Default Logic*

David Poole
Department of Computer Science,
University of British Columbia,
Vancouver, B.C., Canada V6T 1W5
(604) 228 6254
poole@cs.ubc.ca

May 14, 1998

1 Logic and Monotonicity

In nonmonotonic reasoning we want to reach conclusions that we may not reach if we had more information. There seem to be two ways to handle this; we could change the logic to be defeasible, or we could allow there to be some premises of the logical argument that may not be allowed when new information is received. Default logic is a formalisation of the latter; it provides rules that add premises to logical arguments. The advantages of this framework are its simplicity, its naturalness, its power and the abundance of applications.

Suppose we have a logical argument that Tweety flies, based on Tweety being a bird, and the fact that "birds fly". If we subsequently learn that Tweety is an emu, then the conclusion (that Tweety flies) is wrong, but the logical proof is still valid. The logical argument is valid; the conclusion is incorrect, so one of the premises is incorrect. The incorrect premise is "birds fly". We don't want to use this premise when the object under consideration

*Invited to appear in Gabbay and Hogger (Eds.) *Handbook of Logic for Artificial Intelligence and Logic Programming*.

is an emu. The idea that there are premises that we want to use some of the time, but not all of the time is the basis of default logic and its derivatives. Default logic provides a set of rules for adding premises to logical arguments. "Defaults function as meta-rules whose role is to further complete un underlying incomplete first order theory" [Reiter, 1980, section 2.2]. Default logic was invented by Reiter [Reiter, 1980], and has been most fully investigated by Etherington [Etherington, 1987b]. Poole, Goebel and their colleagues have empirically investigated a simple case of default logic [Poole *et al.*, 1987; Poole, 1988b]. Besnard [Besnard, 1989] gives a comprehensive overview of theoretical work in default logic.

Section 2 presents a framework for default reasoning in which defaults provide logical formulae that can be used as premises if they can be consistently added by the use of "default rules". These default rules can take into account the derivability of some formulae and the consistency of other formulae [Reiter, 1980]. Section 3 provides many examples of both the power of, and the assumptions behind, the framework. Section 4 discusses the semantics of the resulting systems. A fixed point and a minimal model semantics are given. Section 5 gives comparisons with other formulations of nonmonotonic reasoning. Implementing default reasoning systems is discussed in section 6. Section 7 considers when we can guarantee the existence of extensions. Complexity of default reasoning is discussed in section 8, empirical investigations are discussed in section 9, applications that have been built on this framework are presented in section 10, and variations that have been suggested are given in section 11.

2 Reiter's Logic for Default Reasoning

We assume we are given a standard first order language over a countable alphabet [Enderton, 1972]. By a formula we mean a well formed formula in this language. By an instance of a formula we mean a substitution of terms in this language for free variables in the formula. A ground instance is one that contains no free variables.

A default theory is a pair $\Delta = \langle D, F \rangle$ where:

F a set of closed formulae, called the "facts";

D a set of "defaults" of the form

$$\frac{\alpha(\overline{x}) : \beta_1(\overline{x}), ..., \beta_m(\overline{x})}{w(\overline{x})}$$

where $\alpha(\overline{x}), \beta_1(\overline{x}), ..., \beta_m(\overline{x}), w(\overline{x})$ are formulae whose free variables are amongst those of $\overline{x}$.

The default rules specify formulae that can be used as premises of a logical argument. The above rule means, intuitively that a ground instance of $w(\overline{x})$ can be used if the corresponding instance of $\alpha(\overline{x})$ is proved and the corresponding instances of $\beta_i(\overline{x})$ are consistent with everything believed.

Definition 2.1 In the default

$$\frac{\alpha(\overline{x}) : \beta_1(\overline{x}), ..., \beta_m(\overline{x})}{w(\overline{x})}$$

$\alpha(\overline{x})$ is the **precondition** of the default; $w(\overline{x})$ is the **consequent** of the default; and the $\beta_i(\overline{x})$ are the **justifications** of the default.

Definition 2.2 A default is **closed** if it does not contain any free variables. A default is **open** otherwise.

Definition 2.3 An **instance** of a default is obtained by uniformly substituting ground terms for the free variables in the default.

Definition 2.4 A default is **normal** if it is of the form:

$$\frac{\alpha(\overline{x}) : w(\overline{x})}{w(\overline{x})}.$$

Definition 2.5 A default is **semi-normal** if it is of the form:

$$\frac{\alpha(\overline{x}) : \beta(\overline{x})}{w(\overline{x})}.$$

such that $\forall \overline{x}\ \beta(\overline{x}) \Rightarrow w(\overline{x})$. This normally happens when $\beta(\overline{x})$ is $w(\overline{x}) \wedge \gamma(\overline{x})$ for some formula $\gamma(\overline{x})$.

Formally, the rule for specifying when a conclusion of a default can be added to an argument is defined in terms of an extension.

Definition 2.6 Given a default theory $\Delta = \langle D, F\rangle$, consider a sequence of formulae $S_0, S_1, S_2, ...$, where $S = \cup_{i=0}^{\infty} S_i$, $S_0 = F$, and

$$\begin{aligned} S_{i+1} \quad = S_i \cup \{w(\overline{c}) : \; & \frac{\alpha(\overline{c}) : \beta_1(\overline{c}), ..., \beta_m(\overline{c})}{w(\overline{x})} \text{ is an instance of a default in } D, \\ & \alpha(\overline{c}) \text{ follows from } S_i, \\ & \beta_j(\overline{c}) \text{ is consistent with } S, \text{ for all } j, 1 \leq j \leq m\} \end{aligned}$$

An **extension** of $\Delta = \langle D, F\rangle$ is the set of logical consequents of S.

An extension provides a set of consequences of a default theory. An instance of the consequent of a default can be used as the premise of a logical argument if

1. we can prove the instance of the precondition from facts and previously assumed consequents. This enforces a groundedness of assumptions; we don't allow circularity in the derivation of a default consequent.

2. the justifications are consistent with the union of the consequents. Note that the $\beta_j(c)$ have to be consistent with S, the union of the S_i.

One other definition that is useful is to consider the set of consequents of a set of defaults that can be used to imply some goal.

Definition 2.7 If g is a closed formula, E is an **explanation** of g from $\Delta = \langle D, F\rangle$ if E is the set of consequents of some D', a set of instances of elements of D such that $E \cup F \models g$, $E \cup F$ entails the preconditions of D', and all of the justifications of D' are consistent with some extension of $\Delta = \langle D, F\rangle$ that contains E.

In other words, an explanation of g is the set of consequents of defaults that are needed to imply g. The definition also ensures that the explanation is in some extension. "g can be explained" is equivalent to "g is in some extension". This is useful as an explanation is a minimalist notion; we only need to find a set of conclusions to prove a goal, the preconditions of the rules, and some reason that the defaults used can be extended to an extension.

The default framework is about arguments, rather than about prediction. An extension is "a set of beliefs which are in some sense "justified" or "reasonable" in light of what is known about the world" [Etherington, 1987b] rather than a prediction of what is true. Different applications may want to make different uses of the conclusions.

