Antoniou, G. and Vlahavas, I. 2004. DR-DEVICE: A defeasible logic system for the Semantic Web. In *2nd Workshop on Principles and Practice of Semantic Web Reasoning*, H. J. Ohlbach and S. Schaffert, Eds. LNCS, vol. 3208. Springer, Berlin, 134–148.

Billington, D. 1993. Defeasible logic is stable. *Journal of Logic and Computation 3*, 370–400.

Billington, D., de Coster, K. and Nute, D. 1990. A modular translation from defeasible nets to defeasible logic. *Journal of Experimental and Theoretical Artificial Intelligence 2*, 151–177.

Gelder, A. V., Ross, K. A. and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 620–650.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. 5th International Conference and Symposium on Logic Programming*. MIT Press, Cambridge, MA, 1070–1080.

Governatori, G. 2005. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems 14,* 2–3, 181–216.

Governatori, G., Dumas, M., ter Hofstede, A. H. and Oaks, P. 2001. A formal approach to protocols and strategies for (legal) negotiation. In *Proc. 8th International Conference on Artificial Intelligence and Law (ICAIL01)*. ACM Press, 168–177.

Governatori, G., Maher, M. J., Billington, D. and Antoniou, G. 2004. Argumentation semantics for defeasible logics. *Journal of Logic and Computation 14,* 5, 675–702.

Governatori, G. and Padmanabhan, V. 2003. A defeasible logic of policy-based intention. In *AI 2003: Advances in Artificial Intelligence*, T. D. Gedeon and L. C. C. Fung, Eds. LNAI, vol. 2903. Springer, Berlin, 414–426.

Governatori, G. and Rotolo, A. 2004. Defeasible logic: Agency, intention and obligation. In *Deontic Logic in Computer Science*, A. Lomuscio and D. Nute, Eds. LNAI, vol. 3065. Springer, Berlin, 114–128.

Governatori, G., Rotolo, A. and Sadiq, S. 2004. A model of dynamic resource allocation in workflow systems. In *Database Technology 2004*, K.-D. Schewe and H. E. Williams, Eds. Conference Research and Practice of Information Technology, vol. 27. Australian Computer Science Association, ACS, 197–206.

Governatori, G., Rotolo, A. and Sartor, G. 2005. Temporalised normative positions in defeasible logic. In *Proc. 10th International Conference on Artificial Intelligence and Law (ICAIL05)*. ACM Press, New York, 25–34.

Grosof, B. N. 1997. Prioritized conflict handling for logic programs. In *Proceedings International Logic Programming Symposium*, J. Maluszynski, Ed. MIT Press, Cambridge, MA, 197–211.

Grosof, B. N. 2004. Representing e-commerce rules via situated courteous logic programs in RuleML. *Electronic Commerce Research and Applications 3,* 1, 2–20.

Grosof, B. N., Gandhe, M. D. and Finin, T. W. 2002. Sweetjess: Translating damlruleml to jess. In *Proc. of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web (RuleML 2002* (14 June 2002), M. Schroeder and G. Wagner, Eds. CEUR Workshop Proceedings, vol. 60. CEUR-WS.org.

Grosof, B. N., Labrou, Y. and Chan, H. Y. 1999. A declarative approach to business rules in contracts: Courteous logic programs in XML. In *Proceedings of the 1st ACM Conference on Electronic Commerce (EC99)*. ACM, ACM Press, New York, 68–77.

Grosof, B. N. and Poon, T. C. 2003. SweetDeal: representing agent contracts with exceptions using XML rules, ontologies, and process descriptions. In *Proc. 12th International Conference on World Wide Web*. ACM Press, New York, 340–349.

Horty, J. F., Thomason, R. H. and Touretzky, D. S. 1987. A skeptical theory of inheritance in nonmonotonic semantic networks. In *AAAI-87*. AAAI Press, Menlo Park, CA, 358–363.

Johnston, B. and Governatori, G. 2003. An algorithm for the induction of defeasible logic theories from databases. In *Database Technology 2003*, K.-D. Schewe and X. Zhou, Eds. Conference Research and Practice of Information Technology, vol. 17. Australian Computer Science Association, ACS, 75–83.

Kunen, K. 1987. Negation in logic programming. *Journal of Logic Programming 4*, 289–308.

Maher, M. J. 2000. A denotational semantics of defeasible logic. In *Computational Logic*, J. W. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, and P. J. Stuckey, Eds. LNCS, vol. 1861. Springer, Berlin, 209–222.

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press

Maher, M. J. 2001. Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming 1,* 6, 691–711.

Maher, M. J. 2002. A model-theoretic semantics for defeasible logic. In *Proc. ICLP2002 Workshop on Paraconsistent Computational Logic*, H. Decker, J. Villadsen, and T. Waragai, Eds. Datalogiske Skrifter, vol. 95. Roskilde University, Roskilde, Denmark, 67–80.

Maher, M. J. and Governatori, G. 1999. A semantic decomposition of defeasible logic. In *Proc. American National Conference on Artificial Intelligence (AAAI-99)*. AAAI Press, Menlo Park, CA, 299–305.

Maher, M. J., Rock, A., Antoniou, G., Billington, D. and Miller, T. 2001. Efficient defeasible reasoning systems. *International Journal on Artificial Intelligence Tools 10,* 4, 483–501.

Marek, V. and Truszczynski, M. 1993. *Nonmonotonic Reasoning*. Springer, Berlin.

Morgensten, L. 1998. Inheritance comes of age: Applying nonmonotonic techniques to problems in industry. *Artificial Intelligence 103*, 1–34.

Nute, D. 1994. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. Gabbay, C. Hogger, and J. Robinson, Eds. Vol. 3. Oxford University Press, Oxford, 353–395.

Prakken, H. 1997. *Logical Tools for Modelling Legal Arguments*. Kluwer, Dordrecht.

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

https://doi.org/10.1017/S1471068406002778 Published online by Cambridge University Press