TLP **13** (1): 107–142, 2013. © Cambridge University Press 2011
doi:10.1017/S147106841100041X First published online 9 August 2011

Interdefinability of defeasible logic and logic programming under the well-founded semantics

FREDERICK MAIER

Kno.e.sis Center, Department of Computer Science & Engineering, Wright State University,
3640 Colonel Glenn Hwy, Dayton, OH 45435, USA
(*e-mail:* fred@knoesis.org, fmaier@uga.edu)

submitted 10 September 2010; revised 21 May 2011; accepted 1 June 2011

Abstract

We provide a method of translating theories of Nute's defeasible logic into logic programs, and a corresponding translation in the opposite direction. Under certain natural restrictions, the conclusions of defeasible theories under the ambiguity propagating defeasible logic ADL correspond to those of the well-founded semantics for normal logic programs, and so it turns out that the two formalisms are closely related. Using the same translation of logic programs into defeasible theories, the semantics for the ambiguity blocking defeasible logic NDL can be seen as indirectly providing an ambiguity blocking semantics for logic programs. We also provide antimonotone operators for both ADL and NDL, each based on the Gelfond–Lifschitz (GL) operator for logic programs. For defeasible theories without defeaters or priorities on rules, the operator for ADL corresponds to the GL operator and so can be seen as partially capturing the consequences according to ADL. Similarly, the operator for NDL captures the consequences according to NDL, though in this case no restrictions on theories apply. Both operators can be used to define stable model semantics for defeasible theories.

KEYWORDS: defeasible logic, logic programming, well-founded semantics, stable model semantics, ambiguity blocking and propagation

1 Introduction

Defeasible logic is a family of rule-based nonmonotonic reasoning (NMR) formalisms originally developed by Nute (Nute 1986, 1994, 1997; Nute *et al.* 1989). Over the years, many variants have been proposed, with the most recent system created by Nute himself—an ambiguity blocking logic that we call NDL—appearing in the late 1990s (Nute 1999, 2003; Donnelly 1999). An ambiguity propagating counterpart to NDL, called ADL, was developed considerably later (Maier and Nute 2006). Working separately, Billington (1993) presented a quantified version of one of Nute's logics and showed it to be cumulative. Billington, together with Antoniou, Maher, Governatori, and others, would later go on to publish a number of papers on this logic and its offshoots (Antoniou *et al.* 2000a, 2000b, 2001, 2006; Governatori *et al.* 2004; Maher and Governatori 1999; Maher *et al.* 2001).

This logic—which we call BDL—and its variants are the ones most frequently encountered in the literature.

Defeasible logic possesses several virtues that warrant its study. It is of low computational complexity compared to, say, default logic (Reiter 1980) or logic programming under the stable model semantics (Gelfond and Lifschitz 1988). For example, some of the logics based on BDL have linear complexity (Maher *et al.* 2001). Also, the different variants of defeasible logic express a variety of different intuitions, and so, as Antoniou *et al.* (2000a) have said, defeasible logic forms a "flexible framework" for knowledge representation. Furthermore, the device primarily responsible for making defeasible logic defeasible—namely, the defeasible rule—is intuitively easy to grasp, arguably easier than the default negation $\sim$ used in logic programming. For example, we at least find the defeasible rule

$$\{bird(X)\} \Rightarrow fly(X)$$

(which might be read as "Birds usually can fly") to be more understandable than its logic program counterpart

$$fly(X) \leftarrow bird(X), \sim \neg fly(X).$$

Such considerations are important when it comes to creating and maintaining knowledge-based systems.

Defeasible logic is nevertheless relatively little known in the NMR community, and relationships to more mainstream NMR formalisms have been only partially studied. There are exceptions. Antoniou and Billington (2001) have shown how an ambiguity propagating variant of BDL can be embedded into default logic, and Brewka (2001) provided a simple scheme for translating the same logic (though without "team defeat") into logic programming under his own prioritized well-founded semantics (WFS) (Brewka 1996). Later, Antoniou *et al.* (2006) provided an alternative embedding of defeasible theories into logic programs and showed a relationship between the BDL-conclusions of theories and both the Kunen (1987) and stable model semantics of their embeddings.

Similar analyses have not been performed for NDL and ADL, however. NDL was not well known at the time Brewka's paper was written, and ADL did not exist until 2006. This is unfortunate, since these logics incorporate features not found in other defeasible logics. Particularly, NDL and ADL include *failure-by-looping*, a mechanism to weed out circular arguments. Its absence contributes greatly to the low complexity of the other logics, but it also means that the logics fail to draw reasonable conclusions in some cases. For example, if given the single rule

$$\{p\} \rightarrow p$$

and nothing else, then the earlier defeasible logics would be unable to conclude anything at all about p; it is neither provable nor refutable in these logics. However, based on the rule alone, there's no reason to believe p, and so it should be unprovable. Both NDL and ADL are able to show this.

Failure-by-looping is conceptually similar to the notion of an *unfounded set* in the WFS for logic programs (Van Gelder *et al.* 1991). Indeed, a recognition of this is what led to the development of adequate semantics (Maier 2010) for both

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

NDL and ADL—semantics that are based explicitly on the WFS. Historically, defeasible logics have been defined proof theoretically, with semantics coming only later. Prior to 2006, the only semantics offered for NDL (Donnelly 1999) was sound but incomplete, even for finite propositional theories.

Given the similarity of the semantics for NDL and ADL to the WFS for logic programs, it is natural to inquire whether each formalism can be translated into the other. For proponents of defeasible logic, the benefit of interdefinability with logic programs would be that preexisting logic program reasoners—e.g., XSB (Sagonas *et al.* 1994) or `smodels` (Syrjänen and Niemelä 2001)—could be used to draw conclusions according to defeasible logic, and many of the theoretical results already known about logic programs could be applied to defeasible theories. For proponents of logic programming, the benefit of interdefinability would be that results known about defeasible logic could be applied to logic programs. Furthermore, interdefinability would allow certain programs to be represented in a more concise and intuitively acceptable manner.

The present paper takes up interdefinability and related issues. It is shown here that for a restricted class of defeasible theories (those with minimal conflict sets, no defeaters, and no priorities on rules), the semantics for ADL corresponds to the WFS for normal logic programs. That is, there exists a rather natural translation of a defeasible theory into a logic program (under the WFS) that preserves the ADL-consequences of the theory. It is also shown that a consequence preserving translation exists in the other direction. The closed-world assumption of the WFS is easily represented as a set of defeasible rules with empty bodies. And so, given the restrictions, either formalism can be embedded into the other.

We also define antimonotone operators α and β for defeasible theories (with α propagating ambiguity and β blocking it) and show that when unprioritized defeasible theories are translated into logic programs, α coincides with the Gelfond–Lifschitz (GL) operator γ (Gelfond and Lifschitz 1988). Given this and the correspondence between ADL and the WFS, it immediately follows that an alternating fixpoint procedure based on α can be used on unprioritized theories to generate the consequences according to ADL. This parallels the known relationship between γ and the WFS (Baral and Subrahmanian 1993). Additionally, a similar fixpoint procedure, based on β, exactly captures the consequences according to NDL. Both operators can be used to define stable model semantics for defeasible theories.

Given that NDL blocks ambiguity and both ADL and the WFS propagate it, it is no surprise that there is no correspondence between the NDL-consequences of a defeasible theory and the well-founded model of its logic program counterpart. In general, the consequences according to the WFS and ADL are a subset of those according to NDL. However, using the same scheme to translate logic programs into defeasible theories, NDL can be viewed as indirectly providing an ambiguity blocking semantics for logic programs.

The remainder of the paper is organized as follows: Sections 2–4 give overviews, respectively, of logic programming with the WFS, the syntax of defeasible logic (in general), and the semantics of NDL and ADL (in particular). The consequence-preserving translations of defeasible theories into logic programs (and vice versa) are given in Sections 5 and 6. Section 7 presents the operators α and β. The

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

correspondence between α and γ is proven, as it is the relationship between β and the semantics for NDL. The operators are used in Section 8 to define semantics for defeasible theories akin to the stable model semantics. We conclude with a brief discussion of related work, discussing, in particular, the differences between ADL/NDL and other versions of defeasible logic.

Two appendices are also included. The first presents the proof systems for NDL and ADL. The second shows how defeasible theories can be transformed into equivalent ones lacking defeaters, extended conflict sets, and priorities on rules. Given this, it follows that α and logic programs under the WFS can both be used to compute all of the ADL consequences of defeasible theories.

2 The WFS for normal logic programs

A normal logic program Π consists of rules of the form

$$a \leftarrow b_1, b_2, \ldots, b_n, \sim c_1, \sim c_2, \ldots, \sim c_m,$$

where n and m are nonnegative integers, a and each b_i and c_i are first-order atomic formulas, and $\sim$ is *default negation*; each c_i is a *default literal.* In the original WFS, an interpretation $\mathscr{I}$ of program Π may be represented as a tuple $\langle \mathscr{T}, \mathscr{U} \rangle$, where $\mathscr{T}$ and $\mathscr{U}$ are disjoint sets of atoms. Interpretations are thus three valued; an atom p is *true* if it is in $\mathscr{T}$, *false* if it is in $\mathscr{U}$, and *undefined* otherwise. Interpretations in defeasible logic are similar, but in defeasible logic $\mathscr{T}$ and $\mathscr{U}$ can contain a mix of atoms *and* their negations (and so it is somewhat awkward to speak of truth and falsity). To avoid confusion, we will say that elements of $\mathscr{T}$ are *well founded* and those of $\mathscr{U}$ are *unfounded*. What remains is said to be *ambiguous*. The WFS selects one interpretation to serve as the canonical model (the well-founded model) of the program. Every program is guaranteed to have exactly one well-founded model.

In the following discussion, we assume that logic programs contain only ground terms and that the number of rules in the program is countable. $At(\Pi)$ is the set of ground atoms associated with a program, while $Lit(\Pi)$ is defined to be $At(\Pi) \cup \{\neg p | p \in At(\Pi)\}$. The literal $\neg p$ is the classical complement of atom p. When it is necessary to talk about a literal and its complement in a general way, we will use of the following notation: if p is an atom a, then $\overline{p}$ is $\neg a$. If p is $\neg a$, where a is an atom, then $\overline{p}$ is a.

The expressions *head(r)* and *body(r)* refer, respectively, to the head and body of the rule r, and $body(r)^-$ and $body(r)^+$ refer, respectively, to the default and non-default literals of $body(r)$. Similarly, r^+ is r with all default literals removed. An *NAF-free* (or *definite*) logic program is a normal logic program containing no default literals.

If $\mathscr{T}$ and $\mathscr{U}$ are allowed to overlap in interpretations, then the set of all interpretations forms a complete lattice under the relation $\sqsubseteq$, where

$$\langle \mathscr{T}_1, \mathscr{U}_1 \rangle \sqsubseteq \langle \mathscr{T}_2, \mathscr{U}_2 \rangle \textit{ iff } \mathscr{T}_1 \subseteq \mathscr{T}_2 \text{ and } \mathscr{U}_1 \subseteq \mathscr{U}_2.$$

This is the so-called knowledge ordering. The bottom $\bot$ of the lattice is $\langle \emptyset, \emptyset \rangle$ and the top $\top$ is $\langle At(\Pi), At(\Pi) \rangle$. The well-founded model of Π is defined using the

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

operators U_Π, T_Π, and W_Π, all of which are monotone on the lattice. T_Π is the *immediate consequence operator*:

$$T_\Pi(\mathscr{I}) = \{head(r) \mid r \in \Pi,\, body(r)^+ \subseteq \mathscr{T}, \text{ and } body(r)^- \subseteq \mathscr{U}\}.$$

U_Π and W_Π are defined via *unfounded sets*, which intuitively are sets for which no external support exists. If Π is a normal logic program and $\mathscr{I} = \langle \mathscr{T}, \mathscr{U} \rangle$ an interpretation, then a set $S \subseteq At(\Pi)$ is an *unfounded set* of Π *wrt* interpretation $\mathscr{I}$ *iff* for each $p \in S$ and each rule $r \in \Pi$ with head p, either

(1) there is a $q \in body(r)^+$ such that $q \in \mathscr{U} \cup S$, or
(2) there is a $q \in body(r)^-$ such that $q \in \mathscr{T}$.

Unfounded sets are closed under union. $U_\Pi(\mathscr{I})$ is the *greatest* unfounded set of Π *wrt* $\mathscr{I}$:

$$U_\Pi(\mathscr{I}) = \bigcup\{A \mid A \text{ is an unfounded set of } \Pi \text{ with respect to } \mathscr{I}\}.$$

$U_\Pi(\mathscr{I})$ and $T_\Pi(\mathscr{I})$ are disjoint, and W_Π combines them to form a new interpretation:

$$W_\Pi(\mathscr{I}) = \langle T_\Pi(\mathscr{I}), U_\Pi(\mathscr{I})\rangle.$$

Beginning with $\bot$, the following sequence $(\mathscr{I}_0, \mathscr{I}_1, \ldots)$ is defined using W_Π:

(1) $\mathscr{I}_0 = W_\Pi \uparrow 0 = \langle \varnothing, \varnothing \rangle$,
(2) $\mathscr{I}_{\alpha+1} = W_\Pi \uparrow \alpha + 1 = W_\Pi(\mathscr{I}_\alpha)$ (for successor ordinals),
(3) $\mathscr{I}_\alpha = W_\Pi \uparrow \alpha = \langle \bigcup_{\beta<\alpha} \mathscr{T}_\beta, \bigcup_{\beta<\alpha} \mathscr{U}_\beta \rangle$ (for limit ordinals).

The well-founded model $wfm(\Pi)$ of Π is defined to be $W_\Pi \uparrow \lambda$, where λ is the closure ordinal of the sequence—i.e., the least λ such that $W_\Pi \uparrow \lambda = W_\Pi \uparrow (\lambda + 1)$. Since W_Π is monotone on the lattice of interpretations, then by the Knaster–Tarski Theorem (Tarski 1955), least $lfp(W_\Pi)$ and greatest $gfp(W_\Pi)$ fixpoints of W_Π exist. The well-founded model may equivalently be defined as $lfp(W_\Pi)$.

Example 1
(1) $p \leftarrow\sim q_0$
(2) $q_n \leftarrow q_{n+1}$ *(for all $n \in \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of nonnegative integers.)*

The well-founded model of the above infinite program is $\langle \{p\}, \{q_n | n \in \mathbb{N}_0\}\rangle$. Clearly, each q_i is intuitively unfounded. In fact, $\mathscr{U}_1 = \{q_n | n \in \mathbb{N}_0\}$. Given this, $p \in T_\Pi(\mathscr{I}_1)$ (in other words, $p \in \mathscr{T}_2$). The closure ordinal of the sequence is 2.

3 Defeasible logic

Like logic programs, defeasible logic deals with sets of rules, where the rules are composed of sets of ground literals (atoms and their classical complements). Unlike logic programs, however, defeasible logic allows three sorts of rules. If S is a finite set of literals and p is a literal, then $S \rightarrow p$ is a *strict* rule, $S \Rightarrow p$ is a *defeasible* rule, and $S \leadsto p$ is an *(undercutting) defeater*. We may read $S \rightarrow p$ as saying "If S, then

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

definitely p," $S \Rightarrow p$ as "If S, then *defeasibly* (*normally*, *apparently*, *evidently*) p," and $S \rightsquigarrow p$ as "If S, then *maybe p*. Strict rules with empty bodies are called *facts* and defeasible rules with empty bodies are called *presumptions*. The rule $\emptyset \Rightarrow p$ may be read as saying "Presumably, p."

Defeasible logic is intended to be a logic of justification, and in our view, such a logic must be nonmonotonic. From an intuitive standpoint, it is possible for a belief to be justified and nevertheless false, and it is also possible for a belief that was once justified to lose its justification—not because its support has been rejected, but because new information has come to light that either contradicts the belief directly or else undermines it by contradicting its support.