Two main ways to use such a system for prediction have been proposed:

brave In brave prediction we predict what is in some arbitrarily chosen extension [Reiter, 1980; Etherington, 1987a]. Choosing a different extension may change what is predicted.

skeptical In sceptical prediction we predict what is in all extensions. We thus predict only what we can explain even if an adversary chooses the defaults [Poole, 1989a].

3 Examples

In this section we show a series of examples that are intended to show the different features of defaults. We start with the simplest forms of defaults, namely normal defaults without preconditions.

Following tradition we use the example of representing the default "birds fly". The example will be complicated slightly by having an exceptional class of baby birds for which we want to make no assumptions about their flying ability. That is we want to represent "birds fly, except for baby birds".

3.1 Normal Defaults

Example 3.1 The "birds fly" example can be represented using the simplest form of defaults as follows:[1]

$$\begin{aligned} D &= \{ \quad \frac{: birdsfly(x)}{birdsfly(x)} \} \\ F &= \{ \quad \forall x \; birdsfly(x) \land bird(x) \Rightarrow flies(x), \end{aligned}$$

[1]We are using the idea of "naming defaults" [Poole, 1988a]. If this is the only place that the predicate "birdsfly" appears, then the default and the first two facts can be replaced by

$$D \;=\{ \quad \frac{: bird(x) \Rightarrow flies(x) \land \neg baby(x)}{bird(x) \Rightarrow flies(x) \land \neg baby(x)} \}$$

and all conclusions that do not involve the predicate $birdsfly$ will be unchanged.

$$\begin{aligned}
&\forall x\ bird(x) \wedge baby(x) \Rightarrow \neg birdsfly(x) \\
&bird(Tweety), \\
&baby(Polly), \\
&bird(Polly), \\
&\neg flies(Fred)\}
\end{aligned}$$

$flies(Tweety)$ can be explained using the explanation $F \cup \{birdsfly(Tweety)\}$. We cannot explain $flies(Polly)$ as $birdsfly(Polly)$ is not consistent with the facts. We can explain $\neg bird(Fred)$, using $F \cup birdsfly(Fred)$. We can also explain $bird(t) \Rightarrow flies(t) \wedge \neg baby(t)$ for every ground term t (except for $t = Polly$).

The next example shows how the use of preconditions can be used to reduce the number of conclusions:

Example 3.2 Consider the following normal default representation of "birds fly, but baby birds are exceptional":

$$\begin{aligned}
D &= \{ \quad \frac{bird(x) : flies(x) \wedge \neg baby(x)}{flies(x) \wedge \neg baby(x)} \} \\
F &= \{ \quad bird(Tweety), \\
&\qquad baby(Polly), \\
&\qquad bird(Polly), \\
&\qquad \neg flies(Fred)\}
\end{aligned}$$

Here we can explain $flies(Tweety)$ and $\neg baby(Tweety)$, by assuming the default for $x = Tweety$. We still cannot explain $flies(Polly)$, and can no longer explain $\neg bird(Fred)$, nor $bird(t) \Rightarrow flies(t) \wedge \neg baby(t)$ for an arbitrary ground term t.

The difference between these representations is that in example 3.2, we must derive that x is a bird before we can use the default (we have to "know" that the individual is a bird), whereas in example 3.1 we could use the default for any individual. There are a number of consequences of this difference:

Example 3.3 Suppose we add

$$bird(Oscar) \vee bird(Sylvester)$$

to the facts F of example 3.1.

From the representation of defaults in example 3.1 we could conclude

$$flies(Oscar) \vee flies(Sylvester)$$

(assuming both $birdsfly(oscar)$ and $birdsfy(sylvester)$).

From example 3.2 we cannot conclude anything about the flying ability of Oscar or Sylvester. The reason is that we can prove neither $bird(Oscar)$ nor $bird(Sylvester)$, and so the default is not applicable for either individual.

Example 3.4 [Makinson, 1988]

$$\begin{aligned} R &= \{ \frac{:p}{p} \\ &\quad\ \frac{p \vee q : \neg p}{\neg p} \} \\ F_1 &= \{ \ \} \\ F_2 &= \{ \ p \vee q \} \end{aligned}$$

In this example, there is one extension of $\langle F_1, R\rangle$, namely $Th(\{p\})$[2]. There are two extensions of $\langle F_2, R\rangle$, namely $Th(\{p\})$ and $Th(\{\neg p, q\})$.

Notice here that the only element of F_2, namely $p \vee q$ is in the only extension of $\langle F_1, R\rangle$. p is in the only extension of $\langle F_1, R\rangle$, but is not in one of the extensions of $\langle F_2, R\rangle$.

This example is interesting [Makinson, 1988] because it demonstrates that normal defaults (with preconditions) satisfy neither "cumulative monotony" (if x and y both follow from the facts, then if we add y to the facts, we still derive x — here y is $p \vee q$ and x is $\neg p$) nor "cumulative transitivity" (if we can derive x by adding a derived consequent to the facts, then we can derive x without the derived consequent — here x is $\neg p$ and the derived consequence is $p \vee q$) when consequence is interpreted in the "brave" sense of prediction.

3.2 Non-Normal Defaults

The following example shows how we can use semi-normal defaults in order to prevent the conclusion $\neg baby(Tweety)$.

[2]If S is a formula $Th(S)$ is the set of logical consequents of S.

Example 3.5 Consider the semi-normal default

$$D \quad = \{ \quad \frac{bird(x) : flies(x) \wedge \neg baby(x)}{flies(x)} \}$$

Using the same facts as in example 3.2, we can now explain $flies(Tweety)$ by assuming the default for $x = Tweety$, but can no longer explain $\neg baby(Tweety)$.

Not concluding $\neg baby(Tweety)$ is, however, done at a cost:

Example 3.6 Consider the default of example 3.5, together with the facts

$$\begin{aligned} F \quad = \{ \quad & bird(Pete), \\ & bird(Mary), \\ & baby(Pete) \vee baby(Mary)\}. \end{aligned}$$

We can now explain $flies(Mary) \wedge flies(Pete)$. The disjunctive exception was not strong enough to block either default (or the use of both defaults).

This example shows that in some sense we have to "know" the individual is exceptional before the default is blocked. Semi-normal defaults are most applicable where the knowledge of the exception is important, as in the following example:

Example 3.7 [Etherington] People who are employed should get paid unless you know that they did not work:

$$\begin{aligned} D \quad &= \{ \quad \frac{employed(x) : get\_paid(x) \wedge worked(x)}{get\_paid(x)}, \\ F \quad &= \{ \quad employed(David), \\ & \qquad employed(John), \\ & \qquad \neg worked(David) \vee \neg worked(John)\} \end{aligned}$$

In this example the conclusion that both David and John should get paid is not unreasonable (particularly if you don't want to be sued).

Example 3.8 Note that the use of disjunctive exceptions does not require explicit disjunctions. Consider the following defaults:

$$\begin{aligned} D &= \{ \frac{bird(x) : flies(x) \wedge \neg baby(x)}{flies(x)}, \\ &\qquad \frac{small(x) : cries(x) \wedge baby(x)}{cries(x)} \} \\ F &= \{ bird(Tweety), \\ &\qquad small(Tweety) \} \end{aligned}$$

In this case, when Tweety is a small bird, there is one extension containing

$$flies(Tweety) \wedge cries(Tweety)$$

We have to know whether Tweety is a baby or not in order to block one of the defaults.

