# Reasoning About Priorities in Default Logic

**Gerhard Brewka**
GMD, Postfach 13 16
53731 Sankt Augustin, Germany
brewka@gmd.de

**Abstract**

In this paper we argue that for realistic applications involving default reasoning it is necessary to reason about the priorities of defaults. Existing approaches require the knowledge engineer to explicitly state all relevant priorities which are then handled in an extralogical manner, or they are restricted to priorities based on specificity, neglecting other relevant criteria. We present an approach where priority information can be represented *within* the logical language.

Our approach is based on PDL, a prioritized extension of Reiter's Default Logic recently proposed by the same author. In PDL the generation of extensions is controlled by an ordering of the defaults. This property is used here in the following way: we first build Reiter extensions of a given default theory. These extensions contain explicit information about the priorities of defaults. We then eliminate every extension $E$ that cannot be reconstructed as a PDL extension based on a default ordering that is compatible with the priority information in $E$. An example from legal reasoning illustrates the power of our approach.

## 1. Introduction

Defaults often conflict with each other. Consistency based approaches, like Reiter's default logic (Reiter 1980) or autoepistemic logic (Moore 1985), produce different extensions in such a case. Basically, there are as many extensions as ways to resolve conflicts among defaults. Approaches based on preferential models, like circumscription (McCarthy 1980) or preferential entailment (Kraus, Lehmann, & Magidor 1990) are intrinsically skeptical and often produce overly weak conclusions if there are many conflicts. For this reason the importance of default priorities is widely acknowledged. Such priorities allow implausible alternatives to be eliminated and are particularly relevant for many practical applications like diagnosis or design where less plausible solutions can be disregarded if defaults are prioritized accordingly.

A number of different techniques for handling priorities of defaults have been developed. Two main types of approaches can be distinguished:

1. approaches which handle explicit priority information that has to be specified by the user and is not part of the logical language, e.g. (Lifschitz 1985; Konolige 1988; Brewka 1989; Grosof 1991),
2. approaches which handle implicit priority information based on the specificity of defaults (Touretzky 1986; Touretzky, Horty, & Thomason 1987; Touretzky, Thomason, & Horty 1991; Pearl 1990; Geffner & Pearl 1992).

Although both types of approaches have provided useful techniques and insights we argue in this paper that a somewhat different treatment of priorities is needed.[1] For real world applications it seems unrealistic to assume that all relevant priorities can be specified by the user explicitly. On the other hand, specificity as the single preference criterion is entirely insufficient in many cases. Therefore we strongly believe that it should be possible to reason about default priorities in the logic in the same way we reason about properties of objects in the domain, e.g., about Tweety's flying ability. We want to be able to represent statements about the priorities in our domain theories and derive conclusions that take this priority information into account appropriately.[2]

One area where the need to reason about priorities has clearly been identified is legal reasoning, see for instance the two recent dissertations (Prakken 1993; Gordon 1993). In the legal domain the priority of one law over another conflicting law may depend on specificity considerations, but also on other criteria like recency (the newer law beats the older one) or authority (federal law beats state law). Note that a more recent general law may override a more specific older law, i.e. specificity is not always the main preference

[1] There are also approaches that combine the two types, like Goldszmidt and Pearl's System-$Z^+$ (Goldszmidt & Pearl 1991). However, this still does not give the expressiveness we want.

[2] An approach that is similar in spirit to ours was independently developed by Prakken (personal communication). Since the technical details in his system are still in flux we are unable to give a detailed comparison at the moment.

criterion. In particular Gordon has argued convincingly that reasoning about the priority of involved laws plays a fundamental role in legal decision making. Any logical model of such decision making should thus include reasoning about priorities.

The approach we propose in this paper is based on Reiter's default logic DL (Reiter 1980), and in particular on an extended version of DL called PDL recently proposed in (Brewka 1993). In PDL a partial order of the defaults controls the generation of extensions. This logic is therefore particularly well-suited for our purposes. Our approach is based on the following ideas:

1. we extend the logical language to make statements about default priorities possible,
2. we generate Reiter extensions of our default theories; these extensions contain information about priorities of defaults,
3. we eliminate all those extensions which cannot be reconstructed as PDL extensions using an ordering that is compatible with their own priority information.