This basic intuition is captured in defeasible logic by allowing defeasible rules to be *defeated* by other rules. Given the rules

(1) $\emptyset \rightarrow lives\_alone$
(2) $\emptyset \rightarrow has\_a\_wife$
(3) $\{lives\_alone\} \Rightarrow \neg married$
(4) $\{has\_a\_wife\} \rightarrow married$,

it is reasonable to conclude $lives\_alone$ and has_a_wife, as both are facts. However, we cannot, on pain of contradiction, simultaneously detach the heads of *both* of the latter two rules. In defeasible logic, to detach the head of rule 3, we must first show that rule 4 *cannot* be applied. This, in fact, we cannot do. Though the different defeasible logics might formalize the intuition differently, rule 3 is defeasible and rule 4 is strict, and rule 4 defeats rule 3. In each logic, one can conclude *married* but not $\neg married$.

The heads of defeaters can never be detached—defeaters exist solely to prevent the application of a conflicting defeasible rule. For example, the defeater $\neg$ *has-intact-flight-feathers* $\rightsquigarrow \neg flies$ might be used to prevent a proof of $flies$ from $\{bird\} \Rightarrow flies$, but it cannot be used to directly prove $\neg flies$.

Strict, defeasible, and defeater rules are collected into *defeasible theories*. Below, if D is a defeasible theory, then $At(D)$ and $Lit(D)$ are defined as they are in logic programming: $At(D)$ is the set of atoms associated with D, while $Lit(D)$ is the set of literals.

Definition 1
A *defeasible theory* D is a triple $\langle R, C, \prec \rangle$, where R is a countable set of rules, C is a countable set of finite sets of literals in $Lit(D)$ such that for any literal $p \in At(D)$, $\{p, \neg p\} \in C$, and $\prec$ is an acyclic binary relation over the non-strict rules in R.

The elements of C are called *conflict sets*. It is these which determine incompatibilities in defeasible theories. Simply put, a collection of rules conflict if their heads constitute a conflict set. The priority relation $\prec$ is used to resolve conflicts between non-strict rules, and it is this relation that in part determines, which rules can be used to defeat others. Different versions of defeasible logic specify precisely how these components are used.

We call a conflict set of the form $\{p, \neg p\}$ a *minimal* conflict set, and we use C_{MIN} to indicate that no conflict sets other than the minimal ones are defined in a given

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

defeasible theory. We say that conflict sets are *closed under strict rules* if, for all $c \in C$, if $A \rightarrow p$ is a rule and $p \in c$, then $(A \cup c - \{p\}) \in C$. It is expensive to close conflict sets under the strict rules of a theory, but it is often necessary to do so in order to draw reasonable conclusions. The non-minimal conflict sets are called *extended* conflict sets. The predecessors of NDL and ADL, including BDL and its variants, do not allow extended conflict sets.

For a given theory, R_s, R_d, and R_u refer to the strict, defeasible, and defeater rules of R, respectively, while $R_s[p]$, $R_d[p]$, and $R_u[p]$ refer to those rules with head p. $C[p]$ denotes the set of conflict sets containing p. The expression $\overline{p}$ refers to the complement of p.

4 Well-founded semantics for ADL and NDL

The proof systems for ADL and NDL are presented in Appendix A. We will not discuss them further here other than to say that adequate semantics (presented below) corresponding to the logics did not exist until recently (Maier 2010). The proof systems for NDL and ADL are sound relative to their counterpart semantics, and while completeness does not hold in general, the proof systems are complete for the class of *locally finite theories* (defined in Appendix A). The restricted nature of the completeness result is unsurprising, since proofs in defeasible logic are required to be finite structures, while no counterpart restriction exists for the semantics.

As Example 2 illustrates, NDL and ADL differ in how ambiguity is handled.

Example 2
$D = \langle R, C_{MIN}, \emptyset \rangle$, R is

(1) $\emptyset \Rightarrow p$
(2) $\emptyset \Rightarrow \neg p$
(3) $\{p\} \Rightarrow \neg q$
(4) $\emptyset \Rightarrow q$.

The first two defeasible rules in the example are vacuously supported and conflict, and there is no mechanism for choosing between them. The literals p and $\neg p$ are *ambiguous* in an intuitive sense, and there is some debate in the literature regarding the proper handling of ambiguity. It is clear that neither p nor $\neg p$ should be considered justified. One possible course of action is to consider both p and $\neg p$ as refuted, which effectively *blocks* or localizes the ambiguity to just those literals. This is the course taken by Horty (Horty *et al.* 1990) and by most forms of defeasible logic, including NDL. If one does this, then since p is refuted, all support for $\neg q$ vanishes, and only q is left with any support. Indeed, under the ambiguity blocking view, q is proved while $\neg q$ is refuted.

Alternatively, one could simply refrain from concluding anything at all about p and $\neg p$. Since the status of $\neg q$ depends upon resolving the status of p, the ambiguity of p is effectively *propagated* to $\neg q$. This is the course taken by ADL. Adopting ambiguity propagation yields a more extreme form of skepticism, in the sense that fewer conclusions can be drawn. In the example, p *might* hold (there is conflicting

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

information about it and no way to resolve the conflict), and if it does hold, then there would be evidence for both q and $\neg q$, and so q and $\neg q$ would be ambiguous.

The WFS for logic programs is also ambiguity propagating. The following logic program Π is the most-natural counterpart to the above defeasible theory:

(1) $p \leftarrow\!\sim \neg p$
(2) $\neg p \leftarrow\!\sim p$
(3) $\neg q \leftarrow\!\sim q, p$
(4) $q \leftarrow\!\sim \neg q$.

If $\neg p$ and $\neg q$ are simply treated as atoms (which is what the original WFS would do), it can be seen that no non-empty subset of $At(\Pi)$ is an unfounded set relative to $\bot = \langle \emptyset, \emptyset \rangle$. Furthermore, $T_\Pi(\bot)$ is empty. As this is so, the well-founded model of the program is simply $\bot$.

We note that in Example 2, the ambiguity of p and $\neg p$ can be resolved by specifying that either rule 1 or 2 takes priority over the other. For instance, if $2 \prec 1$, then both NDL and ADL would conclude p, and $\neg p$ would be refuted. In both logics, q and $\neg q$ would still be ambiguous because neither rule 3 nor 4 is superior to the other.

The semantics for both NDL and ADL are based explicitly on the WFS. As noted earlier, the components $\mathscr{T}$ and $\mathscr{U}$ of interpretations are allowed to contain negative literals. The set of interpretations still forms a complete lattice under $\sqsubseteq$, with $\top$ now being $\langle Lit(D), Lit(D) \rangle$. The operators U_D, T_D, and W_D, as well as the underlying notion of unfounded set, are recast to apply to defeasible theories. Somewhat surprisingly, the difference between the semantics for NDL and that for ADL lies solely in how unfounded sets are defined. It is this definition that determines whether ambiguity is blocked or propagated.

Definition 2

A set $S \subseteq Lit(D)$ is *unfounded* in NDL with respect to D and an interpretation $\mathscr{I} = \langle \mathscr{T}, \mathscr{U} \rangle$ iff for all literals $p \in S$:

(1) For every $r \in R_s[p]$, $body(r) \cap (\mathscr{U} \cup S) \neq \emptyset$.
(2) For every $r \in R_d[p]$,
  (a) $body(r) \cap (\mathscr{U} \cup S) \neq \emptyset$, or
  (b) there is a $c \in C[p]$ such that for each $q \in c - \{p\}$ there is a rule $s \in R[q]$ such that
    (i) $body(s) \subseteq \mathscr{T}$ and,
    (ii) $s \nprec r$.

The definition of unfounded set in ADL is exactly the same as for NDL, save that condition 2b(ii) is replaced with the following requirement: $r \prec s$ *or* s *is strict*.

Examining the definition above and Example 2, it can be seen that $\{p, \neg p\}$ is an unfounded set in NDL relative to D and interpretation $\langle \emptyset, \emptyset \rangle$. It is not unfounded according to ADL, however.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Given an account of unfounded set, U_D is defined identically for NDL and ADL:

$$U_D(\mathscr{I}) = \bigcup\{S \mid S \text{ is an unfounded set } wrt\ D \text{ and } \mathscr{I}\}.$$

The immediate consequence operator for NDL and ADL is defined in terms of *witnesses of provability.*

Definition 3
If D is a defeasible theory and $\mathscr{I} = \langle \mathscr{T}, \mathscr{U} \rangle$ an interpretation, then a rule $r \in R_D$ is a *witness of provability* for p *wrt* D and $\mathscr{I}$ if one of the below conditions applies:

(1) $r \in R_s[p]$ and $body(r) \subseteq \mathscr{T}$.
(2) $r \in R_d[p]$ and $body(r) \subseteq \mathscr{T}$, and for each conflict set $c \in C[p]$, there exists a $q \in c - \{p\}$ such that for all $s \in R[q]$, $s \prec r$ or $body(s) \cap \mathscr{U} \neq \emptyset$.

Given this, the *immediate consequences* of D *wrt* $\mathscr{I}$, written $T_D(\mathscr{I})$, is the set

$$T_D(\mathscr{I}) = \{p \mid \text{there exists a witness of provability for } p\ wrt\ D \text{ and } \mathscr{I}\}.$$

The account of the operator W remains unchanged from the WFS:

$$W_D(\mathscr{I}) = \langle T_D(\mathscr{I}), U_D(\mathscr{I})\rangle.$$

Furthermore, W_D may be used to define the monotonically increasing sequence $(\mathscr{I}_0, \mathscr{I}_1, \ldots)$. The sequence is *coherent*, in the sense that $\mathscr{T}_\alpha \cap \mathscr{U}_\alpha = \emptyset$ for any $\alpha \geqslant 0$. As in the WFS, the well-founded model of the defeasible theory is defined to be the least fixpoint $lfp(W_D)$ of W_D:

$$wfm(D) =_{def} lfp(W_D).$$

Again, it is the case that this fixpoint coincides with the limit of the above sequence.

The well-founded models of defeasible theories (under NDL or ADL) can be viewed as defining both a consequence relation $\mid\!\approx$ and an "anticonsequence" relation $\approx\!\mid$. Analogous relations ($\mid\!\approx_{WFS}$ and $\approx\!\mid_{WFS}$) can be defined for normal logic programs under the WFS.

Definition 4
Let D be a defeasible theory, L one of NDL or ADL, and $wfm(D) = \langle \mathscr{T}, \mathscr{U} \rangle$ D's well-founded model according to L.

(1) $D \mid\!\approx_L p$ *iff* $p \in \mathscr{T}$, and
(2) $D \approx\!\mid_L p$ *iff* $p \in \mathscr{U}$.

5 Translating defeasible theories into logic programs

In the scheme used by Brewka (2001) to translate defeasible theories into logic programs, every defeasible rule $S \Rightarrow p$ becomes $p \leftarrow\sim \overline{p}, S$, and every strict rule $S \rightarrow p$ becomes $p \leftarrow S$. The result of the transformation is a so-called *extended* logic program (which allows both $\sim$ and $\neg$ to be used). Several examples are presented to demonstrate that the two systems do not always agree, and Brewka argues that the results of the defeasible logic are less reasonable. The particular logic investigated

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

by Brewka is an ambiguity propagating variant of BDL (without team defeat, a feature we have not discussed). Brewka dismisses other variants of defeasible logic without discussing them in any detail, mainly because these logics are ambiguity blocking.

Brewka's translation scheme assumes minimal conflict sets. Below, we alter it to encompass theories with extended conflict sets, and we use this modified scheme to compare ADL to the simple WFS, i.e., not to Brewka's prioritized variant. Since conflict sets are sufficient to encode negation, we will assume that all negative literals are just atoms. Furthermore, since defeaters and priorities on rules are not defined for the simple WFS, we assume that no defeaters occur in the theory and that $\prec$ is empty.

Definition 5
Let $D = \langle R, C, \prec \rangle$ be a defeasible theory. For any literal $p \in Lit(D)$,

$$Prod(C[p]) = \{\{a_1, \ldots, a_m\} | (a_1, \ldots, a_m) \in c_1 - \{p\} \times \ldots \times c_m - \{p\}\},$$

where $C[p] = \{c_1, \ldots, c_m\}$.

$Prod(C[p])$ is the set of all sets that can be created by taking a single literal (other than p) from each conflict set containing p (the order in the n-ary product above does not matter). We use these sets when translating defeasible rules of a theory into logic program rules. In order to ensure that the rules in the translation are finite in length, we require that $C[p]$ is finite for each $p \in Lit(D)$.

Definition 6
Let $D = \langle R, C, \emptyset \rangle$ be a defeasible theory such that $R_u = \emptyset$ and $C[p]$ is finite for each $p \in Lit(D)$. The logic program translation Π_D of D is the smallest rule-set such that

(1) if $\{q_1, \ldots, q_n\} \rightarrow p \in R_s$, then $p \leftarrow q_1, \ldots, q_n \in \Pi_D$, and
(2) if $\{q_1, \ldots, q_n\} \Rightarrow p \in R_d$ and $\{a_1, \ldots, a_m\} \in Prod(C[p])$, then $p \leftarrow\sim a_1, \ldots, \sim a_m, q_1, \ldots, q_n \in \Pi_D$.

Let $trans(r)$ denote the set of logic program rules obtained from rule r of the defeasible theory. If we ignore notational differences, then $trans(r) = \{r\}$ if r is strict. For the sake of convenience, we will simply say that $trans(r) = r$. Normally, if r is defeasible, then $trans(r)$ will contain many rules, but if conflict sets are minimal, then it again holds that $trans(r)$ contains only a single rule. In the translation of a defeasible rule r, each a_i is a literal of some conflict set containing p.

5.1 *Soundness and completeness of ADL* wrt *WFS*

Provided $C[p]$ is finite for each $p \in Lit(D)$, ADL is sound *wrt* the WFS. If in addition conflict sets are minimal, ADL is complete *wrt* the WFS.

Below, since the operators for the WFS have direct analogs for defeasible logic and are defined individually for each logic program and defeasible theory, we can use the same basic symbols for each (writing, for instance, T_D and T_Π) without causing confusion. We will use $(\mathscr{I}_{D,0}, \mathscr{I}_{D,1}, \ldots)$ to denote the sequence of interpretations

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

obtained using W_D, and $(\mathscr{I}_{\Pi,0}, \mathscr{I}_{\Pi,1}, \ldots)$ to denote the sequence of interpretations obtained using W_Π. For a given interpretation $\mathscr{I}_{\Pi,\lambda}$, we will write $\mathscr{T}_{\Pi,\lambda}$ and $\mathscr{U}_{\Pi,\lambda}$ to distinguish well-founded and unfounded sets. We will also sometimes indicate the well-founded model of a given theory D (or program Π) by writing $\mathscr{I}_{D,WF}$ ($\mathscr{I}_{\Pi,WF}$).

Proposition 1 (*Soundness of ADL* wrt *WFS*)
Let $D = \langle R, C, \emptyset \rangle$ be a defeasible theory such that $R_u = \emptyset$ and for each $p \in Lit(D)$, $C[p]$ is finite. Let Π be the logic program translation of D. For any $p \in Lit(D)$,

(1) if $D \mathrel{|\!\!\approx}_{ADL} p$, then $\Pi \mathrel{|\!\!\approx}_{WFS} p$, and
(2) if $D \mathrel{\approx\!\!|}_{ADL} p$, then $\Pi \mathrel{\approx\!\!|}_{WFS} p$.