Example 3.9 In order to solve the "problems" of using the contrapositive in example 3.1 and of the disjunctive exception of example 3.3, Brewka [Brewka, 1989a] suggested representing "birds fly" as the semi-normal default

$$\frac{: flies(x)}{bird(x) \Rightarrow flies(x)}$$

Using this default we can no longer conclude $\neg bird(Fred)$ given $\neg flies(Fred)$ (as in example 3.1), but we can still conclude $flies(Oscar) \vee flies(Sylvester)$ from $bird(Oscar) \vee bird(Sylvester)$.

Using the same trick as in example 3.6 (namely disjoining the exception of the semi-normal default), we can create another peculiar example for Brewka's suggestion. We cannot conclude $\neg bird(Fred)$ given $\neg flies(Fred)$, nor can we conclude $\neg bird(George)$ given $\neg flies(George)$, however this default lets us conclude $\neg bird(Fred) \vee \neg bird(George)$ given $\neg flies(Fred) \vee \neg flies(George)$.

In following example, the semi-normal default may be more appropriate than the corresponding normal default:

Example 3.10 [Etherington, 1987a] Someone who has a motive, and may be guilty, should be a suspect:

$$\frac{\textit{has-motive}(x) : \textit{suspect}(x) \wedge \textit{guilty}(x)}{\textit{suspect}(x)}$$

The corresponding normal default:

$$\frac{\textit{has-motive}(x) : \textit{suspect}(x) \wedge \textit{guilty}(x)}{\textit{suspect}(x) \wedge \textit{guilty}(x)}$$

is much less reasonable. We don't also want to conclude that all suspects are guilty. Disjunctive exceptions also seem reasonable for this example; if we know both Pete and Mary have motives and that we know one is not guilty, it is reasonable to conclude that they both are suspects.

The next example gives one possible use for multiple justifications in default rules, and so for default rules that are not even semi-normal.

Example 3.11

$$\frac{: p, \neg p}{u}$$

Using this default we conclude u only if we cannot conclude p or $\neg p$. u may be interpreted as "unknown whether p". We probably don't want to make this into a semi-normal default.

3.3 Equality

In this set of examples we show how equality is affected by default reasoning. Note that defaults add syntactic premises to logical theories, these can include statements of equality and inequality. There is no need for unique names assumptions (assuming different terms denote different objects).

Example 3.12 Defaults that do not contain equality can be used to derive inequalities:

$$\begin{aligned} D &= \{ \frac{: p(x)}{p(x)} \} \\ F &= \{ \neg p(A) \} \end{aligned}$$

Here we can conclude $p(B)$ from which it logically follows that $A \neq B$.

Example 3.13 We can make default assumptions about equality; for example we can have a default that embodies the unique names assumption:

$$\frac{: x \neq y}{x \neq y}$$

From this we conclude that any terms that cannot be shown to denote the same individual do not denote the same individual.

Example 3.14 Defaults can also be used to imply equality. For example, a detective may assume that a series of murders were committed by the same person unless that can be shown not to be:

$$\begin{aligned} D &= \{ \frac{murderer(x) \wedge murderer(y) : x = y}{x = y} \} \\ F &= \{ murderer(Jill\_the\_ripper), \\ &\quad murderer(Night\_stalker), \\ &\quad murderer(Rambo\_follower), \\ &\quad red\_haired(Rambo\_follower), \\ &\quad \neg red\_haired(Night\_stalker) \} \end{aligned}$$

Here there are two extensions, one containing

$$Jill\_the\_ripper = Night\_stalker$$

and the other containing

$$Jill\_the\_ripper = Rambo\_follower.$$

3.4 Skolemization

The definition of defaults given so far differs from Reiter's defaults [Reiter, 1980][section 7.1], in that Reiter allows for Skolem functions to be used as instances of defaults, whereas we have assumed here that we can only have terms of the language before Skolemization. The difference can be seen in the following example:

Example 3.15

$$\begin{aligned} D &= \{ \quad \frac{: p(x)}{p(x)} \} \\ F &= \{ \quad \neg p(a), \\ & \qquad (\forall x\ p(x)) \Rightarrow g \} \end{aligned}$$

In the reading given in this paper we cannot derive g.

Reiter's default logic tells us to Skolemize first, and then allow the substitution of ground terms for variables in the default:

$$\begin{aligned} D &= \{ \quad \frac{: p(x)}{p(x)} \} \\ F_{sk} &= \{ \quad \neg p(a), \\ & \qquad p(C) \Rightarrow g \} \end{aligned}$$

Reiter's default logic allows the conclusion of g, by using the default for $x = C$, where C is the introduced Skolem constant.

I would argue that we should *not* be able to derive g (as it is simply not true that $\forall x\ p(x)$), and thus that Reiter's solution is not correct. This issue is discussed further in [Poole, 1987].

4 Semantics

So far we have defined default logic in terms of providing premises for logical arguments. In this section we provide various forms of semantics for the most expressive forms of defaults.

4.1 Fixed Points

Fixed point semantics provides a definition of a set of consequences.

We give the fixed-point semantics for [Reiter, 1980]'s general defaults[3].

Theorem 4.1 (Reiter80, theorem 2.1) An extension is the smallest set E satisfying

[3] Reiter used this as his definition of extension, and derived the other definition. We believe that the other definition is a more natural definition of an extension.

1. $F \subseteq E$
2. $Th(E) = E$
3. If
$$\frac{\alpha(\overline{c}) : \beta_1(\overline{c}), ..., \beta_m(\overline{c})}{w(\overline{c})}$$
is a ground instance of a default and $\alpha(\overline{c}) \in E$ and $\neg\beta_i(\overline{c}) \notin E$ then $w(\overline{c}) \in E$.

Proof: First of all we should show that an extension satisfies the three points. The first and second are trivial. For the third, if $\alpha(\overline{x}) \in E$, then for some i, $S_i \models \alpha(\overline{x})$, and so by construction $w(\overline{x}) \in S_{i+1}$, so $w(\overline{x}) \in E$.

Suppose there is a set E' that is a subset of extension S and satisfies the above formulae. It is easy to check that, by induction, $S_i \subseteq E'$ for all i. Thus $S = \cup_{i=0}^{\infty} S_i \subseteq E'$, and so $S = E'$. $\square$

4.2 Stable Model Semantics

One interesting subcase of default logic occurs when F is a conjunction of atoms, and the consequent of each default is an atom, the justifications of defaults are all negations of atoms, and the preconditions of defaults are conjunctions of atoms. That is, all of the defaults are of the form

$$\frac{a_1 \wedge ... \wedge a_n : \neg a_{n+1}, ..., \neg a_m}{a_0}$$

where the a_i are all atoms. In this case, extensions are the consequences of conjunctions of atoms, and can be identified with interpretations. The default rule can be put in correspondence with the Prolog (with negation as failure) clause

$$a_0 : -a_1, ..., a_n, \sim a_{n+1}, ..., \sim a_m.$$

where $\sim a$ means the proof for a finitely fails. The correspondence between the default rule and the Prolog clause and the correspondence between a fixed point and an interpretation, form the basis behind stable model semantics [Gelfond and Lifschitz, 1988]. Note that originally stable model semantics was based on autoepistemic extensions [Moore, 1985], rather than default extensions defined here. The differences are due to the groundedness demanded of default extensions [Konolige, 1987]. This difference means that the default extensions are better at defining the stable models.