Readers familiar with recent developments in nonmonotonic reasoning may wonder why we go back to default logic instead of using one of the more recent conditional approaches (Delgrande 1987; Lehmann 1990; Pearl 1990; Boutilier 1992) that are much more en vogue today. There are two answers to this:

1. Conditional nonmonotonic logics have difficulties to deal with irrelevant information. For instance, if the default "birds fly" is given, this does not sanction the conclusion that a particular green bird flies since the property of being green might be relevant to flying. The conditional approaches therefore have to make additional, often rather tedious, meta-theoretic assumptions that make it possible not only to reason about, but also with defaults in a satisfactory way.[3] DL, on the other hand, although weak at reasoning about defaults, is one of the logics that handle irrelevant information nicely. It therefore seems worthwhile to further investigate DL and similar logics.
2. The specificity criterion is built into the logical machinery of the conditional approaches, and in fact this is commonly viewed as one of their main advantages. Given that specificity appears to be only one preference criterion among many others, at least in certain applications, we consider this property as a disadvantage rather than an advantage because it is difficult to see how specificity could be overridden when necessary.

In general, it seems rather difficult to combine the techniques for reasoning about priorities developed in this paper with conditional logics since these techniques are based on the tentative generation and possible rejection of extensions.

For simplicity we will restrict ourselves in this paper to default theories where the number of defaults is finite. Moreover, following the view expressed in (Reiter & Criscuolo 1981; Brewka 1993) we consider the possibility to encode priorities as the main advantage of non-normal defaults, i.e. defaults whose consistency condition is not equivalent to the consequent. Since we investigate other explicit means for representing such priorities we will only be concerned with normal defaults in this paper (the exact definition of normal and non-normal defaults will be given in Section 2).

The rest of the paper is organized as follows: Section 2 briefly reviews the logics underlying our approach, namely Reiter's DL and and its prioritized generalization PDL. Section 3 is the central section of the paper. Here we show how reasoning about default priorities can be accomplished. Section 4 gives an extended example from the area of legal reasoning illustrating the power of the approach. Section 5 concludes.

## 2. Default logic with priorities: a brief review

In this section we briefly review Reiter's default logic DL (Reiter 1980) and in particular our modification of DL called PDL (Brewka 1993) that allows priorities to be represented explicitly and will be the basis of our approach to reasoning about priorities.[4] Our presentation of PDL here has to be very short. For a detailed discussion, motivation, examples, and comparison with other approaches we refer to the original paper (Brewka 1993).

In DL default theories consist of a set of facts $W$ and a set of defaults $D$. Each default is of the form $A{:}B_1, \ldots, B_n/C$ where $A$,$B_i$, and $C$ are closed formulas. We will use open defaults. i.e. defaults containing free variables, to represent all of their ground instances.

A default theory generates extensions which are defined as fixed points of an operator $\Gamma$. $\Gamma$ maps an arbitrary set of formulas $S$ to the smallest deductively closed set $S'$ that contains $W$ and satisfies the condition: if $A{:}B_1, \ldots, B_n/C \in D$, $A \in S'$ and for all $i$ $(1 \leq i \leq n)$ $\neg B_i \notin S$ then $C \in S'$. Extensions represent sets of acceptable beliefs a reasoner might adopt. They can be used to define a skeptical inference relation where a formula is defined to be provable iff it is contained in all extensions of $(D, W)$.

In PDL a strict partial order $<$ over the defaults can be specified in addition to facts and defaults. In (Brewka 1993) we argue that normal defaults of the form $A{:}B/B$ are sufficient if explicit means for representing priorities are available. Therefore PDL is defi-

[3] This "criticism" might seem a bit unfair since the approach proposed in this paper certainly can be viewed as meta-theoretic. However, we need such techniques for reasoning about priorities, not for reasoning with defaults as the conditional approaches.

[4] A similar but not equivalent prioritized version of DL has independently been developed by Baader and Hollunder (Baader & Hollunder 1993).

ned for normal defaults only. The default $A{:}B/B$ will be abbreviated $A \rightarrow B$ in this paper. Here are the necessary definitions:[5]

**Definition 1** *Let $E$ be a set of formulas, $\delta = a \rightarrow c$ a default. We say $\delta$ is active in $E$ iff (1) $a \in E$, (2) $c \notin E$, and (3) $\neg c \notin E$.*

**Definition 2** *Let $\Delta = (D, W, <)$ be a (prioritized) default theory, $\ll$ a strict total order containing $<$. We say $E$ is the (PDL) extension of $\Delta$ generated by $\ll$ iff $E = \bigcup E_i$, where $E_0 := Th(W)$, and*

$$E_{i+1} = \begin{cases} E_i & \text{if no default is active in } E_i \\ Th(E_i \cup \{c\}) & \text{otherwise, where } c \text{ is the consequent of the } \ll\text{-minimal default active in } E_i. \end{cases}$$

**Definition 3** *Let $\Delta = (D, W, <)$ be a (prioritized) default theory. $E$ is a (PDL) extension of $\Delta$ iff there is a strict total order containing $<$ that generates $E$.*

The following simple birds example illustrates these definitions. As usual we just list defaults and formulas, the sets $D$ and $W$ are obvious from syntax.