Proof
The proof is by induction on the sequence $(\mathscr{I}_{D,0}, \mathscr{I}_{D,1}, \ldots)$, showing that for all $\lambda \geqslant 0$, if $p \in \mathscr{T}_{D,\lambda}$ ($p \in \mathscr{U}_{D,\lambda}$), then $p \in \mathscr{T}_{\Pi,WF}$ ($p \in \mathscr{U}_{\Pi,WF}$). Since $\mathscr{I}_{D,0} = \mathscr{I}_{\Pi,0}$, the claim holds for $\lambda = 0$. Suppose it holds for all $\kappa < \lambda$. We may assume *wlog* that λ is a successor ordinal. There are two cases to consider:

(1) Suppose $p \in \mathscr{T}_{D,\lambda}$. Then there exists an $r \in R[p]$ such that $body(r) \subseteq \mathscr{T}_{D,\lambda-1}$ and either (1) r is strict or else (2) r is defeasible and for each conflict set $c \in C[p]$, there exists a $q \in c - \{p\}$ such that for all $s \in R[q]$, $body(s) \cap \mathscr{U}_{D,\lambda-1} \neq \emptyset$. In both cases, $body(r) \subseteq \mathscr{T}_{\Pi,WF}$ by the inductive hypothesis. As such, for any $r' \in trans(r)$, $body(r')^+ \subseteq \mathscr{T}_{\Pi,WF}$. If r is strict then $body(r')^+ = body(r')$ and so by definition of T_Π, $p \in \mathscr{T}_{\Pi,WF}$.
Suppose, instead, that r is defeasible, and let $c \in C[p]$. Then there exists a $q \in c - \{p\}$ such that for each $s \in R[q]$, $body(s) \cap \mathscr{U}_{D,\lambda-1} \neq \emptyset$. Let $s' \in trans(s)$. Since $body(s) = body(s')^+$, it follows that $body(s')^+ \cap \mathscr{U}_{D,\lambda-1} \neq \emptyset$. By inductive hypothesis, $body(s')^+ \cap \mathscr{U}_{\Pi,WF} \neq \emptyset$. Generalizing on s' and then s, every rule for q in Π has a non-default literal in $\mathscr{U}_{\Pi,WF}$, and so by definition of U_Π, $q \in \mathscr{U}_{\Pi,WF}$.
Generalizing on c, every conflict set for p has a literal $q \neq p$ such that $q \in \mathscr{U}_{\Pi,WF}$. Let $Q = \{q_1, \ldots, q_m\}$ be the set of such literals. Obviously, $Q \in Prod(C[p])$, and so there is a rule $r' \in trans(r)$ such that $body(r')^+ = body(r)$ and $body(r')^- = Q$. Since $body(r')^+ \subseteq \mathscr{T}_{\Pi,WF}$, and $Q \subseteq \mathscr{U}_{\Pi,WF}$, it follows that $p \in \mathscr{T}_{\Pi,WF}$.

(2) Suppose $p \in \mathscr{U}_{D,\lambda}$ and let b be any literal in $\mathscr{U}_{D,\lambda}$. $\mathscr{U}_{D,\lambda}$ is by definition unfounded wrt D and $\mathscr{I}_{D,\lambda-1}$. If $r \in R_s[b]$, then there is a $q \in body(r)$ such that $q \in \mathscr{U}_{D,\lambda} \cup \mathscr{U}_{D,\lambda-1}$. U_D is monotone, and so $q \in \mathscr{U}_{D,\lambda}$.
Suppose $r \in R_d[b]$. Then either (1) there is a $q \in body(r)$ such that $q \in \mathscr{U}_{D,\lambda}$, or (2) there is a conflict set $c \in C[b]$ such that for all $a \in c - \{b\}$, there is a $s \in R_s[a]$ such that $body(s) \subseteq \mathscr{T}_{D,\lambda-1}$ (s must be strict since the priority relation is empty). Suppose (2) holds. By inductive hypothesis, $body(s) \subseteq \mathscr{T}_{\Pi,WF}$. Since s is strict, $trans(s) = s$ and so $a \in \mathscr{T}_{\Pi,WF}$. Recall that by definition of $Prod(C[b])$, for each set $Q \in Prod(C[b])$ we have $Q \cap c - \{b\} \neq \emptyset$. By definition of $trans(r)$, for each $t \in trans(r)$, there exists a $Q \in Prod(C[b])$ such that $Q = body(t)^-$. Since this is so, if (2) holds then for each rule $r' \in trans(r)$, there exists a $a \in body(r')^-$ such that $a \in \mathscr{T}_{\Pi,WF}$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Generalizing on r, every logic program rule r' for b has a classical literal $q \in body(r')^+$ such that $q \in \mathscr{U}_{D,\lambda}$, or else a default literal $a \in body(r')^-$ such that $a \in \mathscr{T}_{\Pi,WF}$. Generalizing on b, by definition of unfounded sets for logic programs, $\mathscr{U}_{D,\lambda}$ is an unfounded set relative to Π and $\mathscr{I}_{\Pi,WF}$, and so $\mathscr{U}_{D,\lambda} \subseteq \mathscr{U}_{\Pi,WF}$. Since $p \in \mathscr{U}_{D,\lambda}$, $p \in \mathscr{U}_{\Pi,WF}$. □

As noted, the claim of soundness pertains to defeasible theories with extended conflict sets, provided that $C[p]$ is finite for all p. The other specific requirements are that no defeaters are used and that the rules are unprioritized. Completeness requires more, however. Specifically, conflict sets must also be minimal.

Example 3

Consider the following unprioritized defeasible theory and its corresponding logic program, and suppose that the conflicts sets $\{p, \neg p\}$, $\{q, \neg q\}$, $\{q, \neg p\}$ are used.

(1) $\varnothing \rightarrow p$	(1) p
(2) $\varnothing \Rightarrow \neg p$	(2) $\neg p \leftarrow \sim p$
(3) $\varnothing \Rightarrow q$	(3) $q \leftarrow \sim \neg q, \sim \neg p$
(4) $\{q\} \rightarrow p$	(4) $p \leftarrow q$

In the defeasible theory, p is a fact, and so the presumption of $\neg p$ is defeated. Nevertheless, it is supported, and this is sufficient to prevent q from being concluded—q is ambiguous according to ADL. In the logic program, however, both p and q are well founded, and so, the two formalisms disagree. In contrast, if the conflict sets are minimal, the defeasible theory and corresponding logic program yield the same results. Examples such as above are problematic for ADL and NDL, as the result produced by the logic program is intuitively more reasonable than the one produced by the defeasible theory with conflict sets closed under strict rules. We pick up this topic again in Section 9.

Before we prove that ADL is complete with respect to the WFS—subject to the restrictions noted above—we need to prove the following small lemma.

Lemma 1

Let $D = \langle R, C, \varnothing \rangle$ be a defeasible theory such that $R_u = \varnothing$ and the conflict sets of C are minimal. For all $p \in Lit(D)$, if $r \in R_d[p]$, $body(r) \subseteq \mathscr{T}_{WF}$, and $\overline{p} \in \mathscr{U}_{WF}$, then $p \in \mathscr{T}_{WF}$.

Proof

Suppose $r \in R_d[p]$ and $body(r) \subseteq \mathscr{T}_{WF}$ and $\overline{p} \in \mathscr{U}_{WF}$. Then there must be some least successor ordinal λ such that $body(r) \subseteq \mathscr{T}_{\lambda}$ and $\overline{p} \in \mathscr{U}_{\lambda}$. Recall that $\mathscr{U}_{\lambda}$ is the greatest unfounded set wrt $\mathscr{I}_{\lambda-1}$. Suppose for a proof by contradiction that $p \notin \mathscr{T}_{\lambda+1}$. By definition of unfounded set (and since $C = C_{MIN}$), we have

(1) for all $s \in R_s[\overline{p}]$, $body(s) \cap (\mathscr{U}_{\lambda} \cup \mathscr{U}_{\lambda-1}) \neq \varnothing$, and
(2) for all $s \in R_d[\overline{p}]$, either
  (a) $body(s) \cap (\mathscr{U}_{\lambda} \cup \mathscr{U}_{\lambda-1}) \neq \varnothing$, or
  (b) there is a rule $t \in R_s[p]$ such that $body(t) \subseteq \mathscr{T}_{\lambda-1}$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Since U_D is monotone, $body(s) \cap (\mathscr{U}_\lambda \cup \mathscr{U}_{\lambda-1})$ reduces to $body(s) \cap \mathscr{U}_\lambda$. If 2b above holds, then $p \in \mathscr{T}_\lambda$ and so $p \in \mathscr{T}_{\lambda+1}$. As such, $r \in R_d[p]$ and $body(r) \subseteq \mathscr{T}_\lambda$, and it must be that for each rule $s \in R[\overline{p}]$, $body(s) \cap \mathscr{U}_\lambda \neq \emptyset$. But this implies (via T_D) that $p \in \mathscr{T}_{\lambda+1}$. This is a contradiction, and so (again) $p \in \mathscr{T}_{\lambda+1}$. □

Proposition 2 (*Completeness*)
Let $D = \langle R, C, \emptyset \rangle$ be a defeasible theory such that $R_u = \emptyset$ and the conflict sets of C are minimal. Let Π be the logic program translation of D. For all $p \in Lit(D)$,

(1) if $\Pi \mid\!\approx_{WFS} p$, then $D \mid\!\approx_{ADL} p$, and
(2) if $\Pi \approx\!\mid_{WFS} p$, then $D \approx\!\mid_{ADL} p$.

Proof
The proof is by induction on the sequence $(\mathscr{I}_{\Pi,0}, \mathscr{I}_{\Pi,1}, \ldots)$, showing that for all $\lambda \geqslant 0$, if $p \in \mathscr{T}_{\Pi,\lambda}$ $(p \in \mathscr{U}_{\Pi,\lambda})$, then $p \in \mathscr{T}_{D,WF}$ $(p \in \mathscr{U}_{D,WF})$. Since $\mathscr{I}_{D,0} = \mathscr{I}_{\Pi,0}$, the clam holds for $\lambda = 0$. Suppose it holds for all $\kappa < \lambda$. We may assume that λ is a successor ordinal.

(1) Suppose $p \in \mathscr{T}_{\Pi,\lambda}$. Then there is a rule $s \in trans(r)$ for some $r \in R[p]$ such that $body(s)^+ \subseteq \mathscr{T}_{\Pi,\lambda-1}$ and $body(s)^- \subseteq \mathscr{U}_{\Pi,\lambda-1}$. By inductive hypothesis, $body(s)^+ \subseteq \mathscr{T}_{D,WF}$ and $body(s)^- \subseteq \mathscr{U}_{D,WF}$. If $body(s)^- = \emptyset$, then $r \in R_s[p]$ and $body(r) = body(s)$, and so $p \in \mathscr{T}_{D,WF}$ by definition of $\mathscr{I}_{D,WF}$ and T_D. If $body(s)^- \neq \emptyset$, since conflict sets are minimal, it must be that $body(s)^- = \{\overline{p}\}$, and so, $\overline{p} \in \mathscr{U}_{D,WF}$. Since $body(s)^+ = body(r)$, by Lemma 1, $p \in \mathscr{T}_{D,WF}$.
(2) Suppose $p \in \mathscr{U}_{\Pi,\lambda}$. Let a be a literal of $\mathscr{U}_{\Pi,\lambda}$ and let $r' \in trans(r)$ for some $r \in R[a]$. If $r \in R_s[a]$, then since U_Π is monotone and $\mathscr{U}_{\Pi,\lambda}$ is unfounded relative to $\mathscr{I}_{\Pi,\lambda-1}$, there exists a $b \in body(r')^+$ such that $b \subseteq \mathscr{U}_{\Pi,\lambda}$. Thus there exists a $b \in body(r)$ such that $b \in \mathscr{U}_{\Pi,\lambda}$.
Suppose that r is defeasible. Then either (1) there exists a classical $b \in body(r')$ such that $b \in \mathscr{U}_{\Pi,\lambda}$, or else (2) the literal $\sim \overline{a}$ appears in $body(r')$ and $\overline{a} \in \mathscr{T}_{\Pi,\lambda-1}$. If (1), then $body(r) \cap \mathscr{U}_{\Pi,\lambda} \neq \emptyset$. If (2), then by the inductive hypothesis $\overline{a} \in \mathscr{T}_{D,WF}$, and so there must be a rule $s \in R[\overline{a}]$ such that $body(s) \subseteq \mathscr{T}_{D,WF}$ and either (2.1) s is strict or else (2.2) for all rules $t \in R[a]$ (including $t = r$), $body(t) \cap \mathscr{U}_{D,WF} \neq \emptyset$. Generalizing on r, for each rule $r \in R_s[a]$, $body(r) \cap (\mathscr{U}_{\Pi,\lambda} \cup \mathscr{U}_{D;WF}) \neq \emptyset$. For each $r \in R_d[a]$, either $body(r) \cap (\mathscr{U}_{\Pi,\lambda} \cup \mathscr{U}_{D;WF}) \neq \emptyset$ or else there exists an $s \in R_s[\overline{a}]$ and $body(s) \subseteq \mathscr{T}_{D,WF}$. Generalizing on a, $\mathscr{U}_{\Pi,\lambda}$ is unfounded *wrt* D and $\mathscr{I}_{D,WF}$, and so, $\mathscr{U}_{\Pi,\lambda} \subseteq U_D(\mathscr{I}_{D,WF})$. Since $\mathscr{I}_{D,WF}$ is a fixpoint of W_D, we have $\mathscr{U}_{\Pi,\lambda} \subseteq \mathscr{U}_{D,WF}$, and hence, $p \in \mathscr{U}_{D,WF}$. □

6 Translating logic programs into defeasible theories

A translation in the other direction is also possible. That is, normal logic programs under the WFS can also be translated into defeasible theories under ADL so that the canonical models of each agree. We show this by first translating the normal logic program into an equivalent extended logic program encoding the closed-world assumption. We then show the equivalence between the ADL theory and the extended program. Below, though both $\sim$ and $\neg$ appear in the extended logic

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

program, we intend the original WFS to be used on the programs—a literal $\neg p$ is simply taken as another atom.

Definition 7
Let Π be a normal program. The *explicit version* of Π is the smallest extended program Φ such that

(1) if $p \leftarrow a_1, \ldots, a_n, \sim b_1, \ldots, \sim b_m$ appears in Π, then $p \leftarrow a_1, \ldots, a_n, \neg b_1, \ldots, \neg b_m$ appears in Φ;
(2) for each $p \in At(\Pi)$, the rule $\neg p \leftarrow \sim p$ appears in Φ.

The following lemmas relate the well-founded models of Π and Φ and make the translation of Π into a defeasible theory D_Π apparent (Definition 8).

Lemma 2
Let Π be a normal program and Φ its explicit version. For any $b \in At(\Pi)$ and any ordinal $\lambda \geqslant 0$,

(1) $\neg b \in \mathscr{T}_{\Phi,\lambda}$ *iff* there exists a $\kappa < \lambda$ such that $b \in \mathscr{U}_{\Phi,\kappa}$.
(2) $\neg b \in \mathscr{U}_{\Phi,\lambda}$ *iff* there exists a $\kappa < \lambda$ such that $b \in \mathscr{T}_{\Phi,\kappa}$.

Proof
(1) Suppose $\neg b \in \mathscr{T}_{\Phi,\lambda}$ for some ordinal λ. Then there exists a least successor ordinal $\kappa \leqslant \lambda$ such that $\neg b \in \mathscr{T}_{\Phi,\kappa}$. As $\neg b \leftarrow\sim b$ is the only rule with head $\neg b$, it must be the case that $b \in \mathscr{U}_{\Phi,\kappa-1}$.
Now suppose there is an ordinal $\kappa < \lambda$ such that $b \in \mathscr{U}_{\Phi,\kappa}$. Since $\neg b \leftarrow\sim b$ is a rule in Φ, it must be the case that $\neg b \in \mathscr{T}_{\Phi,\kappa+1}$. Either $\kappa + 1 = \lambda$, or else by monotonicity of the sequence $(\mathscr{I})$, we have $\neg b \in \mathscr{T}_{\Phi,\lambda}$.
(2) Suppose $\neg b \in \mathscr{U}_{\Phi,\lambda}$. Then there exists a least successor ordinal $\kappa \leqslant \lambda$ such that $\neg b \in \mathscr{U}_{\Phi,\kappa}$. Since $\neg b \leftarrow\sim b$ is the only rule in Φ with head $\neg b$, it must be the case that $b \in \mathscr{T}_{\Phi,\kappa-1}$.
Now suppose there is an ordinal $\kappa < \lambda$ such that $b \in \mathscr{T}_{\Phi,\kappa}$. As $\neg b \leftarrow\sim b$ is the only rule of Φ with head b, it must be the case that $\neg b \in \mathscr{U}_{\Phi,\kappa+1}$. Either $\kappa + 1 = \lambda$, or else by monotonicity we have $\neg b \in \mathscr{U}_{\Phi,\lambda}$. □

It immediately follows from the above Lemma that $\Phi \approx_{WFS} b$ *iff* $\Phi \approx\!\!|_{WFS} \neg b$, and $\Phi \approx_{WFS} \neg b$ *iff* $\Phi \approx\!\!|_{WFS} b$.

Lemma 3
Let Π be a normal program and Φ the explicit version of Π. For any $b \in At(\Pi)$,

(1) $\Pi \mathrel{|}\!\!\approx_{WFS} b$ *iff* $\Phi \mathrel{|}\!\!\approx_{WFS} b$.
(2) $\Pi \approx\!\!|_{WFS} b$ *iff* $\Phi \approx\!\!|_{WFS} b$.