4.3 Minimal Models

Intuitively default logic has given a way to specify premises in a logical argument. Default logic has a "maximal model" semantics in that g is in an extension iff g is true in every model of that extension. The set of models of extensions is a subset of the set of models of F. This correspondence allows us to give a maximal model semantics to default logic.

The presentation here follows that in [Etherington, 1987a] which builds on the work in [Etherington, 1987c; Łukaszewicz, 1985]. Sets of models are compared rather than individual models as in other frameworks [Shoham, 1987]. Sets of models are needed as we need to capture derivability, which is needed for both ensuring that preconditions are met, and for ensuring that justifications are consistent.

The maximal model semantics can be defined as follows.

Definition 4.2 If $\mathcal{M}_1$ and $\mathcal{M}_2$ are sets of models of F, we say that $\mathcal{M}_1 >'_R \mathcal{M}_2$ if there is a default of the form

$$\frac{\alpha : \beta_1, ..., \beta_n}{\gamma}$$

(an instance of an element of D) such that for all $M \in \mathcal{M}_2$, α is true in M, and for each i such that $1 \leq i \leq n$, there exists $M \in \mathcal{M}_2$ such that β_i is true in M and

$$\mathcal{M}_1 = \{M \in \mathcal{M}_2 \text{ such that } \gamma \text{ is true in } M\}$$

Let $>_R$ be the transitive closure of $>'_R$.

Let $\mathcal{M}$ be the set of models of F.

Intuitively, we start with $\mathcal{M}$ and find a maximal non-empty model set greater than $\mathcal{M}$. The reason that this is correct is that we are considering the set of models of the S_i in the definition of an extension (definition 2.6) rather than the formulae S_i. The only difference is that the β_i needs to be consistent with the resulting fixed point. In the model-based view this is accomplished by a notion of stability:

Definition 4.3 A set $\mathcal{I}$ of models of F is **stable** for D if there is a set R' of instances of elements of D such that $\mathcal{I} \geq_{R'} \mathcal{M}$, and for each

$$\frac{\alpha : \beta_1, ..., \beta_n}{\gamma} \in R'$$

for each i, $1 \leq i \leq n$, there is some $M_i \in \mathcal{I}$ such that $M_i \models \beta_i$.

Theorem 4.4 (Etherington) *g is in an extension iff g is true in all members of a stable maximal model set greater than $\mathcal{M}$.*

> **Proof idea:** By construction, a stable, minimal model set less than $\mathcal{M}$ is exactly the set of models of an extension. g logically follows from an extension iff it is true in all models of the extension. See [Etherington, 1987a, theorems 3.2 and 3.3] □

We say M is a minimal model if M is a member of some stable, minimal model set greater than $\mathcal{M}$.

Corollary 4.5 *g is in every extension iff g is true in every minimal model.*

> **Proof idea:** By construction, a minimal non-empty model set less than $\mathcal{M}$ is the set of models of an extension. □

4.4 Epistemic Semantics

Lin and Shoham [Lin and Shoham, 1990] have developed an epistemic semantics for both Default Logic and Autoepistemic Logic. They use a modal logic with two operators, K ("known") and A ("assumed"). Given a standard Kripke semantics of these symbols, they develop a preference over Kripke interpretations.

Kripke interpretation M_1 is **preferred** over M_2, written $M_1 \sqsubset M_2$, if $\{\phi : M_1 \models A\phi\} = \{\phi : M_2 \models A\phi\}$[4] and $\{\phi : M_1 \models K\phi\} \subset \{\phi : M_2 \models K\phi\}$.

Kripke interpretation M is a **preferred model** of formulae S if M is a minimal model of S and $\{\phi : M \models K\phi\} = \{\phi : M \models A\phi\}$.

They form a logic GK, with entailment defined by $\Phi \models_{GK} \phi$ if ϕ holds in all preferred models of Φ.

[4] The ϕ range over base formulae.

Default theory $\Delta = \langle D, F \rangle$ is translated into Δ_{GK} as follows. If $p \in F$ then $Kp \in \Delta_{GK}$. Default

$$\frac{p : q_1, \cdots, q_n}{r}$$

is translated into Δ_{GK} as

$$Kp \wedge \neg A \neg q_1 \wedge \cdots \wedge \neg A \neg q_n \Rightarrow KR$$

The following theorem is proved in [Lin and Shoham, 1990].

Theorem 4.6 *A consistent set of sentences E is a default extension of Δ iff there is a preferred model M of Δ_{GK} such that $E = \{\phi : M \models K\phi\}$*

4.5 Sceptical Prediction

A simpler form of semantics can be obtained for sceptical prediction for the case of normal defaults without preconditions. In this case there are no "provability" conditions that we need to be concerned about.

In this case [Poole, 1988b], an explanation of g from $\langle D, F \rangle$ is a set C of instances of conclusions of defaults such that $F \cup C \models g$ and $F \cup C$ is consistent.

Theorem 4.7 If all of the defaults are normal defaults without prerequisites, the following are equivalent:

1. g is in all extensions of $\langle D, F \rangle$.

2. There is a set $\mathcal{C}$ of explanations of g from $\langle D, F \rangle$ such that $\bigvee_{C_i \in \mathcal{C}} \neg C_i$ cannot be explained from $\langle D, F \rangle$.

3. g is true in all minimal models of F, where the ordering on interpretations is defined by $M_1 <_D M_2$ if the assumptions violated in M_1 are a subset of the assumptions violated by M_2. That is, if

$$\{d \in D' : M_1 \models \neg d\} \subset \{d \in D' : M_2 \models \neg d\}$$

   where D' is the set of ground instances of conclusions of defaults in D.

Proof: $1 \Rightarrow 2$. Let $\mathcal{C}$ be the set of all explanations of g. If C is an explanation of $\bigvee_{C_i \in \mathcal{C}} \neg C_i$, then C can be extended to an extension E, in which g goes not appear. Thus if g is in all extensions, no such E exists.

$2 \Rightarrow 1$. Suppose 2 is true. Given such a $\mathcal{C}$, every extension contains one element of $\mathcal{C}$ (otherwise the extension is an explanation of the disjunct in 2). g follows from $F \cup C$, for all $C \in \mathcal{C}$ thus g is in every extension.

$3 \Rightarrow 1$. Suppose g is not in extension E. E is consistent and does not entail g, so there is a model M of $E \wedge \neg g$. M is a model of F, as $F \subseteq E$. M is minimal, as if there is some $M' < M$, there is some $d \in D'$ such that $d \notin E$, d is consistent with E (as M' is a model of $E \wedge d$), which is a contradiction to the maximality of the extension E. Thus g is not true in all minimal models.