1) $b \rightarrow f$
3) $p \rightarrow \neg f$
2) $p \rightarrow b$
4) $p$

In DL we obtain two extensions, namely

$$E = Th(\{p, b, f\})$$

and

$$E' = Th(\{p, b, \neg f\}).$$

Now assume we define $3 < 1$. There are exactly three total orderings of the defaults respecting $<$, namely

$$2 < 3 < 1$$
$$3 < 2 < 1$$
$$3 < 1 < 2$$

It is easy to verify that in each case we obtain $E'$ as the generated extension. Consider as an example the first of the three total orderings. We obtain the following sequence of sets

$$E_0 = Th(\{p\})$$
$$E_1 = Th(\{p, b\})$$
$$E_2 = Th(\{p, b, \neg f\})$$
$$E_3 = E_2$$
$$\ldots$$

The two other orderings lead to the same extension. The single generated extension thus is the one where the preferred default 3 is applied.

Note that the definition of PDL extensions is fully constructive. In (Brewka 1993) we also present an alternative, non-constructive prioritized version of DL that formalizes somewhat different intuitions about the behavior of prioritized default rules. Furthermore, we show that the existence of PDL extensions is guaranteed for finite default theories, and that each PDL extension is a DL extension. The priority ordering can thus be viewed as a filter that distinguishes unwanted from wanted extensions. We will make use of this role in our approach to reasoning about priorities.

[5] Our terminology was influenced by the terminology used in (Baader & Hollunder 1993).

## 3. Reasoning about priorities

To be able to reason about default priorities it must be possible to refer to defaults explicitly, and we must introduce a special predicate symbol representing default priority. We we will therefore extend our logical language in two respects.

1. We will use named defaults (Poole 1988) of the form $d_i{:}a \rightarrow b$ where $d_i$ is taken from a distinct set of default names. We assume that different defaults have different names. Logically, default names are simply constants. These constants can be used for making references to defaults.[6]
2. We use the special two-place predicate symbol $\prec$ to represent default priority. For instance, if $d_1$ and $d_2$ are default names, then $d_1 \prec d_2$ is a formula with the intended meaning: $d_1$ has priority over $d_2$.

A (named) default theory is a pair $(D, W)$ where $D$ is a set of named defaults and $D$ and $W$ are defaults and formulas built from our extended logical language in the usual way. Note that we do not restrict the appearance of $\prec$ to $W$: it is possible to specify defaults about the priorities of other defaults.

We further assume that $W$ contains axioms guaranteeing that $\prec$ is a strict partial order. For this purpose we can use the following formulas

$$\{\forall x, y, z. x \prec y \wedge y \prec z \supset x \prec z, \forall x. \neg(x \prec x)\}.$$

Note that we implicitly assume that these formulas are contained in $W$. We will not explicitly mention them when discussing examples in this paper.

Given a set $D$ of named defaults we use $D^\circ$ to denote the corresponding set of defaults without names. By a DL extension of a named default theory $(D, W)$ we mean a DL extension of the corresponding unnamed theory $(D^\circ, W)$. Similarly, a PDL extension of a named prioritized default theory $(D, W, <)$ is a PDL extension of $(D^\circ, W, <)$.

Here is a rather simplistic example from the birds domain:

$d_1 : bird \rightarrow flies$
$d_2 : penguin \rightarrow \neg flies$
$penguin$
$bird$
$d_2 \prec d_1$

[6] Note that our treatment of names is somewhat different from Poole's: we use constant terms as names whereas Poole uses atomic propositions.

Let us first consider DL extensions of this default theory, neglecting priorities for the time being. There are two such extensions:

$$E_1 = Th(W \cup \{flies\})$$

and

$$E_2 = Th(W \cup \{\neg flies\})$$

Clearly, $E_1$ violates the intuitive meaning of the priority information contained in this extension since the only way to generate $E_1$ is by giving $d_1$ preference over $d_2$, yet $d_2 \prec d_1$ is a premise and thus contained in $E_1$. Only $E_2$ is compatible with its own priority information: $E_2$ is generated by giving $d_2$ preference over $d_1$ as is required according to the information about $\prec$ contained in $E_2$.