Proof
(LR) The proof is by induction on the sequence $(\mathscr{I}_{\Pi,0}, \mathscr{I}_{\Pi,1}, \ldots)$. Suppose for all $\kappa < \lambda$ and all $p \in At(\Pi)$, if $p \in \mathscr{T}_{\Pi,\kappa}$, then $p \in \mathscr{T}_{\Phi,WF}$; if $p \in \mathscr{U}_{\Pi,\kappa}$ then $p \in \mathscr{U}_{\Phi,WF}$. We may assume *wlog* that λ is a successor ordinal.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

(1) Suppose $p \in \mathcal{T}_{\Pi,\lambda}$. Then there is a rule r with head p such that $body(r)^+ \subseteq \mathcal{T}_{\Pi,\lambda-1}$ and $body(r)^- \subseteq \mathcal{U}_{\Pi,\lambda-1}$. Let r' be the rule of Φ corresponding to r. By the inductive hypothesis, $body(r)^+ \subseteq \mathcal{T}_{\Phi,WF}$ and $body(r)^- \subseteq \mathcal{U}_{\Phi,WF}$. By Lemma 2 for each $q \in body(r)^-$, it follows that $\neg q \in \mathcal{T}_{\Phi,WF}$. Since $body(r)^+ \subseteq \mathcal{T}_{\Phi,WF}$ and for each $q \in body(r)^-$, we have $\neg q \in \mathcal{T}_{\Phi,WF}$, it must be the case that $body(r') \subseteq \mathcal{T}_{\Phi,WF}$. Since r' is strict, then by definition of T_Φ and $\mathcal{I}_{\Phi,WF}$, $p \in \mathcal{T}_{\Phi,WF}$.

(2) Suppose $p \in \mathcal{U}_{\Pi,\lambda}$. Let $q \in At(\Pi)$ be any literal such that $q \in \mathcal{U}_{\Pi,\lambda}$. Then for all $r \in R_\Pi[q]$, there is an $a \in body(r)^+$ such that $a \in \mathcal{U}_{\Pi,\lambda}$ or else a $\sim b \in body(r)^-$ such that $b \in \mathcal{T}_{\Pi,\lambda-1}$. If $b \in \mathcal{T}_{\Pi,\lambda-1}$, then by the inductive hypothesis $b \in \mathcal{T}_{\Phi,WF}$ and so from Lemma 2 $\neg b \in \mathcal{U}_{\Phi,WF}$. Generalizing, for all rules r for q, each corresponding rule r' has a classical literal $a \in body(r')$ such that $a \in \mathcal{U}_{\Pi,\lambda}$ or else a (still classical literal) $\neg b$ such that $\neg b \in \mathcal{U}_{\Phi,WF}$. Generalizing on q, $\mathcal{U}_{\Pi,\lambda}$ is unfounded *wrt* Φ and $\mathcal{I}_{\Phi,WF}$. As such, $\mathcal{U}_{\Pi,\lambda} \subseteq U_\Phi(\mathcal{I}_{\Phi,WF}) = \mathcal{U}_{\Phi,WF}$, and so $p \in \mathcal{U}_{\Phi,WF}$.

(RL) The proof is by induction on the sequence $(\mathcal{I}_{\Phi,0}, \mathcal{I}_{\Phi,1}, \ldots)$. Suppose for all $\kappa < \lambda$ and $p \in At(\Pi)$, if $p \in \mathcal{T}_{\Phi,\kappa}$, then $p \in \mathcal{T}_{\Pi,WF}$; if $p \in \mathcal{U}_{\Phi,\kappa}$, then $p \in \mathcal{U}_{\Pi,WF}$. We may assume *wlog* that λ is a successor ordinal. Let p be any atom of $At(\Pi)$.

(1) Suppose $p \in \mathcal{T}_{\Phi,\lambda}$. Then there is a rule r' with head p such that $body(r') \subseteq \mathcal{T}_{\Phi,\lambda-1}$. By Lemma 2, for each $\neg b \in body(r')$, we have $b \in \mathcal{U}_{\Phi,\eta}$ for some $\eta < \lambda$. Let r be the rule of Π corresponding to r'. By the inductive hypothesis, $body(r)^+ \subseteq \mathcal{T}_{\Pi,WF}$ and $body(r)^- \subseteq \mathcal{U}_{\Pi,WF}$. By definition of T_Π and $\mathcal{I}_{\Pi,WF}$, $p \in \mathcal{T}_{\Pi,WF}$.

(2) Now suppose $p \in \mathcal{U}_{\Phi,\lambda}$ and let q be any literal such that $q \in \mathcal{U}_{\Phi,\lambda}$. If q is a classical negative literal, then no rules for q appear in Π. Suppose q is an atom. Since $q \in \mathcal{U}_{\Phi,\lambda}$, for all rules r' with head q, there is a classical literal $a \in body(r)$ such that $a \in \mathcal{U}_{\Phi,\lambda}$. If a is of the form $\neg b$, then by Lemma 2, $b \in \mathcal{T}_{\Phi,\eta}$ for some $\eta < \lambda$. By the inductive hypothesis, each such b is in $\mathcal{T}_{\Pi,WF}$. Recall that if $\neg b$ appears in the body of r', then $\sim b$ appears in the corresponding rule r of Π. Generalizing on r', for each rule r in Π with head q, there is an $a \in body(r)^+$ such that $a \in \mathcal{U}_{\Phi,\lambda}$, or else a $b \in body(r)^-$ such that $b \in \mathcal{T}_{\Pi,WF}$. Generalizing on q, $\mathcal{U}_{\Phi,\lambda}$ is unfounded *wrt* Π and $\mathcal{I}_{\Pi,WF}$. As such, $\mathcal{U}_{\Phi,\lambda} \subseteq U_\Pi(\mathcal{U}_{\Pi,WF}) = \mathcal{U}_{\Pi,WF}$, and so $p \in \mathcal{U}_{\Pi,WF}$. □

Definition 8

Let Π be a normal logic program. If rule r

$$p \leftarrow a_1, \ldots, a_n, \sim b_1, \ldots, \sim b_m$$

appears in Π, then r_{D_Π} is the rule

$$\{a_1, \ldots, a_n, \neg b_1, \ldots, \neg b_m\} \rightarrow p.$$

Definition 9

If Π is a normal logic program, then the defeasible theory translation D_Π of Π is $\langle Str \cup Pr, C_{MIN}, \emptyset\rangle$, where

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

(1) $Str = \{r_{D_\Pi} | r \in \Pi\}$,
(2) $Pr = \{\emptyset \Rightarrow \neg p | p \in At(\Pi)\}$.

The default literals in the program have become presumptions in the defeasible theory. The rules of the original program are strict in the defeasible theory. It should be obvious that translating D_Π back into a logic program using the Brewka inspired scheme yields Φ. Given the soundness and completeness results of the last section and also Lemma 2, it follows that p is well founded in D_Π under ADL if and only if $\neg p$ is unfounded under ADL, and $\neg p$ is well-founded in ADL if and only if p is unfounded in ADL. Given Lemma 3, the results of D_Π under ADL agree with those of Π *wrt* $At(\Pi)$.

Proposition 3
If Π is a normal logic program, then for any $p \in At(\Pi)$,

(1) $D_\Pi \mathrel{|\!\approx}_{ADL} p$ iff $D_\Pi \mathrel{\approx\!|}_{ADL} \neg p$,
(2) $D_\Pi \mathrel{|\!\approx}_{ADL} \neg p$ iff $D_\Pi \mathrel{\approx\!|}_{ADL} p$.

Proof
Let Φ be the explicit normal form of Π, and suppose $D_\Pi \mathrel{|\!\approx}_{ADL} p$. By Proposition 1, $\Phi \mathrel{|\!\approx}_{WFS} p$. By Lemma 2, $\Phi \mathrel{\approx\!|}_{WFS} \neg p$. By Proposition 2, $D_\Pi \mathrel{\approx\!|}_{ADL} \neg p$. Now suppose $D_\Pi \mathrel{\approx\!|}_{ADL} p$. By Proposition 1, $\Phi \mathrel{\approx\!|}_{WFS} p$. By Lemma 2, $\Phi \mathrel{|\!\approx}_{WFS} \neg p$. By Proposition 2, $D_\Pi \mathrel{|\!\approx}_{ADL} \neg p$. The remaining cases are analogous. □

Proposition 4
If Π is a normal logic program, then for any $p \in At(\Pi)$,

(1) $\Pi \mathrel{|\!\approx}_{WFS} p$ *iff* $D_\Pi \mathrel{|\!\approx}_{ADL} p$,
(2) $\Pi \mathrel{\approx\!|}_{WFS} p$ *iff* $D_\Pi \mathrel{\approx\!|}_{ADL} p$.

Proof
Suppose $D_\Pi \mathrel{|\!\approx}_{ADL} p$. Then $\Phi \mathrel{|\!\approx}_{WFS} p$ by Proposition 1. By Lemma 3, $\Pi \mathrel{|\!\approx}_{WFS} p$. Now suppose $\Pi \mathrel{|\!\approx}_{WFS} p$. Then $\Phi \mathrel{|\!\approx}_{WFS} p$ by Lemma 3. By Proposition 2, $D_\Pi \mathrel{|\!\approx}_{ADL} p$. Again, the remaining cases are analogous. □

Example 4
A logic program Π, its explicit form Φ, and its defeasible logic translation D_Π are shown below.

Π	Φ	D_Π
1. $p \leftarrow\sim q$	1. $p \leftarrow \neg q$	1. $\{\neg q\} \rightarrow p$
2. $q \leftarrow\sim p$	2. $q \leftarrow \neg p$	2. $\{\neg p\} \rightarrow q$
	3. $\neg q \leftarrow\sim q$	3. $\emptyset \Rightarrow \neg q$
	4. $\neg p \leftarrow\sim p$	4. $\emptyset \Rightarrow \neg p$

In the rules of Π, we have replaced each $\sim a$ (where a is an atom) with $\neg a$ and added the rules $\neg a \leftarrow\sim a$. The explicitly negative literals occur nowhere in Π. The well-founded model of both Π and Φ is empty. In D_Π, in order to show $\neg p$, we must first show that $\neg q$ is unfounded, and furthermore, to show $\neg q$, we must first show that $\neg p$ is unfounded. Because of this nothing can be determined in ADL about p, $\neg p$, q, or $\neg q$.

7 Antimonotone operators for ADL and NDL

The account of the WFS provided above is found in Van Gelder *et al.* (1991). It is, however, more typical today to present the WFS in terms of the so-called GL operator γ, which was first defined for the stable model semantics (Gelfond and Lifschitz 1988). In this section, we review the definition of γ and use it to define ambiguity blocking and propagating operators for defeasible theories. With some restrictions, these can be used to calculate the consequences of theories according to ADL and NDL. As shown in the next section, they can also be used to define stable model semantics for defeasible theories.

The GL operator γ works with Herbrand interpretations—sets of ground atoms. If S is a Herbrand interpretation, then atom p is true in S if $p \in S$, and false if $p \notin S$. If Π is a normal logic program and S a Herbrand interpretation, then

$$\gamma_\Pi(S) =_{def} T_{\Pi^S} \uparrow \omega,$$

where Π^S is the so-called *reduct* of Π *wrt* S. Specifically, Π^S is the NAF-free program obtained by

(1) deleting from Π all rules r such that $body(r)^- \cap S \neq \emptyset$, and
(2) deleting all remaining default literals.

For an NAF-free program Π, the immediate consequence operator T_Π reduces to

$$T_\Pi(S) =_{def} \{head(r) | \; r \in \Pi \text{ and } body(r) \subseteq S\}.$$

The sequence $T_\Pi \uparrow 0$, $T_\Pi \uparrow 1, \ldots$, is defined for ordinals $\lambda \geqslant 0$:

(1) $T_\Pi \uparrow 0 = \emptyset$.
(2) $T_\Pi \uparrow \lambda + 1 = T_\Pi(T_\Pi \uparrow \lambda)$ (for successor ordinals $\lambda + 1$).
(3) $T_\Pi \uparrow \lambda = \bigcup_{\kappa<\lambda} T_\Pi \uparrow \kappa$ (for limit ordinals λ).

The set of Herbrand interpretations forms a complete lattice under $\subseteq$, and furthermore, T_Π is continuous on this lattice. As such, $lfp(T_\Pi) = T_P \uparrow \omega$ (van Emden and Kowalski 1976). $T_\Pi \uparrow \omega$ is sometimes written as $Cl(\Pi)$, and so if Π is normal, then $\gamma_\Pi(S) = Cl(\Pi^S)$.

The γ operator is antimonotone, and so γ^2 is monotone. As shown by Baral and Subrahmanian (1993), the well-founded model of Π can be defined in terms of γ_Π^2. Specifically, $wfm(\Pi) = \langle \mathscr{T}, \mathscr{U} \rangle$, where $\mathscr{T} = lfp(\gamma_\Pi^2)$, and $\mathscr{U} = At(\Pi) - \gamma_\Pi(\mathscr{T})$.

Like γ, the ambiguity propagating (α) and blocking (β) operators for defeasible theories are defined using reducts and an immediate consequence operator. β is defined for all defeasible theories, but α is only defined for a restricted class. We consider α first.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Definition 10
Let $D = \langle R, C, \emptyset \rangle$ be a defeasible theory such that $R_u = \emptyset$. If $S \subseteq Lit(D)$, then the α-reduct D_α^S of D *wrt* S is the set of rules $R_s \cup R_d^S$, where

$$R_d^S = \{r | r \in R_d \text{ and } (\forall c \in C[head(r)])(\exists q \in c - \{head(r)\})(q \notin S)\}.$$

Definition 11
Let R be a set of strict and defeasible rules taken from D. If $S \subseteq Lit(D)$, then

$$T_R(S) =_{def} \{p | r \in R \text{ and } body(r) \subseteq S\}.$$

Definition 12
If R is a set of strict and defeasible rules, the sequence $T_R \uparrow 0$, $T_R \uparrow 1, \ldots$ is defined for ordinals $\lambda \geqslant 0$:

(1) $T_R \uparrow 0 = \emptyset$,
(2) $T_R \uparrow \lambda + 1 = T_R(T_R \uparrow \lambda)$ (for successor ordinals $\lambda + 1$),
(3) $T_R \uparrow \lambda = \bigcup_{\kappa < \lambda} T_R \uparrow \kappa$ (for limit ordinals λ).

As with logic programs, where R is a set of (defeasible and strict rules), we define $Cl(R)$ as $T_R \uparrow \omega$.

Definition 13
Let $D = \langle R, C, \emptyset \rangle$ be a defeasible theory such that $R_u = \emptyset$. For any $S \subseteq Lit(D)$, $\alpha_D(S) =_{def} Cl(D_\alpha^S)$.

Under the translation of defeasible theories into logic programs, there is a correspondence between α and γ. In order for the correspondence to hold, $C[p]$ is still required to be finite for each $p \in Lit(D)$, and both R_u and $\prec$ must be empty.

Proposition 5
If $D = \langle R, C, \emptyset \rangle$ is a defeasible theory such that $R_u = \emptyset$ and $C[p]$ is finite for each $p \in Lit(D)$, and if Π is the logic program translation of D, then for any $S \subseteq Lit(D)$,

$$\alpha_D(S) = \gamma_\Pi(S).$$

Proof
The proof proceeds by induction on the simple immediate consequence operator T used to compute the closure of reducts. Note that this operator is continuous and $T \uparrow \omega = \bigcup_{n < \omega} T \uparrow n$, and so it suffices to show that for each $n < \omega$,

$$T_{\Pi^S} \uparrow n = T_{D_\alpha^S} \uparrow n.$$

The claim trivially holds for $n = 0$. Suppose it holds for all $i < n$ and let $p \in T_{D_\alpha^S} \uparrow n$. Then there is an $r \in R_{sd}[p]$ in the reduct D_α^S such that $body(r) \subseteq T_{D_\alpha^S} \uparrow (n-1)$. If r is strict, then there exists an $r' \in \Pi$ such that $r' = r$ (ignoring notational differences; observe that $body(r) = body(r')$). Since r' lacks default literals, $r' \in \Pi^S$. Since $body(r) \subseteq T_{D_\alpha^S} \uparrow (n-1)$, by the inductive hypothesis $body(r) \subseteq T_{\Pi^S} \uparrow (n-1)$, and so by definition of the immediate consequence operator $p \in T_{\Pi^S} \uparrow n$. If r is defeasible, then for all $c \in C[p]$, there exists a $q \in c - \{p\}$ such that $q \notin S$. As such,

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

there exists a rule $r' \in trans(r)$ such that for each $\sim b$ in the body of r' we have $b \notin S$. As this is so, $r' \in \Pi^S$ and every default literal in the body of r' has been deleted, and so, $body(r) = body(r')$. Since $body(r) \subseteq T_{D^S_\alpha} \uparrow (n-1)$, by the inductive hypothesis $body(r) \subseteq T_{\Pi^S} \uparrow (n-1)$. It follows that $p \in T_{\Pi^S} \uparrow n$.