$1 \Rightarrow 3$. Suppose g is not true in minimal model M. Let E be the set of consequences of $F \cup \{d \in D' : M \models d\}$. E is an extension, as E is consistent (M is a model of E), and if some $d \in D'$, $d \notin E$, then $E \models \neg d$ (otherwise $E \wedge d$ has a model M', in which case $M' < M$, a contradiction to the minimality of M). g is not in extension E (as it is not a consequence of E, as it is false in a model of E). $\Box$

The minimal model definition of theorem 4.7 is subtly but importantly different to the minimal models definition of circumscription [McCarthy, 1980; Lifschitz, 1985], with no fixed predicates. We are minimising over the syntactic forms of the models (the sets we are comparing are sets of ground atomic formulae). In Circumscription, the minimization is in the semantic domain (minimising over individuals rather than over constants). We do not require the unique names assumptions (as, for example, the violation set $\{ab(a), ab(b)\}$ cannot be reduced by making $a = b$). This syntactic minimisation is also why we can minimise equality; the minimisation occurs before the terms have assigned to individuals. We can thus affect this assignment. When minimising in the semantic domain, the minimisation occurs after terms have been assigned to individuals; thus the minimisation cannot affect equality [Etherington *et al.*, 1985], and the unique names hypothesis is needed (as, for example, the violation set $\{ab(a), ab(b)\}$ can be reduced by making $a = b$).

Theorem 4.7 can be traced to a number of sources. If we let 3' be the circumscriptive version of 3, which are the same under the unique names and domain closure assumptions. $1 \Leftrightarrow 3'$ is due to Etherington [Etherington, 1987b; Etherington, 1987a]. $1 \Leftrightarrow 2$ is due to Poole [Poole, 1989a]. $2 \Leftrightarrow 3'$ is due to Przymusinski [Przymusinski, 1989] and Ginsberg [Ginsberg, 1989]. The form of 3 presented here adapted from Geffner [Geffner, 1989], by removing the priorities. $1 \Leftrightarrow 3$, as far as I know, is new to this paper.

5 Relationship to other Formalisms

5.1 Autoepistemic Logic

Autoepistemic logic [Moore, 1985] is a logic that allows an agent to reason about their own knowledge and ignorance. Konolige [Konolige, 1987] has shown the equivalence between closed defaults and a form of "strongly grounded" autoepistemic logic.

Konolige showed that every autoepistemic formula is equivalent to a conjunction of formulae in the form

$$(L\alpha \wedge \neg L \neg \beta_1 \wedge ... \wedge \neg L \neg \beta_m) \Rightarrow w$$

where $L\phi$ is read as ϕ is known, and that this, interpreted in terms of strongly grounded extensions, is the same as the default

$$\frac{\alpha : \beta_1, ..., \beta_m}{w}.$$

See chapter ?? of this handbook for details on the form of autoepistemic logic that is equivalent to default logic.

5.2 Circumscription

Circumscription [McCarthy, 1980; Lifschitz, 1985; McCarthy, 1986] is a way to specify that the only exceptions to a formula are those that are known (see chapter ??? of this handbook for full details). It is generally given [McCarthy, 1986] as a second order formula. Circumscription can also be defined semantically as determining what is true in all of the models that minimise the extent of some formula [McCarthy, 1980; Lifschitz, 1985].

Etherington [Etherington, 1987b; Etherington, 1987a] has shown the relationship between default logic and circumscription. He shows that minimizing the predicate P, with all other predicates varying, in circumscription corresponds to the use of the default $\frac{:\neg P(x)}{\neg P(x)}$ as long as the facts entail domain closure and unique names axioms. See the discussion in section 4.5.

Some major differences are that circumscription cannot be used to affect equality [Etherington, 1987a, Theorems 5.7 and 5.8], and so cannot be used for the examples of section 3.3 in this chapter.

One of the things that circumscription can do which default logic, as defined here, cannot do is to conclude universal conclusions. For example, by minimising $p(x)$, but knowing $p(a)$, circumscription can conclude

$$\forall x\; x \neq a \Rightarrow \neg p(x).$$

Default logic cannot conclude the universal formula, but can only conclude $\neg p(t)$ for each ground term t that is different to a. Poole [Poole, 1987] has investigated the problem of allowing syntactically more general instances of defaults. Lifschitz [Lifschitz, 1990] has suggested that the variables in open defaults should refer to individuals in the domain rather than being syntactic. He has developed a version of default logic that allows for universal conclusions, but at the expense of not being able to affect equality.

6 Implementation

In this section we discuss a number of proposals of how default reasoning can be implemented. The main problem with implementing default logic is that we need to check consistency of the antecedents with an extension. There are two main approaches that have been suggested.

1. The first is by "forward-chaining" to produce an extension. This can only be used where the extensions are finitely representable, for example, in the propositional case.

2. The second is to "backward-chaining" to determine whether some proposition is in some extension (or in all extensions). This has been suggested in the semi-monotonic cases (adding defaults cannot remove

conclusions) where consistency with only the defaults used is adequate for consistency with an extension.

6.1 Forward Chaining Default Prover

A forward-chaining default prover [Etherington, 1987a] follows the definition of an extension. We non-deterministically choose a default whose precondition has been derived, and whose justifications are consistent with what has been derived (either by finding a model, or by finite failure using a complete inference system), and then infer the consequent of the default. This choice may have to be undone if some justifications used are not consistent with some subsequent default.

Junker and Konolige [Junker and Konolige, 1990] have developed a proof procedure based on translating default logic into a truth maintenance system. They develop a finite representation for extensions of closed default theories, and develop a correspondence between the fixed point of a TMS and the representation for an extension.

6.2 Backward Chaining Default Prover

In this section we show how a theorem prover (see e.g., [Chang and Lee, 1973]) can be used to determine whether some formula can be explained using normal defaults. This is based on Reiter's implementation [Reiter, 1980, section 7.2].

The algorithm for normal defaults can be given the following abstract definition to explain g from $\langle D, F\rangle$:

1. Try to prove g from $F \cup CONS(D)$, where $CONS(D)$ is the set of consequents of rules in D. Make R the set of instances of rules in D used in the proof.

2. Ground R (substitute a new constant for each of the free variables in R). We thus have created a ground proof of g from $A \cup CONS(R)$.

3. Using the same algorithm explain the precondition for each default in R. Let R^+ be the union of all of the defaults used in the derivation of g and all preconditions of defaults.

4. Try to prove $A \wedge CONS(R^+)$ is inconsistent. If a complete search fails to find a proof of inconsistency, R^+ is an explanation for g.

This algorithm was first given by Reiter [Reiter, 1980, section 7.2]. [Poole *et al.*, 1987] gives a Prolog interpreter for normal defaults without prerequisites. [Poole, 1991a] shows how explanation can be implemented by compiling into Prolog.