What we need, thus, is a way to eliminate every extension containing priority information which is in conflict with the way the extension was generated. Given the techniques developed for PDL it is not difficult to see how this can be done. Basically, a DL extension will "survive" if it can be reconstructed as a PDL extension with a generating total order $\ll$ that is compatible with its own priority information. Compatibility will be tested by producing a syntactic description of the generating order and testing consistency of this description with the extension. Here are the necessary definitions.

**Definition 4** *Let $\Delta = (D, W)$ be a named default theory, $E$ a DL extension of $\Delta$, and $\ll$ a strict total order of $D^\circ$. We say $\ll$ is compatible with $E$ iff*

$$E \cup \{d_i \prec d_k \mid d_i\colon r_i \in D, d_k\colon r_k \in D, r_i \ll r_k\}$$

*is consistent.*

**Definition 5** *Let $\Delta = (D, W)$ be a named default theory, $E$ a DL extension of $\Delta$. We say $E$ is a priority extension of $\Delta$ iff it is a PDL extension of $\Delta$ generated by a total order $\ll$ that is compatible with $E$.*

Reconsidering our birds example it is obvious that the single total ordering generating $E_1$, namely

$$\{(bird \rightarrow flies) \ll (penguin \rightarrow \neg flies)\}$$

is not compatible with $E_1$. $E_1$ is no priority extension for that reason. $E_2$, on the other hand, is a priority extension since it is generated by the total ordering

$$\{(penguin \rightarrow \neg flies) \ll (bird \rightarrow flies)\}$$

which, obviously, is compatible with $E_2$.

Given the expressiveness of our language it is not astonishing that unsatisfiable preference information can be specified and that this may lead to the non-existence of extensions. Here is a simple example. Assume $D$ consists of the two defaults

$d_1 : true \rightarrow d_2 \prec d_1$
$d_2 : true \rightarrow d_1 \prec d_2$

$W$ contains no formulas other than the two axioms for $\prec$. We obviously obtain two DL extensions. $E_1$ generated by $d_1$ contains the formula $d_2 \prec d_1$. However, the single total order compatible with $E_1$ prefers $d_2$ and hence does not reproduce $E_1$. Similarly, the second extension $E_2$ generated by $d_2$ contains the formula $d_1 \prec d_2$. Now the single total order compatible with $E_2$ prefers $d_1$. Again $E_2$ is not reproducible. This shows that our default theory has no priority extension at all.

Of course, we could try to weaken the notion of priority extensions to guarantee their existence, at least in the finite case. However, we strongly believe that the behaviour of our formalization is entirely reasonable. The specification of unsatisfiable priorities should lead to an exceptional situation requiring a reformulation of the knowledge base rather than being handled implicitly by the logical machinery. Otherwise there is a danger that mistakes of the knowledge engineer will remain unnoticed.

In the limiting case where a default theory $\Delta$ does not contain any constraining priority information, for instance since the predicate symbol $\prec$ is not explicitly mentioned anywhere in $\Delta$, priority extensions coincide with DL extensions. This follows from the fact that every strict partial ordering of the defaults is then compatible with every extension. As shown in (Brewka 1993) every DL extension can be reconstructed as a PDL extension in this case.

In the next section we will discuss a more realistic legal example that demonstrates the full power of our approach.

## 4. A legal reasoning example

The example we want to discuss in this section is taken from Gordon's dissertation (Gordon 1993, p.7). We somewhat simplified it for our purposes. Assume a person wants to find out if her security interest in a certain ship is perfected. She currently has possession of the ship. According to the Uniform Commercial Code (UCC, §9-305) a security interest in goods may be perfected by taking possession of the collateral. However, there is a federal law called the Ship Mortgage Act (SMA) according to which a security interest in a ship may only be perfected by filing a financing statement. Such a statement has not been filed. Now the question is whether the UCC or the SMA takes precedence in this case. There are two known legal principles for resolving conflicts of this kind. The principle of *Lex Posterior* gives precedence to newer laws. In our case the UCC is newer than the SMA. On the other hand, the principle of *Lex Superior* gives precedence to laws supported by the higher authority. In our case the SMA has higher authority since it is federal law.