Now suppose $p \in T_{\Pi^S} \uparrow n$. Then there is a rule $t \in \Pi^S$ such that $body(t) \subseteq T_{\Pi^S} \uparrow (n-1)$. If t corresponds to a strict rule of D, then $t \in D^S_\alpha$ and by inductive hypothesis $body(t) \subseteq T_{D^S_\alpha} \uparrow (n-1)$ and so by definition of the immediate consequence operator $p \in T_{D^S_\alpha} \uparrow n$. If t corresponds to a defeasible rule t', since t appears in Π^S, it must be the case that every default literal of t has been deleted. This means that for each conflict set $c \in C[p]$, there is an element $q \in c - \{p\}$ such that $q \notin S$. Since this is so, then by definition of D^S_α, $t' \in D^S_\alpha$. By the inductive hypothesis, $body(t) \subseteq T_{D^S_\alpha} \uparrow (n-1)$ and so as before, $p \in T_{D^S_\alpha} \uparrow n$. □

Earlier, we showed a correspondence between the well-founded model for logic programs and the well-founded model for ADL. This correspondence holds for theories with minimal conflict sets and no defeaters or priorities on rules. Given the correspondence just shown between α and the GL-operator γ, we can now see that α can be used to determine the consequences of ADL for these theories. Note, however, that in Proposition 5 conflict sets need not be minimal. In this way, the operator defines a consequence relation that more generally corresponds to the WFS consequences than does ADL. Returning to Example 3, the defeasible theory's well-founded model according to α coincides with the well-founded model of the corresponding logic program. Both differ from the ADL-consequences of the theory.

As shown in Appendix B, it is possible to transform a defeasible theory into an equivalent one in which defeaters, priorities on rules, and extended conflict sets do not appear. As this is so, α can in fact be used to compute the consequences of theories according to ADL. Nevertheless, this is not as satisfying as having an operator that more naturally corresponds to ADL, and we do not know at this point whether α can be easily modified to serve this purpose.

Unlike α, the blocking operator β places no special restrictions on defeasible theories. As shown below, the alternating fixpoint procedure defined with it can be used to compute the well-founded model according to NDL.

Definition 14
Let $D = \langle R, C, \prec \rangle$ be a defeasible theory and $S \subseteq Lit(D)$. The β-reduct of D *wrt* S (written D^S_β) is $R_s \cup R^S_d$, where R^S_d is the set of rules r such that

- $r \in R_d$, and
- $(\forall c \in C[head(r)])(\exists q \in c - \{head(r)\})(\forall s \in R[q])[body(s) \nsubseteq S$ or $s \prec r]$.

Here, no defeaters are included in the reduct.

Definition 15
If D is a defeasible theory and $S \subseteq Lit(D)$, then $\beta_D(S) =_{def} Cl(D^S_\beta)$.

Definition 16
Let D be a defeasible theory. We define the following sequence:

(1) $X_D \uparrow 0 =_{def} \emptyset$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

(2) $X_D \uparrow \lambda + 1 =_{def} \beta_D^2(X_D \uparrow \lambda)$ (for successor ordinals $\lambda + 1$).
(3) $X_D \uparrow \lambda =_{def} \bigcup_{\kappa<\lambda} X_D \uparrow \kappa$ (for limit ordinals λ).

Below, since we will only use a single theory D, we will omit D as a subscript, writing, e.g., $\beta(S)$ instead of $\beta_D(S)$. Furthermore, we will omit the β when writing the blocking reduct of D wrt S, writing D^S instead of D^S_β. The α-reduct is never used here, and so there will be no confusion.

Proposition 6
Let $\mathscr{I}_{D,WF} = \langle \mathscr{T}_{D,WF}, \mathscr{U}_{D,WF} \rangle$ be the *wfm* of defeasible theory D *wrt* NDL. For all $\lambda \geqslant 0$,

(1) if $p \in X_D \uparrow \lambda$, then $p \in \mathscr{T}_{D,WF}$, and
(2) if $p \notin \beta(X_D \uparrow \lambda)$, then $p \in \mathscr{U}_{D,WF}$.

Proof
By definition of $X \uparrow 0$, $p \notin X \uparrow 0$ for any $p \in Lit(D)$. If $p \notin \beta(X \uparrow 0)$, then it is impossible to derive p from the rules of D under any circumstances, and so p is in $\mathscr{U}_{D,WF}$. Suppose the hypothesis holds for all $\kappa < \lambda$. We prove each case.

(1) Suppose $p \in X \uparrow \lambda$. There must exist a least successor ordinal $\kappa \leqslant \lambda$ such that $p \in X \uparrow \kappa$. Recall that

$$X \uparrow \kappa = \beta(\beta(X \uparrow \kappa - 1)) = Cl(D^{\beta(X\uparrow\kappa-1)}) = T_{D^{\beta(X\uparrow\kappa-1)}} \uparrow \omega.$$

Suppose that for all $i < m$, if $a \in T_{D^{\beta(X\uparrow\kappa-1)}} \uparrow i$, then $a \in \mathscr{T}_{D,WF}$ (this obviously holds for $i = 0$). Let $p \in T_{D^{\beta(X\uparrow\kappa-1)}} \uparrow m$. Then there is an r in $D^{\beta(X\uparrow\kappa-1)}$ such that $body(r) \subseteq T_{D^{\beta(X\uparrow\kappa-1)}} \uparrow (m-1)$. By inductive hypothesis, $body(r) \subseteq \mathscr{T}_{D,WF}$.
If r is defeasible, then since $r \in D^{\beta(X\uparrow\kappa-1)}$, for all conflict sets $c \in C[p]$, there exists a $q \in c - \{p\}$ such that for each rule $s \in R[q]$, either $s \prec r$ or $body(s) \not\subseteq \beta(X \uparrow \kappa - 1)$. If the latter, then by inductive hypothesis, $body(s) \cap \mathscr{U}_{D,WF} \neq \emptyset$. So, there is a rule r of D such that r is strict and $body(r) \subseteq \mathscr{T}_{D,WF}$, or else r is defeasible, $body(r) \subseteq \mathscr{T}_{D,WF}$, and for all conflict sets $c \in C[p]$, there exists a $q \in c - \{p\}$ such that for each rule $s \in R[q]$, either $s \prec r$ or $body(s) \cap \mathscr{U}_{D,WF} \neq \emptyset$. By definition of immediate consequence in NDL and $\mathscr{I}_{D,WF}$, $p \in \mathscr{T}_{D,WF}$.

(2) Now suppose $p \notin \beta(X \uparrow \lambda)$, and let A be the set of elements not in $\beta(X \uparrow \lambda)$. Let r be a rule for p. If r is strict, then r is in the reduct of D relative to $X \uparrow \lambda$ and there is some $q \in body(r)$ such that $q \in A$ (this must be the case since $\beta(X \uparrow \lambda)$ is closed). If r is defeasible, then r is either in the reduct or not. If it is, then as before there is some $q \in body(r)$ such that $q \in A$. If not, then there is a conflict set $c \in C[p]$ such that for all $q \in c - \{p\}$, there is a rule $s \in R[q]$ such that $body(s) \subseteq X \uparrow \lambda$ and $s \not\prec r$. From Case 1, if $body(s) \subseteq X \uparrow \lambda$, then $body(s) \subseteq \mathscr{T}_{D,WF}$. Generalizing on r and then on p, for each $a \in A$ and each $r \in R_{D,sd}[a]$, either there exists a $q \in body(r)$ such that $q \in A$ or else $r \in R_d[a]$ and there exists a conflict set $c \in C[a]$ such that for each $v \in c - \{a\}$, there is a rule $s \in R[v]$ such that $body(s) \subseteq \mathscr{T}_{D,WF}$ and $s \not\prec r$. It can be seen that A is unfounded under NDL with respect to D and $\mathscr{I}_{D,WF}$. As such $A \subseteq U_D(\mathscr{I}_{D,WF}) \subseteq \mathscr{U}_{D,WF}$. Since $p \in A$, $p \in \mathscr{U}_{D,WF}$. □

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Proposition 7

Let D be a defeasible theory and $(\mathscr{I}_D)$ the sequence of interpretations defined for D under the NDL semantics. For any $\lambda \geqslant 0$, there exists a $\eta \geqslant 0$ such that

(1) if $p \in \mathscr{T}_{D,\lambda}$, then $p \in X \uparrow \eta$, and
(2) if $p \in \mathscr{U}_{D,\lambda}$, then $p \notin \beta(X \uparrow \eta)$.

Proof

The hypothesis is trivially satisfied for $\lambda = 0$. Suppose it holds for all ordinals less than λ. We consider each case.

(1) Suppose $p \in \mathscr{T}_\lambda$. Then, one of two cases applies:

(a) There exists a rule $r \in R_s[p]$ such that $body(r) \subseteq \mathscr{T}_\kappa$ for some successor ordinal $\kappa < \lambda$. If that is the case, then by inductive hypothesis, there exists an $\eta \geqslant 0$ such that $body(r) \subseteq X \uparrow \eta$. Since r is strict and $X \uparrow \eta$ is closed under strict rules, $p \in X \uparrow \eta$.

(b) There exists a defeasible rule r such that $body(r) \subseteq \mathscr{T}_\kappa$ for some $\kappa < \lambda$ and for all $c \in C[p]$, there is a $q \in c - \{p\}$ such that for all rules $s \in R[q]$, either $s \prec r$ or else there exists a $v \in body(s)$ such that $v \in \mathscr{U}_\kappa$. If the latter, then by inductive hypothesis, there exists a η such that $v \notin \beta(X \uparrow \eta)$. Since $body(r) \subseteq \mathscr{T}_\kappa$, then by inductive hypothesis, $body(r) \subseteq X \uparrow \imath$ for some ordinal $\imath$. Note that for any ordinals α and γ, if $\alpha < \gamma$, then $X \uparrow \alpha \subseteq X \uparrow \gamma$ and $\beta(X \uparrow \gamma) \subseteq \beta(X \uparrow \alpha)$, and so for any literal b, if $b \notin \beta(X \uparrow \alpha)$, then for all $\gamma > \alpha$ it follows that $b \notin \beta(X \uparrow \gamma)$. With that in mind, generalizing on s and then c, and letting $\imath'$ be the least ordinal such that $\eta < \imath'$ and $\imath < \imath'$ for any of the above $\imath$'s, we have $body(r) \subseteq X \uparrow \imath'$ and for all $c \in C[p]$, there is a $q \in c - \{p\}$ such that for each $s \in R[q]$, $body(s) \nsubseteq \beta(X \uparrow \imath')$ or $s \prec r$. As such $r \in D^{\beta(X \uparrow \imath')}$. Since $body(r) \subseteq X \uparrow \imath'$, by monotonicity, we have $body(r) \subseteq X \uparrow \imath' + 1$. Recall that $X \uparrow \imath' + 1 = \beta(\beta(X \uparrow \imath')) = Cl(D^{\beta(X \uparrow \imath')})$. We thus have $r \in D^{\beta(X \uparrow \imath')}$ and $body(r) \subseteq Cl(D^{\beta(X \uparrow \imath')})$. From this, it follows that $p \in Cl(D^{\beta(X \uparrow \imath')})$, i.e., $p \in X \uparrow \imath' + 1$.

(2) Suppose $p \in \mathscr{U}_\lambda$. If λ is a limit ordinal, then there exists some successor ordinal $\kappa < \lambda$ such that $p \in \mathscr{U}_\kappa$. By inductive hypothesis, there exists some $\eta \geqslant 0$ such that $p \notin \beta(X \uparrow \eta)$. So suppose λ is a successor ordinal. By definition, $\mathscr{U}_\lambda$ is an unfounded set *wrt* to D and $\mathscr{I}_{\lambda-1}$.

Let $a \in \mathscr{U}_\lambda$ and $r \in R_{sd}[a]$. As such, either (1) there is a $v \in body(r)$ such that $v \in \mathscr{U}_\lambda \cup \mathscr{U}_{\lambda-1}$ (which by monotonicity of U_D means $v \in \mathscr{U}_\lambda$) or else (2) $r \in R_d[a]$ and there exists a conflict set $c \in C[a]$ such that for each $q \in c - \{a\}$, there is a rule $s \in R[q]$ such that $body(s) \subseteq \mathscr{T}_{\lambda-1}$ and $s \not\prec r$. Suppose (2) holds. Since $body(s) \subseteq \mathscr{T}_{\lambda-1}$, then by inductive hypothesis, there exists a $\gamma \geqslant 0$ such that $body(s) \subseteq X \uparrow \gamma$. Generalizing on q, there exists a $\eta \geqslant 0$ such that for each $q \in c - \{a\}$, there exists a $s \in R[q]$ such that $body(s) \subseteq X \uparrow \eta$ and $s \not\prec r$. As this is so, by definition of reducts for NDL, $r \notin D^{X \uparrow \eta}$. Thus, if $r \in D^{X \uparrow \eta}$, then $body(r) \cap \mathscr{U}_\lambda \neq \emptyset$. Generalizing on r and then a, we may conclude that for each $v \in \mathscr{U}_\lambda$ and each $r \in R_{sd}[v]$, if $r \in D^{X \uparrow \eta}$, then $body(r) \cap \mathscr{U}_\lambda \neq \emptyset$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Suppose for a proof by contradiction that $\mathscr{U}_\lambda \cap Cl(D_\beta^{X\uparrow\eta}) \neq \emptyset$. Then there is a least integer $i > 0$ such that $\mathscr{U}_\lambda \cap T_{D^{X\uparrow\eta}} \uparrow i \neq \emptyset$. Let $v \in \mathscr{U}_\lambda \cap T_{D^{X\uparrow\eta}} \uparrow i$. Since $v \in T_{D^{X\uparrow\eta}} \uparrow i$, it follows that there exists a rule $r \in R_{sd}[v]$ such that $r \in D^{X\uparrow\eta}$ and $body(r) \subseteq T_{D^{X\uparrow\eta}} \uparrow (i-1)$. However, since $r \in D^{X\uparrow\eta}$, it must be that $body(r) \cap \mathscr{U}_\lambda \neq \emptyset$. Thus, $\mathscr{U}_\lambda \cap T_{D^{X\uparrow\eta}} \uparrow i-1 \neq \emptyset$. This is a contradiction, and so $\mathscr{U}_\lambda \cap Cl(D_\beta^{X\uparrow\eta}) = \emptyset$.

Since $p \in \mathscr{U}_\lambda$ and $Cl(D_\beta^{X\uparrow\eta}) = \beta(X \uparrow \eta + 1)$, it follows that $p \notin \beta(X \uparrow \eta + 1)$.

□

From the above propositions, a correspondence between the sequence $X \uparrow 0$, $X \uparrow 1, \ldots$, and the well-founded model according to NDL is established.

Proposition 8

If D is a defeasible theory, $\mathscr{I}_{D,WF}$ its well-founded model according to NDL, and λ the closure ordinal of the sequence $X_D \uparrow 0$, $X_D \uparrow 1, \ldots$, then

$$\mathscr{I}_{D,WF} = \langle X_D \uparrow \lambda, Lit(D) - \beta_D(X_D \uparrow \lambda)\rangle.$$

8 Stable sets for defeasible theories

NDL and ADL, like the WFS for logic programs, are *directly skeptical* formalisms. If a literal p is a consequence of a theory, then there must be some rule for it with a body that is also a consequence of the theory. This is in contrast to *indirectly skeptical* formalisms, such as default logic (Reiter 1980) and the stable model and answer-set (Gelfond and Lifschitz 1991) semantics for logic programs, where consequences are defined indirectly via the intersection of *extensions* (stable models, answer-sets). These formalisms allow *floating conclusions* (Makinson and Schlecta 1991)—i.e., consequences that appear in every extension but that have no support appearing in every extension. Directly skeptical formalisms do not allow floating conclusions.