6.3 Sceptical Prediction

The above procedures considered whether some proposition is in some extension. We should also consider the question of whether some proposition is in all extensions. The idea behind implementing this [Poole, 1988a; Kautz and Selman, 1989] is that proposition g is in all extension if it is in an extension even when an adversary can choose the default. g is not in all extensions if there is an extension which does not contain g; if we can show that an adversary cannot generate such an extension, then g must be in all extensions.

For the forward chaining default provers, to determine if g is in all extensions we try to generate an extension in which g does not appear. When there is a choice of which default to choose, we let an adversary choose the default. If an adversary can generate an extension which does not contain g, then g is not in all extensions. If we can demonstrate that there are no choices for the adversary which lead to an extension not containing g, then g is in all extensions.

For the backward chaining default provers, we use the results of theorem 4.7. These have only been built for normal defaults without prerequisites.

This result can be transformed into a procedure to compute membership in all extensions [Poole, 1989a; Ginsberg, 1989; Przymusinski, 1989]. If we assume that we can compute explanations (section 6.2), then we find explanations of g and then fail to find explanations for the negation of the disjunct of the justifications of the explanations of g.

Example 6.1 Consider the following example[5]

$$D \quad = \{ \quad \frac{: rh(x)}{rh(x)}, \frac{: qd(x)}{qd(x)}, \frac{: hs(x)}{hs(x)},$$

[5]This example is based on an example by Matt Ginsberg, which is based on an example due to Ray Reiter.

$$
\begin{aligned}
& \frac{: hp(x)}{hp(x)}, \frac{: dp(x)}{dp(x)}, \frac{: qr(x)}{qr(x)}\} \\
F \quad = \{ \; & \forall x \; republican(x) \wedge rh(x) \Rightarrow hawk(x), \\
& \forall x \; quaker(x) \wedge qd(x) \Rightarrow dove(x), \\
& \forall x \; hawk(x) \wedge hs(x) \Rightarrow support\text{-}star\text{-}wars(x), \\
& \forall x \; hawk(x) \wedge hp(x) \Rightarrow politically\text{-}motivated(x), \\
& \forall x \; dove(x) \wedge dp(x) \Rightarrow politically\text{-}motivated(x) \\
& \forall x \; quaker(x) \wedge qr(x) \Rightarrow religious(x)\} \\
& \forall x \; \neg(dove(x) \wedge hawk(x)), \\
& quaker(dick), \\
& republican(dick) \; \}
\end{aligned}
$$

Consider the process of trying to determine $support\text{-}star\text{-}wars(dick)$. There is one explanation for it namely,

$$F \cup \{rh(dick), hs(dick)\}$$

There is one set of ground instances of defaults which, if an adversary had chosen, would make this argument inapplicable:

$$F \cup \{qd(dick)\}$$

Thus $pro\text{-}star\text{-}wars(dick)$ is not in all extensions.

Consider determining $politically\text{-}motivated(dick)$. There are two explanations for it:

$$F \cup \{qd(dick), dp(dick)\}$$

$$F \cup \{rh(dick), hp(dick)\}$$

There is no explanation for the negation of the disjunction of the explanations

$$F \wedge ((qd(dick) \wedge dp(dick)) \vee (rh(dick) \wedge hp(dick)))$$

and so $politically\text{-}motivated(dick)$ is in all extensions.

7 When does an extension exist?

One of the most important questions to investigate is when we can guarantee an extension exists. This question has been investigated by Etherington [Etherington, 1987a, pp. 84-88].

Etherington noticed that the only time there is no extension is when there are defaults such that the justification of one default is inconsistent with the consequent of a default that must subsequently be applied.

Example 7.1 The following has no extensions:

$$\begin{aligned} D &= \{ \frac{a : b \wedge c}{c}, \frac{c : \neg b}{\neg b} \} \\ F &= \{ a \} \end{aligned}$$

Notice here that the justification of the first default is denied by the second default and the second default must be applied after the first.

Etherington [Etherington, 1987a] defined a notion of **ordered** default theories that disallows the above sort of circularity, and showed that every ordered, semi-normal default theory has at least one extension.

8 Complexity

Default logic, in general, is not even semi-decidable. Normal logical consequence is semi-decidable, but when the proof can involve a consistency check, the resulting system may be (depending on the underlying logic) undecidable.

When we consider propositional default logic, determining whether some proposition follows from some facts and default rules is decidable, but NP-complete.

Kautz and Selman [Kautz and Selman, 1989] have investigated the complexity of propositional, disjunction-free default logic under various syntactic restrictions. They considered the three problems:

1. finding an extension;
2. determining if a given proposition is true in some extension and
3. determining if a given proposition is true in all extensions.

They found that the three kinds of problems are strictly ordered in difficulty.

The first and simplest, finding an extension, is NP-complete for unary defaults (defaults with a single un-negated precondition, a literal consequent, and a justification that is either just the consequent or, if the consequent is positive, the consequent and a single negative literal). If this restriction is strengthened to be disjunction-free, ordered theory there is an $O(n^2)$ algorithm.

Determining if a given literal is in any extension is NP-hard for either disjunction free normal theories or unary ordered theories. There is, however, an $O(n)$ algorithm for to determine if a given literal appears in any extension of a Horn default theory (normal defaults with only positive literals in the prerequisite).

For the most difficult of these problems, it is co-NP-hard to determine if a given literal appears in every extension even for Horn default theories. There is an $O(n^3)$ algorithm to determine if a given literal appears in every extension of a normal unary theory.

9 Empirical Investigations

Theorist, developed by Poole and Goebel and their associates [Poole *et al.*, 1987; Poole, 1988a] is an empirical framework in which to investigate logic-based hypothetical reasoning. It is an attempt to test the conjecture that commonsense reasoning can be captured by allowing user specified hypotheses that can be used in an argument if consistent. Theorist is deliberately the simplest hypothetical reasoning system based on first order predicate calculus. The user-specified hypotheses can either be treated as normal defaults without prerequisites for default reasoning, as possible causes in abductive reasoning or as possible components in design.

Towards the goal of an empirical framework for default and abductive reasoning, efficient implementations [Poole *et al.*, 1987; Poole, 1991a] have been built, representational methodologies [Poole, 1990] (discussing how to "program" the logic) and various applications have been developed (see next section). In an effort to push the framework to its limits, and to enable refutable hypotheses, much of the work has concentrated on showing how to use the system to solve problems.

Results that have come from the empirical investigation have included how the "lottery paradox" naturally arises in default reasoning [Poole, 1991b], and how a system that just considers the logical content of the facts, and has modular defaults, cannot have an automatic preference for more specific defaults [Poole, 1991b]. The latter can be seen, in the propositional case, by considering the defaults "mammals don't fly" and "bats fly" with the facts being *bat* and $bat \Rightarrow mammal$. By specificity we would want to conclude fly. The facts are logically equivalent to the facts *mammal* and $mammal \Rightarrow bat$, from which, by specificity we would conclude $\neg bat$. Thus the logical content of the facts do not convey all the information necessary for the default conclusion.

10 Applications

Default logic has been used to formalise many applications.