As we will see our approach allows us to formalize exactly this kind of reasoning. We use the ground instances of the following named defaults to represent the relevant article of the UCC, the SMA, Lex Posterior (LP), and Lex Superior (LS):

$UCC : possession \rightarrow perfected$
$SMA : ship \wedge \neg fin\text{-}statement \rightarrow \neg perfected$
$LP(d_1, d_2) : more\text{-}recent(d_1, d_2) \rightarrow d_1 \prec d_2$
$LS(d_1, d_2) : fed\text{-}law(d_1) \wedge state\text{-}law(d_2) \rightarrow d_1 \prec d_2$

The following facts are known about the case:

1) $possession$
2) $ship$
3) $\neg fin\text{-}statement$
4) $more\text{-}recent(UCC, SMA)$
5) $fed\text{-}law(SMA)$
6) $state\text{-}law(UCC)$

For this default theory we obtain four different Reiter extensions, namely

$E_1 = Th(W \cup \{perfected, UCC \prec SMA\})$
$E_2 = Th(W \cup \{\neg perfected, UCC \prec SMA\})$
$E_3 = Th(W \cup \{perfected, SMA \prec UCC\})$
$E_4 = Th(W \cup \{\neg perfected, SMA \prec UCC\})$

Two of these extensions are not priority extensions, namely $E_2$ and $E_3$. The priority information in $E_2$ requires that $UCC$ gets preference over $SMA$, yet to derive $\neg perfected$ it is necessary to violate this requirement. Similarly, the priority information in $E_3$ requires that $SMA$ gets preference over $UCC$, yet to derive $perfected$ this requirement has to be violated.

The two other extensions, $E_1$ and $E_4$, are priority extensions as can easily be verified. Hence the question whether the security interest is perfected is still open.

The intuitive reason for this is obvious: we have a conflict between the relevant instances of Lex Posterior and Lex Superior. Such a conflict can, for instance, be resolved by a universal rule that gives preference to the latter in all cases. To model this we have to add to $W$ the formula

7) $\forall x, y, v, w. LS(x, y) \prec LP(v, w)$

Let $W'$ denote this new set of facts. Again we obtain four DL extensions $E'_i$ whose definition is obtained from the definition of $E_i$ by replacing $W$ with $W'$. $E'_2$ and $E'_3$ are not priority extensions for the same reasons $E_2$ and $E_3$ are not priority extensions. But now also $E'_1$ violates its own priority information. To see this note that every extension must contain the instance of 7)

$$LS(SMA, UCC) \prec LP(UCC, SMA).$$

which states that Lex Posterior can only be used to derive a priority of UCC over SMA if Lex Superior cannot be used to derive the opposite priority. This condition is obviously violated. In other words, there is no total ordering $\ll$ of the defaults in $D$ such that the default instance $LS(SMA, UCC)$ precedes the default instance $LP(UCC, SMA)$ but nevertheless $UCC \prec SMA$ is contained in the PDL extension of $(D, W, \ll)$.

The single priority extension in our example is thus $E'_4$, i.e., we derive that the security interest in the ship is not perfected.

We have used above a formula in $W$ to give Lex Superior priority over Lex Posterior in all cases. Of course, there might be exceptions also to this conflict resolution strategy. To model this, we could simply replace 7) by a corresponding default and include a description of what to do in the exceptional cases in our default theory. We could then distinguish the following different levels:

1. the level of the basic laws $UCC$ and $SMA$,
2. the level of principles solving conflicts among basic laws, i.e. Lex Posterior and Lex Superior,
3. the level solving conflicts among the latter, and
4. the level regulating the applicability of and specifying exception handling mechanisms for the strategy described in level 3).

This illustrates that there is no fixed highest level of reasoning about priorities: we can always add a further level describing priorities of defaults at the next lower level.

## 5. Conclusions

In this paper we have shown that it is possible to reason about default priorities within default logic. We introduced names for defaults and a special predicate symbol $\prec$ to express default priorities explicitly within the logical language. The information about default priorities contained in the Reiter extensions of a default theory was used to filter out those who can be reconstructed as PDL extensions in a way that is compatible with their own priority information.

Although we avoided the term so far, it would certainly be adequate to view this filtering process as a form of reflection. A meta-level process, namely the process of generating an extension, is matched against the outcome of this process, the extension itself. This is done by producing a syntactic description of relevant aspects of the process, namely the priorities involved in the generation of the extension, and by testing the consistency of this syntactic description with the generated extension.