Example 5 (Ginsberg's extended Nixon Diamond)

$D = \langle R, C_{MIN} \cup \{\{dove, hawk\}\}, \emptyset\rangle$, where R is

(1) $\emptyset \rightarrow nixon$

(2) $\{nixon\} \rightarrow republican$ (3) $\{nixon\} \rightarrow quaker$

(4) $\{quaker\} \Rightarrow dove$ (5) $\{republican\} \Rightarrow hawk$

(6) $\{hawk\} \rightarrow \neg dove$ (7) $\{dove\} \rightarrow \neg hawk$

(8) $\{hawk\} \Rightarrow extremist$ (9) $\{dove\} \Rightarrow extremist.$

The logic program counterpart to the above theory is

(1) $nixon$

(2) $republican \leftarrow nixon$ (3) $quaker \leftarrow nixon$

(4) $dove \leftarrow \sim \neg dove, \sim hawk, quaker$ (5) $hawk \leftarrow \sim \neg hawk, \sim dove, republican$

(6) $\neg dove \leftarrow hawk$ (7) $\neg hawk \leftarrow dove$

(8) $extremist \leftarrow \sim \neg extremist, hawk$ (9) $extremist \leftarrow \sim \neg extremist, dove.$

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Here, the positive ADL- and NDL-consequences of the theory agree with the well-founded model of the logic program: *nixon*, *republican*, and *quaker* are all well founded, but no other literal is. In ADL, the literals *dove*, *hawk*, and *extremist* are all ambiguous. They are unfounded in NDL.

The logic program has two stable models, where S is a stable model of Π if $\gamma_\Pi(S) = S$:

$$S_1 = \{nixon, republican, quaker, dove, \neg hawk, extremist\}, \text{ and}$$
$$S_2 = \{nixon, republican, quaker, \neg dove, hawk, extremist\}.$$

Since *extremist* appears in each such model, it is taken as a consequence of the program according to the stable model semantics. Since neither *dove* nor *hawk* appears in both models, *extremist* is a floating conclusion.

It is indeed possible to use both α and β to define indirectly skeptical semantics similar to the stable model semantics for logic programs. We do that here. As before, the semantics based on α only applies to a restricted class of defeasible theories.

Definition 17
Let $D = \langle R, C, \prec \rangle$ be a defeasible theory and $S \subseteq Lit(D)$.

(1) If $\prec = \emptyset$ and $R_u = \emptyset$, then S is an *α-stable set* of D iff $S = \alpha_D(S)$.
(2) S is a *β-stable set* of D iff $S = \beta_D(S)$.

Definition 18
Let $D = \langle R, C, \prec \rangle$ be a defeasible theory and $p \in Lit(D)$.

(1) If $\prec = \emptyset$ and $R_u = \emptyset$,
    (a) $D \mid\!\approx_\alpha p$ *iff* $p \in S$ for all α-stable sets S.
    (b) $D \approx\!\mid_\alpha p$ *iff* $p \notin S$ for all α-stable sets S.
(2) For arbitrary theories D,
    (a) $D \mid\!\approx_\beta p$ *iff* $p \in S$ for all β-stable sets S.
    (b) $D \approx\!\mid_\beta p$ *iff* $p \notin S$ for all β-stable sets S.

In Example 5, D has two α-stable sets, and these correspond to S_1 and S_2 above. As such, $D \mid\!\approx_\alpha extremist$. There is only one β-stable set, however,

$$\{nixon, republican, quaker\}.$$

These three literals must appear in any β-stable set. However, this implies that rules 4 and 5 can appear in no β-reduct of D, and so *extremist* can appear in no β-stable set. It is not a floating conclusion according to the semantics based on β.

As in the case for logic programs, the well-founded models according to ADL and NDL, respectively, are contained within the stable sets defined using α and β.

Proposition 9
Let $D = \langle R, C, \prec \rangle$ be a defeasible theory.

(1) If $\prec = \emptyset$, $R_u = \emptyset$, and $\langle \mathscr{T}, \mathscr{U} \rangle$ is the well-founded model of D according to ADL, then for any α-stable set M of D, $\mathscr{T} \subseteq M$, and $\mathscr{U} \cap \alpha_D(M) = \emptyset$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

(2) If $\langle \mathscr{T}, \mathscr{U} \rangle$ is the well-founded model of D according to NDL, then for any β-stable set M of D, $\mathscr{T} \subseteq M$, and $\mathscr{U} \cap \beta_D(M) = \varnothing$.

Proof

The proofs are the same for both ADL and NDL, and so we consider only α_D. Define $X \uparrow 0$, $X \uparrow 1$, etc., as above. Clearly, since $X \uparrow 0 = \varnothing$, $X \uparrow 0 \subseteq M$. Suppose $X \uparrow \kappa \subseteq M$ for each $\kappa < \lambda$. We may assume *wlog* that λ is a successor ordinal. Observe that $X \uparrow \lambda = \alpha_D^2(X \uparrow \lambda - 1)$. By inductive hypothesis, $X \uparrow \lambda - 1 \subseteq M$. Since α_D^2 is monotone, $\alpha_D^2(X \uparrow \lambda - 1) \subseteq \alpha_D^2(M)$. Since $X \uparrow \lambda = \alpha_D^2(X \uparrow \lambda - 1)$ and M is a stable set, it follows that $X \uparrow \lambda \subseteq M$. Generalizing, $\mathscr{T} \subseteq M$. Since α_D is antimonotone, it follows that $\alpha_D(M) \subseteq \alpha_D(\mathscr{T})$, and so $Lit(D) - \alpha_D(\mathscr{T}) \subseteq Lit(D) - \alpha_D(M)$. That is, $\mathscr{U} \subseteq Lit(D) - M$, and so, $\mathscr{U} \cap M = \varnothing$. □

No stable set of a defeasible theory is a subset of another. This parallels the case for the stable models/answer-sets of logic programs (Gelfond and Lifschitz 1991) and is the result of α and β being antimonotone. For instance, if S_1 and S_2 are α-stable sets such that $S_1 \subseteq S_2$, then $\alpha_D(S_2) \subseteq \alpha_D(S_1)$, and so $S_2 \subseteq S_1$.

Proposition 10

If S_1 and S_2 are distinct α (β) stable sets of defeasible theory D, then $S_1 \nsubseteq S_2$.

Given the close connection between α and γ, for each defeasible theory D with no defeaters and priorities, then provided that $C[p]$ is finite for each $p \in Lit(D)$, we may conclude that the α-stable sets of D correspond to the stable models of its logic program translation. Furthermore, as shown below (Propositions 11–13), a correspondence for the translation in the reverse direction also holds. That is, if Π is a normal logic program, then the stable models of Π correspond to the α-stable sets of D_Π.

Observe that this implies that any stable model of Φ is a classical interpretation (and so can be represented as a set of atoms). Below, if $X \subseteq At(\Pi)$, let

$$X^{\neg} = X \cup \{\neg p | p \in At(\Pi) \text{ and } p \notin X\}.$$

Proposition 11

Let Π be a normal logic program and Φ the explicit version of Π. For any $M \subseteq At(\Pi)$, $T_{\Pi^M} \uparrow \omega = (T_{\Phi^{M^{\neg}}} \uparrow \omega \cap At(\Pi))$.

Proof

We show that for all $i \geqslant 0$ and $p \in At(\Pi)$, $p \in T_{\Pi^M} \uparrow i$ implies $p \in T_{\Phi^{M^{\neg}}} \uparrow \omega$, and $p \in T_{\Phi^{M^{\neg}}} \uparrow i$ implies $p \in T_{\Pi^M} \uparrow \omega$. The case for $i = 0$ is vacuous. Suppose the claim holds for all $i < n$.

If $p \in T_{\Pi^M} \uparrow n$, then there is a rule $r \in \Pi$ such that $body(r)^+ \subseteq T_{\Pi^M} \uparrow (n-1)$ and $q \notin M$ for each $q \in body(r)^-$. Let $r' \in \Phi$ be the strict rule corresponding to r. By inductive hypothesis, $body(r)^+ \subseteq T_{\Phi^{M^{\neg}}} \uparrow \omega$. For each q, $\neg q \leftarrow \varnothing \in \Phi^{M^{\neg}}$ and so $\neg q \in T_{\Phi^{M^{\neg}}} \uparrow \omega$. It follows that $body(r') \subseteq T_{\Phi^{M^{\neg}}} \uparrow \omega$, and so $p \in T_{\Phi^{M^{\neg}}} \uparrow \omega$.

If $p \in T_{\Phi^{M^{\neg}}} \uparrow n$, then there is a rule $r \in \Phi^{M^{\neg}}$ such that $body(r) \subseteq T_{\Phi^{M^{\neg}}} \uparrow (n-1)$. By inductive hypothesis, $a \in T_{\Pi^M} \uparrow \omega$ for each atom $a \in body(r)$. For each $\neg q \in body(r)$, it must be that $\neg q \leftarrow \varnothing \in \Phi^{M^{\neg}}$, and so $q \notin M^{\neg}$ and $q \notin M$. Rule r corresponds

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

to a rule $r' \in \Pi$ such that atom $a \in body(r)$ *iff* $a \in body(r')^+$, and $\neg q \in body(r)$ *iff* $q \in body(r')^-$. Given that no $q \in body(r')^-$ appears in M, $p \leftarrow body(r')^+ \in \Pi^M$. Since $body(r')^+ \subseteq T_{\Pi^M} \uparrow \omega$, it follows that $p \in T_{\Pi^M} \uparrow \omega$. □

Proposition 12

Let Π be a normal logic program and Φ its explicit version. $M \subseteq At(\Pi)$ is a stable model of Π *iff* $M^\neg$ is a stable model of Φ.

Proof

$M^\neg \cap At(\Pi) = M$, and from Proposition 11, $T_{\Pi^M} \uparrow \omega = (T_{\Phi^{M^\neg}} \uparrow \omega \cap At(\Pi))$. If M is a stable model of Π, $M = (T_{\Phi^{M^\neg}} \uparrow \omega \cap At(\Pi))$, and so for each $p \in At(\Pi)$, $p \in M^\neg$ *iff* $p \in T_{\Phi^{M^\neg}} \uparrow \omega$. If $\neg p \in T_{\Phi^{M^\neg}} \uparrow \omega$, then $\neg p \leftarrow \emptyset \in \Phi^{M^\neg}$ and so $p \notin M^\neg$. If that is so, then $p \notin M$ and (by definition of $M^\neg$) $\neg p \in M^\neg$. Conversely, if $\neg p \in M^\neg$, then $p \notin M$, and so $\neg p \leftarrow \emptyset \in \Phi^{M^\neg}$. Consequently, $\neg p \in T_{\Phi^{M^\neg}} \uparrow \omega$. As such, $M^\neg$ is a stable model of Φ. If, in turn, $M^\neg$ is a stable model of Φ, $T_{\Pi^M} \uparrow \omega = (M^\neg \cap At(\Pi)) = M$, and so M is a stable model of Π. □

Proposition 13

Let Π be a normal logic program and D_Π its defeasible logic translation. M is a stable model of Π *iff* $M^\neg$ is an α-stable set of D_Π.

Proof

Translating D_Π into a logic program using the Brewka-inspired scheme yields Φ. As implied by Proposition 5, the α-stable sets of D_Π correspond to the stable models of Φ. However, by Proposition 12, there is a one-to-one correspondence between the stable models of Φ and those of Π. □

Example 6

The defeasible theory from Example 2 and its logic program translation are shown again below.

(1) $\emptyset \Rightarrow p$	(1) $p \leftarrow\!\sim \neg p$
(2) $\emptyset \Rightarrow \neg p$	(2) $\neg p \leftarrow\!\sim p$
(3) $\{p\} \Rightarrow \neg q$	(3) $\neg q \leftarrow\!\sim q, p$
(4) $\emptyset \Rightarrow q$	(4) $q \leftarrow\!\sim \neg q.$

The α-stable sets of the defeasible theory are $\{p, q\}$, $\{p, \neg q\}$, and $\{\neg p, q\}$. These are also the stable models of the counterpart logic program. The only β-stable set is $\{q\}$, however. Neither p nor $\neg p$ can appear in any stable set (the rules for them would be deleted in any β-reduct), and this implies that $\neg q$ cannot appear, either.

As reported earlier (Maier 2010), ADL is more conservative than NDL, in the sense that for all D and p, if $D \mid\!\approx_{ADL} p$, then $D \mid\!\approx_{NDL} p$. From this, it readily follows that $D \mid\!\approx_{ADL} p$ implies $D \mid\!\approx_\beta p$. However, the similar claim does not hold if $\mid\!\approx_\alpha$ is used in place of $\mid\!\approx_{ADL}$. That is, $D \mid\!\approx_\alpha p$ does not imply $D \mid\!\approx_\beta p$.

Example 7

$D = \langle R, C_{MIN}, \emptyset \rangle$, where R is

(1) $\emptyset \Rightarrow p$

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

(2) $\emptyset \Rightarrow \neg p$
(3) $\{p\} \Rightarrow q$
(4) $\{\neg p\} \Rightarrow q$.

Here, the α-stable sets are $\{p, q\}$ and $\{\neg p, q\}$, and so $D \mid\!\approx_\alpha q$. However, the only β-stable set is $\emptyset$, which implies $D \approx\!\mid_\beta q$.

ADL is also more conservative than NDL in the sense that $D \approx\!\mid_{ADL} p$ implies $D \approx\!\mid_{NDL} p$. We do not know yet whether $D \approx\!\mid_\alpha p$ implies $D \approx\!\mid_\beta p$.

9 Related work

As stated above, NDL (Donnelly 1999; Nute 1999; Nute 2003) and ADL (Maier and Nute 2006) were the first defeasible logics to incorporate failure-by-looping, and this cycle check essentially requires the proof systems to be tree-based—different branches of computation must be kept distinct. In BDL (Billington 1993) and most other variants of defeasible logic, proofs are linear sequences of tagged literals. In these logics, cycles cannot be detected, and this affects the conclusions they can draw. Maher and Governatori (1999), however, do provide a WFS for BDL that correctly handles cycles. Presumably, the BDL proof system is sound but not complete relative to this semantics.

The logics based on BDL also differ from NDL and ADL in that they make a distinction between *strict* and *defeasible* derivations. For example, the expression $+\Delta p$ in a derivation indicates that p is derivable using only the strict rules of a theory, while $+\delta p$ means that p is derivable using the theory as a whole (the corresponding negative expressions $-\Delta p$ and $-\delta p$ indicate that p is refutable). Significantly, if the body of a strict rule r is only defeasibly derivable, then the rule is treated as a defeasible rule, i.e., a rule that can be defeated. This prevents BDL from inferring contradictions except for those due to strict rules alone.

Example 8
$D = \langle R, C_{MIN}, \emptyset \rangle$, R is

$$(1)\ \emptyset \Rightarrow \mathit{married}, \quad (2)\ \{\mathit{married}\} \rightarrow \neg \mathit{bachelor}, \quad (3)\ \emptyset \Rightarrow \mathit{bachelor}.$$

In BDL and its variants, *married* and *bachelor* do not conflict, and so *married* is defeasibly derivable (there is a proof ending in $+\delta \mathit{married}$). However, since the body of rule 2 is only defeasibly derivable, rule 2 is considered defeasible. Since BDL blocks ambiguity, in that logic both *bachelor* and $\neg$*bachelor* are defeasibly refuted (in the ambiguity propagating logic described by Antoniou *et al.* 2000b, both literals are ambiguous). In contrast, if conflict sets are closed under strict rules, then NDL and ADL hold that *married* and *bachelor* conflict and refrain from deriving either (they are refuted in NDL and ambiguous in ADL). Antoniou (2006) calls the approach taken in ADL and NDL the “purist view,” and he defends the alternative. Brewka (2001) rejects the dual treatment of strict rules, however: Strict rules are used to specify definitions, necessary relationships, etc. To treat them sometimes as defeasible rules undermines this. Essentially the same argument was made when the semantics for ADL and NDL was first developed (Maier 2010). Extended conflict

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

sets were introduced in NDL and ADL to avoid drawing inconsistent conclusions based on defeasible rules while at the same time maintaining the monolithic nature of strict rules.