Etherington and Reiter [Etherington and Reiter, 1983; Etherington, 1987a] and Poole [Poole, 1988a] both discuss how inheritance hierarchies can be formalised in default logic. Etherington and Reiter uses the semi-normal default

$$\frac{A(x) : B(x) \wedge \neg C_1(x) \neg ... \neg C_n(x)}{B(x)}$$

to represent an arc from node A to (possibly negated) node B that is cancelled by arcs from nodes $C_1 ... C_n$. Poole [Poole, 1985] gave a proposal to automatically prefer more specific arguments over more general ones. This proposal has been further developed by Simari and Loui [Simari and Loui, 1990]

Reiter [Reiter, 1987] and Poole [Poole, 1988b; Poole, 1989b] show how default logic can be used for diagnosis. In particular, normality of components is assumed as a default. Abnormalities in an extension correspond to a diagnosis. When there are possible faults we have the absence of faults as a default. Faults are only concluded if needed.

Poole [Poole, 1990] discusses applications in diagnosis, image interpretation, plan recognition and commonsense reasoning.

Jones and Tubman [Jones and Poole, 1985; Tubman, 1986] have used a version of Theorist for the diagnosis of children with learning disabilities.

Goebel and Goodwin [Goebel and Goodwin, 1987] show how default logic can be used for planning and temporal prediction. They have considered how chronological minimization can be implemented in Theorist.

In natural language, Mercer [Mercer, 1987; Mercer, 1990] and Csinger [Csinger and Poole, 1989] have considered how default logic can be used to formalise the problem of presupposition in natural language. Perrault [Perrault, 1987] has used default logic for a formalization of speech act theory. Van Arragon [van Arragon, 1990] uses nested default reasoning for user modelling. Dunin-Keplitz [Dunin-Keplitz, 1984] has used default reasoning for resolution of anaphora. Saint-Dizier has used default logic for formalizing generalized quantifiers [Saint-Dizier, 1988].

11 Variations

One of the features of a reasoning formalism is how naturally it permits variations. Here we only give pointers to these variations, rather than describing them in detail.

Łukaszewicz [Łukaszewicz, 1988] gives an alternate definition of extension for Reiter's defaults. In Łukaszewicz's system there is always an extension for a set of defaults. Extensions are defined in such a manner that justifications cannot be denied by "subsequent" defaults. Justifications of defaults have to be consistent with the justifications and consequents of all of the defaults in the extension. In a similar idea, Poole [Poole, 1988a] has proposed the idea of constraints that have to be consistent with default conclusions, but are not part of the default conclusions.

In response to examples such as examples 3.3, 3.4, 3.6 and 3.8 in section 3 (first raised in [Poole, 1988a], [Makinson, 1988] and [Poole, 1989c]), there have been some recent proposals to allow for case analysis in preconditions and justifications of defaults [Brewka, 1991; Delgrande and Jackson, 1991; Gelfond *et al.*, 1991; Schaub, 1991]. There is a close relationship amongst these proposals as well as to the use of constraints in [Poole, 1988a].

Poole [Poole, 1985] discusses how a preference for explanations can be used for the automatic preference of more specific defaults over more general defaults. Goebel and Goodwin [Goebel and Goodwin, 1987] show how "chronological minimization" (preferring defaults that occur earlier in time), as explanation preference, can be used for temporal reasoning. Neufeld and

Poole [Neufeld and Poole, 1987] discuss one way to incorporate probability into default reasoning.

Jones and Poole [Jones and Poole, 1985] and Brewka [Brewka, 1989b] show, in quite different ways, how we can do default reasoning in hierarchies of defaults.

Goodwin and Goebel [Goodwin and Goebel, 1988] show how we often want to produce conditional explanations in planning applications.

Acknowledgements

This work could not have been done without the ideas of Ray Reiter, David Etherington, and Randy Goebel. Thanks to David Etherington, David Makinson and Kave Eshghi for comments on earier drafts of this paper. This research was supported under NSERC grant OGPOO44121.

References

[Besnard, 1989] P. Besnard. *An Introduction to Default Logic.* Symbolic Computation. Springer-Verlag, Berlin, 1989.

[Brewka, 1989a] G. Brewka. *Nonmonotonic Reasoning: from Theoretical Foundations Towards Efficient Computation.* PhD thesis, Universität Hamburg, Hamburg, Germany, 1989.

[Brewka, 1989b] G. Brewka. Preferred subtheories: An extended logical framework for logical reasoning. *Proc. 11th International Joint Conf. on Artificial Intelligence (IJCAI-89)*, pages 1043–1048, 1989.

[Brewka, 1991] G. Brewka. Cumulative default logic: in defence of nonmonotonic inference rules. *Artificial Intelligence*, 50(2):183–205, July 1991.

[Chang and Lee, 1973] C. L. Chang and R. C. T. Lee. *Symbolic Logical and Mechanical Theorem Proving.* Academic Press, New York, 1973.

[Csinger and Poole, 1989] Andrew Csinger and David Poole. From utterance to belief: Default reasoning in user-modelling. In *Proceedings of the Conference for Knowledge Based Computing Systems, KBCS-89*, pages 408–419, Bombay, India., December 1989.

[Delgrande and Jackson, 1991] J. P. Delgrande and W. K. Jackson. Default logic revisited. In *Proc. Second International Conf. on Principles of Knowledge Representation and Reasoning*, pages 118–127, Cambridge, MA, April 1991.

[Dunin-Keplitz, 1984] B. Dunin-Keplitz. Default reasoning in anaphora resolution. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-84)*, Pisa, Italy, 1984.

[Enderton, 1972] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, Orlando, FL, 1972.

[Etherington and Reiter, 1983] D. W. Etherington and R. Reiter. On inheritance hierarchies with exceptions. In *Proc. 3rd National Conference on Artificial Intelligence*, pages 104–108, Washington, D. C., August 1983.

[Etherington *et al.*, 1985] D. W. Etherington, R. E. Mercer, and R. Reiter. On the adequacy of predicate circumscription for closed-world reasoning. *Computational Intelligence*, 1(1):11–15, 1985.

[Etherington, 1987a] D. W. Etherington. *Reasoning with Incomplete Information*. Research Notes in Artificial Intelligence. Pitman, London, 1987.

[Etherington, 1987b] D. W. Etherington. Relating default logic and circumscription. In *Proc. 11th International Joint Conf. on Artificial Intelligence (IJCAI-89)*, pages 489–494, Milan, Italy, August 1987.

[Etherington, 1987c] D. W. Etherington. A semantics for default logic. In *Proc. 10th International Joint Conf. on Artificial Intelligence (IJCAI-87)*, pages 495–498, Milan, Italy, August 1987.

[Geffner, 1989] H. Geffner. *Default Reasoning: Causal and Conditional Theories*. PhD thesis, Department of Computer Science, UCLA, Los Angeles, CA, November 1989.

[Gelfond and Lifschitz, 1988] M. Gelfond and V Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. Bowen, editors, *Proceedings of the Fifth Logic Programming Symposium*, pages 1070–1080, Cambridge, MA, 1988.

[Gelfond *et al.*, 1991] M. Gelfond, H. Przymusinska, V. Lifschitz, and M. Truszczynski. Disjunctive defaults. In *Proc. Second International Conf. on Principles of Knowledge Representation and Reasoning*, pages 230–237, Cambridge, MA, April 1991.