The system we presented allows default priorities to be handled in an extremely flexible way. Explicit priorities, as they can, for instance, be specified in prioritized circumscription (Lifschitz 1985), hierarchic autoepistemic logic (Konolige 1988) or preferred subtheories (Brewka 1989), can be represented by simply asserting corresponding atomic formulas in $W$. Similarly, priorities based on specificity can be represented by asserting formulas corresponding to the output of the specificity algorithms used, e.g. Pearl's Z-ordering (Pearl 1990) or the specificity ordering developed in (Brewka 1993). However, our approach gives us much more flexibility. We can, for instance, express that certain priorities should hold under specific conditions only, derive priorities from the available information, and specify strategies how to resolve priority conflicts, as was done in the legal example in the last section.

Although the presentation of our priority handling techniques in this paper was based on PDL it should be obvious that they do not depend on this choice of the underlying nonmonotonic system. These techniques can easily be combined with any consistency based nonmonotonic system that produces extensions and can handle explicit priorities, like hierarchic autoepistemic logic (Konolige 1988) or preferred subtheories (Brewka 1989).

We have pushed the expressiveness of default logic to an extreme in our approach. Of course, there is a price to pay for this: the computation becomes much more difficult. As always there are two possible solutions for this problem: finding good approximations or finding interesting special cases with reasonable computational properties. A first step into the latter direction has been made in (Junker 1993) where Junker develops enumeration based proof procedures for Horn theories with dynamically derived preferences.

Much more work of this kind is needed. Nevertheless, since priorities play such an extremely important role for many applications we hope that this approach might in the long run help making nonmonotonic reasoning techniques more widely applicable for solving real world problems.

## Acknowledgements

I would like to thank Tom Gordon, Joachim Hertzberg, Ulrich Junker, and Henry Prakken for interesting discussions on the topic of this paper.

## References

Baader, F., and Hollunder, B. 1993. How to prefer more specific defaults in terminological default logic. In *Proc. IJCAI-93, Chambery, France.*

Boutilier, C. 1992. *Conditional Logics for Default Reasoning and Belief Revision.* Ph.D. Dissertation, Dep. of Computer Science, Univ. of Toronto.

Brewka, G. 1989. Preferred subtheories - an extended logical framework for default reasoning. In *Proc. IJCAI-89, Detroit.*

Brewka, G. 1993. Adding priorities and specificity to default logic. Technical report, GMD.

Delgrande, J. 1987. A first-order conditional logic for prototypical properties. *Artificial Intelligence, 33(1).*

Geffner, H., and Pearl, J. 1992. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence 53.*

Goldszmidt, M., and Pearl, J. 1991. System $Z^+$: A formalism for reasoning with variable-strength defaults. In *Proc. AAAI-91.*

Gordon, T. 1993. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice.* Ph.D. Dissertation, TU Darmstadt.

Grosof, B. 1991. Generalizing prioritization. In *Proc. Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge.*

Junker, U. 1993. Dynamic generation of assumptions and preferences. unpublished manuscript.

Konolige, K. 1988. Hierarchic autoepistemic theories for nonmonotonic reasoning. In *Proc. AAAI-88.*

Kraus, S.; Lehmann, D.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence.*

Lehmann, D. 1990. What does a conditional knowledge base entail? In *Proc. First International Conference on Principles of Knowledge Representation and Reasoning.*

Lifschitz, V. 1985. Computing circumscription. In *Proc. 9th Int. Joint Conference on Artificial Intelligence.*

McCarthy, J. 1980. Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence 13.*

Moore, R. C. 1985. Semantical considerations on nonmonotonic logic. *Artificial Intelligence 25.*

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proc. Third Conference on Theoretical Aspects of Reasoning About Knowledge.*

Poole, D. 1988. A logical framework for default reasoning. *Artificial Intelligence 36* 27–47.

Prakken, H. 1993. *Logical Tools for Modelling Legal Argument.* Ph.D. Dissertation, VU Amsterdam.

Reiter, R., and Criscuolo, G. 1981. On interacting defaults. In *Proc. IJCAI-81, Vancouver.*

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

Touretzky, D. S.; Horty, J. F.; and Thomason, R. H. 1987. A clash of intuitions: The current state of nonmonotonic multiple inheritance systems. In *Proc. IJCAI-87, Milan.*

Touretzky, D. S.; Thomason, R. H.; and Horty, J. F. 1991. A skeptic's menagerie: Conflictors, preemptors, reinstaters, and zombies in nonmonotonic inheritance. In *Proc. IJCAI-91, Sydney.*

Touretzky, D. S. 1986. *The Mathematics of Inheritance.* London: Pitman Research Notes in Artificial Intelligence.