Ambiguity propagation appeared in defeasible logic around the year 2000 (Antoniou *et al.* 2000a). Up to that point, all defeasible logics were ambiguity blocking. In Antoniou *et al.* (2000a), the basic propagating logic—based on BDL—is presented as a system embedded in a logic program. A formal proof system appeared separately (Antoniou *et al.* 2000b; Antoniou and Billington 2001).

BDL is itself a modification of an earlier defeasible logic (Nute *et al.* 1989; Billington *et al.* 1990). To this logic, BDL adds variables and function symbols, and, importantly, it allows the precedence relation to range over both strict and defeasible rules. In the earlier logic (as in NDL and ADL), strict rules are superior to all defeasible rules and no strict rule is superior to any other strict rule. In his analysis, Brewka (2001) shows that when the precedence relation is restricted to defeasible rules, the defeasible logic is sound but not complete *wrt* his prioritized WFS.

In a separate line of work, Billington has developed a family of formalisms that are generally called *plausible logic* (Billington and Rock 2001; Billington 2004, 2005a, 2005b, 2008). Plausible logic is based on defeasible logic, using both strict and defeasible rules, but it expands it to handle arbitrary clauses. Unlike in defeasible logic, disjunctions can be proved. Extended conflict sets are not used, but the logics have what is called the *general conflict property* (Billington 2008), meaning that defeasible rules conflict if they cannot all fire without contradicting the strict part of the theory. Proofs are again sequences of tagged formulas, and these tags are used to define multiple consequence relations (which correspond to different levels of certainty). Through the use of tags, the proof system simultaneously allows both the blocking and propagation of ambiguity. Loop detection is discussed in Billington (2004) and Billington (2008). Given the number of NMR formalisms in existence today and the differing intuitions they embody, a formalism such as plausible logic—which attempts to unify these intuitions into a single system—appears very attractive.

In Billington (2007), multiple semantics for plausible theories are provided, corresponding to differing intuitions about acceptable consequences. Plausible theories are related to default theories (Reiter 1980), and it is shown how the framework can provide an ambiguity blocking semantics for default logic. Given the known relationships between default logic and the stable model semantics for logic programs (Marek and Truszczynski 1989), the work in Billington (2007) can be seen as applying to logic programs.

Other variants of defeasible logic have been related to different NMR formalisms. A Dung-like argumentation semantics for BDL and its variants is provided in Governatori *et al.* (2000) and Governatori *et al.* (2004). The relationship between defeasible logic (again, an ambiguity propagating variant of BDL) and default logic is addressed in Antoniou and Billington (2001). A means of translating defeasible theories into default theories is given, and it is shown that every defeasible consequence appears in every extension of the corresponding default theory. The

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

paper does not address refutations—i.e., it is not proven whether a literal defeasibly refuted is absent from every default extension.

The logic-programming embedding used by Antoniou *et al.* first appeared in Maher and Governatori (1999). It is shown there that the BDL-consequences of a defeasible theory correspond to those of the counterpart program under the Kunen semantics (Kunen 1987). The same paper presents the WFS for BDL mentioned above and shows that the consequences under this semantics correspond to the well-founded model of the program. In Antoniou *et al.* (2006), it is shown that under the translation, the conclusions of the defeasible theory correspond to the intersection of stable models of the program. This result holds only for what the authors call *decisive* theories—theories in which every literal is either provable or refutable (or, equivalently, theories whose dependency graph is acyclic). Without decisiveness, the correspondence holds only in one direction: every literal provable in the defeasible logic appears in the intersection of stable models.

We note that the translation used by Antoniou *et al.* is not at all like the Brewka-inspired scheme described above, and in our opinion, it does not by itself expose a close relationship between defeasible logic and logic programming. In their method of translation, the defeasible logic proof system is explicitly encoded in the logic program. For example, the proof-conditions governing strict derivations are represented (in Prolog notation) as

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

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

A statement X is definitely (strictly) derivable ($+\Delta X$) if X is a fact of the theory, or if there is a strict rule with head X and every literal of the body is also definitely derivable. The rules of a defeasible theory are represented as facts in the logic program. For example,

```
strict(rule1, bird(a), [swan(a)]).
defeasible(rule2, white(a), [swan(a)]).
```

In this fashion, the logic program encodes both the defeasible theory (as terms appearing in facts and rules) and the proof system itself. In the Brewka-inspired scheme, it is only the defeasible theory that is translated and not the entirety of the proof system. Because of this, we consider the relationships between ADL and the WFS, proven above, to be more insightful.

The fixpoint semantics for NDL and ADL appear in Maier (2010), and it is shown there that the proof systems for NDL and ADL are sound with respect to their counterpart semantics and that they are complete for locally finite theories. It is also shown there that, when the priorities on rules are transitive, ADL and NDL satisfy versions of Cut and Cautious Monotony (that is, they are cumulative). It is widely accepted that a good nonmonotonic formalism should satisfy these.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Defeasible theories, such as the one shown in Example 3, are problematic for both ADL and NDL. In that example, q is not well founded in either NDL or ADL, but it intuitively should be (in the corresponding logic program, q is indeed well founded). Examples such as this show that, while extended conflict sets are needed in some cases to draw reasonable conclusions, their use can cause problems in other cases. An alternative to using extended conflict sets is to keep conflict sets minimal while adding all possible transpositions of strict rules to the defeasible theory. If this is done, then the intuitively correct result can be drawn in Example 3. Doing this (or else closing conflict sets under strict rules), allows NDL and ADL to satisfy Consistency Preservation. That is, the logics cannot be used to derive contradictions that do not follow from the strict rules alone. This is shown in Maier (2010).

10 Conclusion

Nute's logic NDL was developed in isolation of the WFS, but the desire to handle theories containing cycles appears to be the same. While it is unsurprising that the consequences under NDL do not correspond to those of the WFS—NDL blocks ambiguity while the WFS propagates it—we have shown here that under natural translations of defeasible theories into logic programs (and vice versa), the consequences according to ADL and the WFS actually do coincide. This, in a sense, is surprising, as ADL was developed by making only a minor modification to the proof system of NDL.

The present research was initiated with an eye toward practicality. The ability to translate defeasible theories into logic programs means that existing logic programming systems can be used to reason according to ADL. In the other direction, NDL indirectly provides an ambiguity blocking semantics for logic programs, and ADL provides a representation of logic programs under the WFS that in some cases is intuitively easier to comprehend (this is an arguable point; nevertheless, we suppose that some at least will find $\Rightarrow$ more readily understood than default negation).

The antimonotone operator defined for ADL only works properly when defeaters are not present in the defeasible theory and when the priority relation over rules is empty. Both defeaters and priorities can in fact be compiled way, however. That is, a defeasible theory D of ADL or NDL can be transformed into an equivalent one E such that $R_u = \emptyset$, $\prec = \emptyset$, and C is minimal. This is shown in Appendix B. A similar transformation is discussed in the context of other defeasible logics in Antoniou *et al.* (2001). Nevertheless, while the elimination of priorities and defeaters allows the use of α to compute all of the ADL consequences of a theory, it is not a very satisfying solution, as it requires expressing important elements of the logic (e.g., conflicts, priorities) directly in the rules of the theory. In that sense, the transformation is similar to the embedding noted above of BDL into logic programs. An operator that does not require any sort of transformation in order to do its work would be far better.

Similar work on adding priorities to the WFS has been performed, notably by Brewka (1996), and also by Schaub and Wang (2002). Both have developed

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

prioritized WFS for extended logic programs, and in both cases, the models can be computed in polynomial time relative to the size of the program. At this point, we do not know how ADL relates to these formalisms, and we have not investigated whether their way of handling preferences can be easily adopted for use with ADL (or other defeasible logics). It is certainly the case, however, that the two logic programming formalisms yield results different than ADL, for the simple reason that both formalisms are explosive. For example, in both formalisms, the well-founded model of the program

(1) p
(2) $\neg p$
(3) $q \leftarrow r, s, t$

is the set of all literals. In contrast, q would be considered unfounded according to ADL. In our view, this is the correct conclusion, as we really have no reason to believe q. Other varieties of defeasible logic would similarly consider q unfounded; none would conclude q. By their nature, defeasible logics are paraconsistent.

Acknowledgements

Portions of this work were performed while the author was a doctoral student at the University of Georgia in Athens, Georgia.

Appendix A. NDL and ADL proof systems

Proofs in NDL and ADL form argument trees, with nodes labeled with tagged literals (for a given node n, $label(n)$ refers to the label of n). In earlier defeasible logics, such as BDL, proofs are linear sequences of tagged literals.

Definition 19
Let D be a defeasible theory. A *defeasible argument tree for* D is a finite tree τ such that every node of τ is labeled with one of $+p$ or $-p$, where p is any literal in $Lit(D)$. If τ is a defeasible argument tree for D and n is a node in τ, then τ is a *positive* node iff n is labeled $+p$, and n is a *negative* node iff n is labeled $-p$.

Definition 20
Let A be a set of literals, and n be a node of a defeasible argument tree τ.

(1) A *succeeds* at n iff for all $q \in A$, there is a child of n labeled $+q$.
(2) A *fails* at n iff there is a $q \in A$ and a child of n labeled $-q$.

A tree over D with root $+p$ indicates that p is defeasibly derivable from D; a tree over D with root $-p$ indicates that p is defeasibly refuted. In order to count as a valid proof in NDL or ADL, the nodes of the tree must satisfy certain conditions. We discuss the conditions for NDL first.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Definition 21
An argument tree τ over defeasible theory D is an NDL-*proof* for D iff for each node n of τ, one of the following obtains:

(1) $label(n) = +p$ and either

(a) there is an $r \in R_s[p]$ such that $body(r)$ succeeds at n, or
(b) there is an $r \in R_d[p]$ such that

(i) $body(r)$ succeeds at n, and
(ii.) for all $c \in C[p]$ there is a $q \in c - \{p\}$ such that for all $s \in R[q]$, either $body(s)$ fails at n or else $s \prec r$.

(2) $label(n) = -p$ and

(a) for all $r \in R_s[p]$, $body(r)$ fails at n, and
(b) for all $r \in R_d[p]$, either

(i) $body(r)$ fails at n, or
(ii) there is a $c \in C[p]$ such that for all $q \in c - \{p\}$, there is a $s \in R[q]$ such that $body(s)$ succeeds at n and $s \not\prec r$.

(3) $label(n) = -p$ and n has an ancestor m in τ with $label(m) = -p$, and all nodes between n and m are negative.

Definition 22
Let D be a defeasible theory and τ an NDL-proof for D.

(1) τ is an NDL-*proof of* p *in* D iff τ is an NDL-proof for D, $p \in Lit(D)$, and the root node of τ is labeled $+p$. If such a proof exists, then $D \mid\!\sim_{NDL} p$.
(2) τ is an NDL-*refutation of* p *in* D iff τ is an NDL-proof for D, $p \in Lit(D)$, and the root node of τ is labeled $-p$. If such a refutation exists, then $D \sim\!\mid_{NDL} p$.

The third condition in Definition 21 is called *failure-by-looping*, and it prevents a literal from being derived using a circular argument. According to the condition, the nodes between n and m must all be negative. This ensures that literals are not simultaneously provable and refutable. It is failure-by-looping that requires the proofs to be trees rather than linear sequences of literals.

NDL is an ambiguity blocking logic. Returning to Example 2, the conclusions are $D \sim\!\mid_{NDL} p$, $D \sim\!\mid_{NDL} \neg p$, $D \sim\!\mid_{NDL} \neg q$, and $D \mid\!\sim_{NDL} q$. NDL can be modified, however, in a simple way to make it propagate ambiguity—yielding ADL. In ADL, a defeasible rule r can only be defeated by a conflicting set of rules that are strict or else of higher priority (in NDL, rules simply not inferior to r can be used). The modification to the proof system is shown in Definition 23. Proofs and refutations in ADL are otherwise defined as they are in NDL.

Definition 23
An argument tree τ for D is an ADL-*proof for* D iff each node n of τ satisfies conditions 1, 2a, 2b(i), or 3 of Definition 21, or else the modified condition 2b(ii) below:

2b(ii) there is a $c \in C[p]$ such that for all $q \in c - \{p\}$, there is a $s \in R[q]$ such that $body(s)$ succeeds at n and s is strict or else $r \prec s$.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Proofs in ADL and NDL are finite trees and so must work with finite sets of literals. The fixpoint semantics for NDL and ADL can work with infinite sets, however, and because of this the proof systems cannot be complete with respect to their counterpart semantics. Nevertheless, the proof systems are sound with respect to the semantics, and for *locally finite theories*, they are also complete.

Definition 24
Let D be a defeasible theory and $p \in Lit(D)$.

(1) $Dep_D(p)$ is the smallest set such that (i) $p \in Dep_D(p)$; (ii) for each $q \in Dep_D(p)$, if $c \in C[q]$, then $c \subseteq Dep_D(p)$; and (iii) for each $q \in Dep_D(p)$, if $r \in R[q]$, then $body(r) \subseteq Dep_D(p)$.
(2) Literal p is *locally finite in D* iff $Dep_D(p)$ is finite.
(3) D is *locally finite* iff each literal of $Lit(D)$ is locally finite in D.

Proposition 14
(Maier 2010) If D is a defeasible theory, $p \in Lit(D)$, and L one of NDL or ADL,

(1) $D \mathrel{|\!\!\sim}_L p$ implies $D \mathrel{|\!\!\approx}_L p$, and $D \mathrel{\sim\!\!|}_L p$ implies $D \mathrel{\approx\!\!|}_L p$.
(2) If p is locally finite in D, $D \mathrel{|\!\!\approx}_L p$ implies $D \mathrel{|\!\!\sim}_L p$ and $D \mathrel{\approx\!\!|}_L p$ implies $D \mathrel{\sim\!\!|}_L p$.

Appendix B. Eliminating priorities and defeaters

If $D = \langle R_D, C_D, \prec_D \rangle$ is a defeasible theory such that $C_D[p]$ is finite for all $p \in Lit(D)$, then D can be translated into an equivalent theory $E = \langle R_E, C_E, \prec_E \rangle$, called the *defeater- and priority-free form* of D, in which $R_{E,u} = \emptyset$, $\prec_E = \emptyset$, and C_E is minimal. Specifically, E is the smallest theory such that the following hold (in the following, $A \dashrightarrow p$ stands for an arbitrary rule):

(1) If rule $r\colon A \dashrightarrow p$ is in R_D, the rules

$r'\colon\ A \rightarrow su(r)$
$r''\colon\ \{su(r)\} \dashrightarrow fi(r)$

appear in R_E, with r'' strict (defeasible) if r is strict (defeasible or a defeater).

(2) If $r\colon A \dashrightarrow p$ is strict or defeasible, the following also occurs in R_E:

$r'''\colon\ \{fi(r)\} \rightarrow p.$

(3) Let $c = \{q_1, \ldots, q_n, p\} \in C_D[p]$ be a conflict set, $r \in R_{D,d}[p]$ and $s_1, \ldots, s_n$ rules such that $s_i \in R_D[q_i]$ and $s_i \not\prec_D r$. The following rule appears in R_E:

$$\{su(s_1), \ldots, su(s_n)\} \dashrightarrow \neg fi(r).$$

It is strict if $s_i \in R_s$ or $r \prec_D s_i$ for all s_i, and defeasible otherwise.

The rules of E explicitly encode when a rule r of D is supported and when it may fire. In item 3, only rules s_i that could defeat r are used. For any conflict set $c \in C_D[head(r)]$, let $trans(c, r)$ denote the set of rules for $\neg fi(r)$ created from c. Importantly, the conflict sets of E are minimal. If D itself uses minimal conflict sets, then if $r \in R_{D,sd}[p]$ and $s \in R[\neg p]$, condition 3 above reduces to $\{su(s)\} \rightarrow \neg fi(r)$ if s is strict or $r \prec_D s$, and to $\{su(s)\} \Rightarrow \neg fi(r)$ if $s \not\prec_D r$.