[Ginsberg, 1989] M. L. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39(2):209–230, June 1989.

[Goebel and Goodwin, 1987] R. G. Goebel and S. D. Goodwin. Applying theory formation to the planning problem. In *Proc. of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, pages 207–232. Morgan Kaufmann, 1987.

[Goodwin and Goebel, 1988] S. D. Goodwin and R. G. Goebel. Non-monotonic reasoning in temporal domains: the knowledge independence problem. In M. Reinfrank et al., editors, *Non-Monotonic Reasoning, Proceedings of the 2nd International Workshop*, pages 187–201. Springer-Verlag, Grassau, Germany, June 1988.

[Jones and Poole, 1985] M. Jones and D. Poole. An expert system for educational diagnosis based on default logic. In *Fifth International Workshop on Expert Systems and their Applications*, pages 673–683, Avignon, France, May 1985.

[Junker and Konolige, 1990] U. Junker and K. Konolige. Computing the extensions of autoepistemic and default logics with a truth maintenance system. In *Proc. 8th National Conference on Artificial Intelligence*, pages 278–283, Boston, MA, July 1990.

[Kautz and Selman, 1989] H. A. Kautz and B. Selman. Hard problems for simple default logics. In R. Brachman, H. Levesque, and R. Reiter, editors, *Proc. First International Conf. on Principles of Knowledge Representation and Reasoning*, pages 189–197, Toronto, Canada, May 1989.

[Konolige, 1987] K. Konolige. On the relation between default and autoepistemic logic. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 195–226. Morgan Kaufmann, Los Altos, CA, 1987.

[Lifschitz, 1985] V. Lifschitz. Computing circumscription. In *Proc. 9th International Joint Conf. on Artificial Intelligence (IJCAI-85)*, pages 121–127, Los Angeles, CA, August 1985.

[Lifschitz, 1990] V. Lifschitz. On open defaults. In *Proc. ESPRIT Symposium on Computational Logic*, Brussels, 1990.

[Lin and Shoham, 1990] F. Lin and Y. Shoham. Epistemic semantics for fixed-point non-monotonic logics. Technical Report STAN-CS-90-1322, Dept. of Computer Science, Stanford University, CA, 1990.

[Łukaszewicz, 1985] W. Łukaszewicz. Two results on default logic. In *Proc. 9th International Joint Conf. on Artificial Intelligence (IJCAI-85)*, pages 459–461, Los Angeles, CA, August 1985.

[Łukaszewicz, 1988] W. Łukaszewicz. Considerations on default logic: an alternative approach. *Computational Intelligence*, 4(1):1–16, February 1988.

[Makinson, 1988] D. Makinson. General theory of cumulative inference. In M. Reinfrank et al., editors, *Non-Monotonic Reasoning, Proceedings of the 2nd International Workshop*, pages 1–18. Springer-Verlag, Grassau, FRG, June 1988.

[McCarthy, 1980] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2):27–39, 1980.

[McCarthy, 1986] J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28(1):89–116, February 1986.

[Mercer, 1987] Robert Mercer. A default logic approach to the derivation of natural language presuppositions. Technical Report 35, University of British Columbia, October 1987.

[Mercer, 1990] R. E. Mercer. Deriving natural language presuppositions from complex conditionals. In P. F. Patel-Schneider, editor, *Proc. 8th Canadian Artificial Intelligence Conf.*, pages 114–120, Ottawa, Ont., May 1990.

[Moore, 1985] R. C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, January 1985.

[Neufeld and Poole, 1987] E. M. Neufeld and D. Poole. Towards solving the multiple extension problem: combining defaults and probabilities. In *Proc. Third Workshop on Reasoning with Uncertainty*, pages 305–312, Seattle, July 1987.

[Perrault, 1987] C. R. Perrault. An application of default logic to speech act theory. CSLI 90, Center for the Study of Language and Information, Stanford, CA, 1987.

[Poole *et al.*, 1987] D. Poole, R. Goebel, and R. Aleliunas. Theorist: A logical reasoning system for defaults and diagnosis. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352. Springer-Verlag, New York, NY, 1987.

[Poole, 1985] D. Poole. On the comparison of theories: Preferring the most specific explanation. In *Proc. 9th International Joint Conf. on Artificial Intelligence (IJCAI-85)*, volume I, pages 144–147, Los Angeles, CA, August 1985.

[Poole, 1987] D. Poole. Variables in hypotheses. In *Proc. 10th International Joint Conf. on Artificial Intelligence (IJCAI-87)*, pages 905–908, Milan, Italy, August 1987.

[Poole, 1988a] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1988.

[Poole, 1988b] D. Poole. Representing knowledge for logic-based diagnosis. In *International Conference on Fifth Generation Computing Systems*, pages 1282–1290, Tokyo, Japan, November 1988.

[Poole, 1989a] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[Poole, 1989b] D. Poole. Normality and faults in logic-based diagnosis. In *Proc. 11th International Joint Conf. on Artificial Intelligence (IJCAI-89)*, pages 1304–1310, Detroit, August 1989.

[Poole, 1989c] D. Poole. What the lottery paradox tells us about default reasoning. In *First International Conference on the Principles of Knowledge*

Representation and Reasoning (KR89), pages 333–340, Toronto, Canada, May 1989.

[Poole, 1990] D. Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, December 1990.

[Poole, 1991a] D. Poole. Compiling a default reasoning system into Prolog. *New Generation Computing Journal*, 9(1):3–38, 1991.

[Poole, 1991b] D. Poole. The effect of knowledge on belief: conditioning, specificity and the lottery paradox in default reasoning. *Artificial Intelligence*, 49:281–307, 1991.

[Przymusinski, 1989] T. C. Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38(1):49–73, February 1989.

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

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

[Saint-Dizier, 1988] P. Saint-Dizier. Default logic, natural language and generalized quantifiers. In *Proc. Conference on Computational Linguistics (COLING-88)*, pages 555–560, Budapest, 1988.

[Schaub, 1991] T. Schaub. Assertional default theories: A semantical view. In *Proc. Second International Conf. on Principles of Knowledge Representation and Reasoning*, pages 496–506, Cambridge, MA, April 1991.

[Shoham, 1987] Y. Shoham. Nonmonotonic logics: Meaning and utility. In *Proc. 10th International Joint Conf. on Artificial Intelligence (IJCAI-87)*, pages 388–393, Milan, August 1987.

[Simari and Loui, 1990] G. R. Simari and R. P. Loui. Confluence of argument systems: Poole's rules revisited. In K. Konolige, editor, *Preprints of the Third International Workshop on Nonmonotonic Reasoning*, pages 223–232, South Lake Tahoe, CA, May 1990.

[Tubman, 1986] J. B. Tubman. An expert system for educational diagnosis using theorist. Master's thesis, University of Waterloo, Waterloo, Ontario, 1986.

[van Arragon, 1990] P. van Arragon. Nested default reasoning with priority levels. In P. F. Patel-Schneider, editor, *Proc. 8th Canadian Artificial Intelligence Conf.*, pages 77–83, Ottawa, Ontario, May 1990.