Proposition 15

Let D be a defeasible theory such that for all $p \in Lit(D)$, $C_D[p]$ is finite, and let E be the defeater- and priority-free form of D. If $p \in \mathcal{T}_{D,WF}$ ($p \in \mathcal{U}_{D,WF}$), then $p \in \mathcal{T}_{E,WF}$ ($p \in \mathcal{U}_{E,WF}$).

Proof

The proof proceeds by induction on the sequence $\mathcal{I}_{D,0}, \mathcal{I}_{D,1}, \ldots$ and shows that for all $\kappa \geqslant 0$, if $p \in \mathcal{T}_{D,\kappa}$ then $p \in \mathcal{T}_{E,WF}$, and if $p \in \mathcal{U}_{D,\kappa}$ then $p \in \mathcal{U}_{E,WF}$. The claim holds trivially for $\kappa = 0$. Suppose that it holds for all $\kappa < \lambda$. We can assume without loss of generality that λ is a successor ordinal.

(1) Suppose $p \in \mathcal{T}_{D,\lambda}$. Then there is an $r \in R_D[p]$ such that $body(r) \subseteq \mathcal{T}_{D,\lambda-1}$ and either (1) $r \in R_{D,s}$ or else (2) $r \in R_{D,d}$ and for each $c \in C_D[p]$, there is a $q \in c - \{p\}$ and for all $s \in R_D[q]$, $body(s) \cap \mathcal{U}_{D,\lambda-1} \neq \emptyset$ or $s <_D r$. By the inductive hypothesis, $body(r) \subseteq \mathcal{T}_{E,WF}$, and so $su(r) \in \mathcal{T}_{E,WF}$. If (1) holds, r'' is strict, and so $p \in \mathcal{T}_{E,WF}$.
So suppose (2) obtains. Then r'' is defeasible. By the inductive hypothesis, for every $c \in C_D[p]$, there is a $q \in c - \{p\}$ such that for every rule $s \in R_D[q]$, either (i) $body(s) \cap \mathcal{U}_{E,WF} \neq \emptyset$, or else (ii) $s <_D r$. In other words, if $s \not<_D r$ then $body(s) \cap \mathcal{U}_{E,WF} \neq \emptyset$ and so $su(s) \in \mathcal{U}_{E,WF}$. As such, for every rule $t \in trans(c, r)$, $body(t) \cap \mathcal{U}_{E,WF} \neq \emptyset$. Generalizing on c, for every rule $t \in R_E[\neg fi(r)]$, $body(t) \cap \mathcal{U}_{E,WF} \neq \emptyset$. By definition of T_E and $\mathcal{I}_{E,WF}$, both $fi(r) \in \mathcal{T}_{E,WF}$ and $p \in \mathcal{T}_{E,WF}$.

(2) Now suppose that $p \in \mathcal{U}_{D,\lambda}$. Let $X = \mathcal{U}_{D,\lambda} \cup \{fi(r) | \ r \in R_D[p] \text{ and } p \in \mathcal{U}_{D,\lambda}\} \cup \{su(r), fi(r) | \ body(r) \cap \mathcal{U}_{D,\lambda} \neq \emptyset\}$. We will show that X is unfounded *wrt* E and $\mathcal{I}_{E,WF}$. Note that there are three types of literal in X: those in $\mathcal{U}_{D,\lambda}$, and those of the form $fi(r)$ or $su(r)$. Regarding the first two types, if $p \in \mathcal{U}_{D,\lambda}$, then (by definition of X) $fi(r) \in X$ for all rules $\{fi(r)\} \rightarrow p \in R_E[p]$. This exhausts all rules in E for p. If $su(r) \in X$, then by definition of X, $body(r) \cap \mathcal{U}_{D,\lambda} \neq \emptyset$.
Regarding the third type, suppose $fi(r) \in X$. Recall $r \in R_D[p]$ for some p, and r' is the only rule in E for $fi(r)$. If $body(r) \cap \mathcal{U}_{D,\lambda} \neq \emptyset$, then $su(r) \in X$. If $body(r) \cap \mathcal{U}_{D,\lambda} = \emptyset$, then (by definition of X) $p \in \mathcal{U}_{D,\lambda}$. Since $p \in \mathcal{U}_{D,\lambda}$ and $body(r) \cap \mathcal{U}_{D,\lambda} = \emptyset$, r (and r') must be defeasible, and there must be a $c \in C_D[p]$ such that for all $q_i \in c - \{p\}$, there is a rule $s_i \in R_D[q_i]$ such that $body(s_i) \in \mathcal{T}_{D,\lambda-1}$ and $r <_D s_i$ or s_i is strict (for NDL, $s_i \not<_D r$). This implies that the rule $t : \{su(s_1), su(s_2), \ldots, su(s_\lambda)\} \rightarrow \neg fi(r)$ appears in E (for NDL t is defeasible). By the inductive hypothesis, for each s_i, $body(s_i) \in \mathcal{T}_{E,WF}$, and so $su(s_i) \in \mathcal{T}_{E,WF}$.
X is thus unfounded *wrt* E and $\mathcal{I}_{E,WF}$ (i.e., $X \subseteq \mathcal{U}_{E,WF}$), and so $p \in \mathcal{U}_{E,WF}$.

□

Proposition 16

Let D be a defeasible theory such that for all $p \in Lit(D)$, $C_D[p]$ is finite, and let E be the defeater- and priority-free form of D. If $p \in \mathcal{T}_{E,WF}$ ($p \in \mathcal{U}_{E,WF}$), then $p \in \mathcal{T}_{D,WF}$ ($p \in \mathcal{U}_{D,WF}$).

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Proof

The proof proceeds by induction on the sequence $\mathscr{I}_{E,0}$, $\mathscr{I}_{E,1}$, … and shows that for all $\kappa \geqslant 0$, if $p \in \mathscr{T}_{E,\kappa}$ then $p \in \mathscr{T}_{D,WF}$, and if $p \in \mathscr{U}_{E,\kappa}$ then $p \in \mathscr{U}_{D,WF}$. The claim holds trivially for $\kappa = 0$. Suppose that it holds for all $\kappa < \lambda$. We can again assume without loss of generality that λ is a successor ordinal.

(1) Suppose $p \in \mathscr{T}_{E,\lambda}$. Then there is an $r \in R_D[p]$, matching rules $r', r'', r''' \in R_E$, and a least $\eta < \lambda$ such that $\{fi(r), su(r)\} \cup body(r) \subseteq \mathscr{T}_{E,\eta}$. By the inductive hypothesis, $body(r) \subseteq \mathscr{T}_{D,WF}$. If $r \in R_{D,s}[p]$, then clearly $p \in \mathscr{T}_{D,WF}$.
So suppose $r \in R_d[p]$. For every $t \in R_E[\neg fi(r)]$, there is an $su(s) \in body(t)$ such that $su(s) \in \mathscr{U}_{E,\eta}$. This implies that $body(s) \cap \mathscr{U}_{E,\eta} \neq \emptyset$. By the inductive hypothesis, $body(s) \cap \mathscr{U}_{D,WF} \neq \emptyset$. Given this (and the definition of $trans(c,r)$), for any $c \in C_D[p]$, there must be a $q \in c - \{p\}$ such that for each $s \in R_D[q]$, $body(s) \cap \mathscr{U}_{D,WF} \neq \emptyset$ or else $s \prec_D r$. By definition of T_D and $\mathscr{I}_{D,WF}$, $p \in \mathscr{T}_{D,WF}$.

(2) Now suppose $p \in \mathscr{U}_{E,\lambda}$ and let a be any literal in $Lit(D) \cap \mathscr{U}_{E,\lambda}$. Then for each rule $r''' \colon \{fi(r)\} \rightarrow a$, $fi(r) \in \mathscr{U}_{E,\lambda}$ and for each rule $r'' \colon \{su(r)\} \dashrightarrow fi(r)$, either (1) $su(r) \in \mathscr{U}_{E,\lambda}$, or (2) r'' is defeasible and there is a $t : su(s_1), \ldots, su(s_n) \dashrightarrow \neg fi(r)$ such that $body(t) \subseteq \mathscr{T}_{E,\lambda-1}$ and t is strict (for NDL, $t \not\prec_E r''$). If (1) it follows that $body(r) \subseteq \mathscr{U}_{E,\lambda}$. If (2) then for each $su(s_i) \in body(t)$, $body(s_i) \subseteq \mathscr{T}_{E,\gamma}$ for some $\gamma < \lambda$ and by the inductive hypothesis $body(s_i) \subseteq \mathscr{T}_{D,WF}$. Given the construction of rules such as t, there exists a $c \in C_D[a]$ such that for all $q \in c - \{a\}$ there is a $s \in R_D[q]$ such that $body(s) \subseteq \mathscr{T}_{D,WF}$ and $r \prec_D s$ or s strict (for NDL, $s \not\prec_D r$). Generalizing on a, $\mathscr{U}_{E,\lambda}$ is unfounded *wrt* D and $\mathscr{T}_{D,WF}$, and so $p \in \mathscr{U}_{D,WF}$.

□

References

Antoniou, G. 2006. Defeasible reasoning: A discussion of some intuitions. *International Journal of Intelligent Systems 21*(6), 545–558.

Antoniou, G. and Billington, D. 2001. Relating defeasible and default logic. In *Proc. of 14th Australian Joint Conference on Artificial Intelligence (AUS-AI '01)*. Lecture Notes in Computer Science, vol. 2256. Springer, Berlin, 13–24.

Antoniou, G., Billington, D., Governatori, G. and Maher, M. J. 2000a. A flexible framework for defeasible logics. In *Proc. of 17th National Conference on Artificial Intelligence (AAAI '00)*. AAAI Press, Menlo Park, CA, 405–410.

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., Governatori, G. and Maher, M. J. 2006. Embedding defeasible logic into logic programming. *Theory and Practice of Logic Programming 6*(6), 703–735.

Antoniou, G., Billington, D., Governatori, G., Maher, M. J. and Rock, A. 2000b. A family of defeasible logics and its implementation. In *Proc. of 14th European Conference on Artificial Intelligence (ECAI '00)*. IOS Press, Amsterdam, 459–463.

Baral, C. and Subrahmanian, V. S. 1993. Dualities between alternative semantics for logic programming and nonmonotonic reasoning. *Journal of Automated Reasoning 10*(3), 399–420.

Billington, D. 1993. Defeasible logic is stable. *Journal of Logic and Computation 3*(4), 379–400.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Billington, D. 2004. A plausible logic which detects loops. In *Proc. of 10th International Workshop on Nonmonotonic Reasoning (NMR '04)*, 65–71.

Billington, D. 2005a. The proof algorithms of plausible logic form a hierarchy. In *Proc. of 18th Australian Joint Conference on Artificial Intelligence (AUS–AI '05)*. Lecture Notes in Computer Science, vol. 3809. Springer, Berlin, 796–799.

Billington, D. 2005b. A fixed-point semantics for plausible logic. In *Proc. of 18th Australian Joint Conference on Artificial Intelligence (AUS–AI '05)*. Lecture Notes in Computer Science, vol. 3809. Springer, Berlin, 812–815.

Billington, D. 2007. Entailment semantics for rules with priorities. In *Proc. of 20th International Joint Conference on Artifical Intelligence (IJCAI '07)*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 256–261.

Billington, D. 2008. Propositional clausal defeasible logic. In *Proc. of 11th European conference on Logics in Artificial Intelligence (JELIA '08)*. Lecture Notes in Computer Science, vol. 5293. Springer, Berlin, 34–47.

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

Billington, D. and Rock, A. 2001. Propositional plausible logic: Introduction and implementation. *Studia Logica 67*(2), 243–269.

Brewka, G. 1996. Well-founded semantics for extended logic programs with dynamic preferences. *Journal of Artificial Intelligence Research 4*, 19–36.

Brewka, G. 2001. On the relationship between defeasible logic and well-founded semantics. In *Proc. of 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR '01)*. Lecture Notes in Computer Science, vol. 2173. Springer, Berlin, 121–132.

Donnelly, S. 1999. *Semantics, Soundness, and Incompleteness for a Defeasible Logic*, Master's thesis. The University of Georgia, Athens, Georgia.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. of 5th International Conference on Logic Programming (ICLP '88)*. MIT Press, Cambridge, MA, 1070–1080.

Gelfond, M. and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9*(3–4), 365–385.

Governatori, G., Maher, M. J., Antoniou, G. and Billington, D. 2000. Argumentation semantics for defeasible logics. In *Proc. of 6th Pacific Rim international conference on Artificial intelligence (PRICAI '00)*. Lecture Notes in Computer Science, vol. 1886. Springer, Berlin, 27–37.

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

Horty, J. F., Thomason, R. H. and Touretzky, D. S. 1990. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence 42*(2–3), 311–348.

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

Maher, M. J. and Governatori, G. 1999. A semantic decomposition of defeasible logics. In *Proc. of 16th National Conference on Artificial Intelligence (AAAI '99)*. AAAI/MIT Press, Menlo Park, CA/Cambridge, MA, 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.

Maier, F. 2010. Well-founded semantics for defeasible logic. *Synthese 176*(2), 243–274.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press

Maier, F. and Nute, D. 2006. Ambiguity propagating defeasible logic and the well-founded semantics. In *Proc. of 10th European Conference on Logics in Artificial Intelligence (JELIA '06)*. Lecture Notes in Artificial Intelligence, vol. 4160, Springer, Berlin, 306–318.

Makinson, D. and Schlechta, K. 1991. Floating conclusions and zombie paths: Two deep difficulties in the 'directly skeptical' approach to inheritance nets. *Artificial Intelligence 48*(2), 199–209.

Marek, W. and Truszczynski, M. 1989. Stable semantics for logic programs and default theories. In *Proc. of 1989 North American Conference on Logic Programming (NACLP '89)*. MIT Press, Cambridge, MA, 243–256.

Nute, D. 1986. *LDR: A Logic for Defeasible Resasoning*. ACMC Research Report 01-0013. The University of Georgia, Athens, Georgia.

Nute, D. 1994. Defeasible logic. In *Handbook of Logic for Artificial Intelligence and Logic Programming*, Vol. III, D. Gabbay and C. Hogger, Eds. Oxford University Press, New York, 353–395.

Nute, D. 1997. Apparent obligation. In *Defeasible Deontic Logic: Essays in Nonmonotonic Normative Reasoning*, D. Nute, Ed. Kluwer Academic Publishers, Dordrecht, Netherlands, 287–316.

Nute, D. 1999. Norms, priorities, and defeasibility. In *Norms, Logics and Information Systems*, P. McNamara and H. Prakken, Eds. IOS Press, Amsterdam, 201–218.

Nute, D. 2003. Defeasible logic: Theory, implementation, and applications. In *Proc. of 14th International Conference on Applications of Prolog (INAP '01)*. Lecture Notes in Computer Science, vol. 2543. Springer, Berlin, 151–169.

Nute, D., Billington, D. and Coster, K. D. 1989. Defeasible logic and inheritance hierarchies with exceptions. In *Proc. of Tübingen Workshop on Semantic Networks and Nonmonotonic Reasoning*, vol. I. SNS Bericht, 89-48, University of Tübingen, 69–82.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence 13*(1–2), 81–132.

Sagonas, K. F., Swift, T. and Warren, D. S. 1994. XSB as an efficient deductive database engine. *ACM SIGMOD Record 23*(2), 442–453.

Schaub, T. and Wang, K. 2002. Preferred well-founded semantics for logic programming by alternating fixpoints. Preliminary report. In *Proc. of 9th International Workshop on Non-Monotonic Reasoning (NMR '02)*, 238–246.

Syrjänen, T. and Niemelä, I. 2001. The Smodels system. In *Proc. of 6th International Conference Logic Programming and Nonmonotonic Reasoning (LPNMR '01)*. Lecture Notes in Computer Science, vol. 2173. Springer, Berlin, 434–438.

Tarski, A. 1955. A lattice theoretic fixpoint theorem and its application. *Pacific Journal of Mathematics 5*(2), 285–309.

van Emden, M. and Kowalski, R. A. 1976. The semantics of predicate logic as a programming language. *Journal of the ACM 23*(4), 733–742.

Van Gelder, A., Ross, K. A. and Schlipf, J. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38*(3), 619–649.

https://doi.org/10.1017/S147106841100041X Published online by Cambridge University Press