The Role of Default Logic in Knowledge Representation

James P. Delgrande (jim@cs.sfu.ca)
Simon Fraser University

Torsten Schaub (torsten@cs.uni-potsdam.de)
Universität Potsdam

Abstract. Various researchers in Artificial Intelligence have advocated formal logic as an analytical tool and as a formalism for the representation of knowledge. Our thesis in this paper is that commonsense reasoning frequently has a nonmonotonic aspect, either explicit or implicit, and that to this end Default Logic (DL) provides an appropriate elaboration of classical logic for the modelling of such phenomena. That is, DL is a very general, flexible, and powerful approach to nonmonotonic reasoning, and its very generality and power makes it suitable as a tool for modelling a wide variety of applications.

We propose a general methodology for using Default Logic, involving the naming of default rules and the introduction of special-purpose predicates, for detecting conditions for default rule applicability and controlling a rule's application. This allows the encoding of specific strategies and policies governing the set of default rules. Here we show that DL can be used to formalise preferences among properties and the inheritance of default properties, and so we essentially use DL to axiomatise such phenomena.

1. Introduction

First-order logic (FOL) has long and often been advocated as an appropriate tool for formalising knowledge about particular domains and for the analysis of various systems and approaches. The role of logic in Artificial Intelligence (AI) and Knowledge Representation is discussed in, for example, (Moore, 1982; Hayes, 1985b); in addition, many introductory AI texts assume or promote the centrality of formal logic in Knowledge Representation and Reasoning. Among other domains, logic has been used to formalise knowledge about liquids (Hayes, 1985a), time (Allen and Hayes, 1989), actions and planning (McCarthy and Hayes, 1969; Levesque *et al.*, 1998), and concepts, as well as modal notions concerning possible worlds, knowledge, belief, etc. (McCarthy, 1979; Moore, 1980). It has been used to analyse, for instance, relational database systems (Reiter, 1984) and assumption-based truth maintenance systems (Reiter and de Kleer, 1987). Default and diagnostic reasoners have been built via specifying how logic is to be used (Poole, 1988).

Our thesis in this paper is that commonsense reasoning frequently has a nonmonotonic aspect, and that to this end Default Logic is an appropriate elaboration of classical logic for the modelling of such phenomena. As we discuss in the next section, Default Logic has found wide application in AI, most obviously in the direct encoding of default information, but also in areas ranging from database theory to natural language understanding. To some extent our theme that Default Logic can be used as a general knowledge representation formalism is implicit in the quantity and breadth of approaches that make use of it. However we further argue this point by suggesting a methodology for Default Logic in which Default Logic appears more broadly applicable to various diverse phenomena than might be otherwise suspected.

The general framework is quite straightforward. We begin with a language for expressing some phenomenon in which we are interested, along with a notion of what inferences should obtain in this language. A translation is given, such that the original theory is mapped into a (standard, Reiter) default theory, wherein the desired inferences provably obtain. This translation serves two purposes. It provides a semantics for the original theory, in that it demonstrates how the original theory is expressible in Default Logic. Second, it provides a direction for implementation: given a modular translation into Default Logic and a (presumed) Default Logic theorem prover, it is straightforward to implement the desired inferences of the original theory. In one example developed here, we show how a preference ordering on properties can be mapped into a (standard, Reiter) default theory, such that one can determine the highest-ranked, consistently-assumable properties.

We do not provide a specific encoding and translation, but rather we outline a methodology by means of which one may carry out such an encoding and translation. To illustrate, consider a general default rule $\frac{\alpha : \beta}{\gamma}$ with informal meaning that if α is true and β is consistent with a set of beliefs then γ is believed. Assume that constant n is associated with this rule as its name in the object theory. Then the rule $\frac{\neg\beta :}{\mathsf{bl}(n)}$ will be applied just when the original rule cannot be applied due to its justification being not consistent with the set of beliefs. $\mathsf{bl}(n)$ is a newly-introduced predicate; the concluding of $\mathsf{bl}(n)$ signals in the object theory that the original rule is not applied. Similarly, if we replace the rule $\frac{\alpha : \beta}{\gamma}$ by $\frac{\mathsf{ok}(n) \wedge \alpha : \beta}{\gamma}$, where $\mathsf{ok}(\cdot)$ is a newly-introduced predicate, then application of this rule can be controlled in the object theory, in that the rule cannot be applied unless $\mathsf{ok}(n)$ has been asserted. This notion of adding "tags" to detect and control rule application yields surprisingly powerful results. Using this technique, we have shown in (Delgrande and Schaub, 1997a) how an ordered default theory, consisting of a theory with a partial order on the default rules, can be translated into a "standard" default theory in which the rule ordering is adhered to. Hence, among other things, we show that an explicit ordering on a set of default rules adds nothing to the overall expressibility of Default Logic.

The paper is organized as follows. Following a brief introduction to Default Logic and a discussion of approaches that have employed it, we summarise the basic components of our approach. We then illustrate the applicability of the approach in two "modelling exercises". First, we review our earlier and most basic approach to dealing with preferences on rules. Second, we show how Default Logic can be used to implement a mechanism for default property inheritance. The overall theme is that by using Default Logic we can axiomatise various diverse phenomena. Hence we suggest that, in addition to directly representing nonmonotonic theories, Default Logic is appropriate as a general AI formalism in which specific phenomena may be encoded. Thus, via various examples we show how Default Logic can be employed to provide a semantics for such phenomena and, on the other hand, provide an encoding for reasoning with such phenomena.

2. Default Logic

Default logic (Reiter, 1980) augments classical logic by *default rules* of the form $\frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma}$ where α, $\beta_1, \ldots, \beta_n$, γ are sentences of first-order or propositional logic. Here we deal with *singular* defaults for which $n = 1$. A singular rule is *normal* if β is equivalent to γ; it is *semi-normal* if β implies γ. (Janhunen, 1999) shows that any default rule can be transformed into a set of semi-normal defaults. We sometimes denote the *prerequisite* α of a default δ by $\mathit{Prereq}(\delta)$, its *justification* β by $\mathit{Justif}(\delta)$, and its *consequent* γ by $\mathit{Conseq}(\delta)$. Accordingly, $\mathit{Prereq}(D)$ is the set of prerequisites of all default rules in D; $\mathit{Justif}(D)$ and $\mathit{Conseq}(D)$ are defined analogously. Empty components, such as no prerequisite or even no justifications, are assumed to be tautological. *Open defaults* with unbound variables are taken to stand for all corresponding instances. A set of default rules D and a set of formulas W form a *default theory* (D, W) that may induce a single, multiple, or even zero *extensions* in the following way.

DEFINITION 2.1. *Let (D, W) be a default theory and let E be a set of formulas. Define $E_0 = W$ and for $i \geq 0$:*

$$\begin{aligned} D_i &= \left\{ \frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma} \in D \,\middle|\, \alpha \in E_i, \neg\beta_1 \notin E, \ldots, \neg\beta_n \notin E \right\} \\ E_{i+1} &= \mathit{Th}(E_i) \cup \{ \mathit{Conseq}(\delta) \mid \delta \in D_i \} \end{aligned}$$

Then E is an extension for (D, W) iff $E = \bigcup_{i=0}^{\infty} E_i$.

Any such extension represents a possible set of beliefs about the world at hand. The above procedure is not constructive since E appears in the specification of E_{i+1}. We define $GD(D, E) = \bigcup_{i=0}^{\infty} D_i$ as the set of *generating defaults* of extension E. An enumeration $\langle \delta_i \rangle_{i \in I}$ of default rules is *grounded* in a set of formulas W, if we have for every $i \in I$ that $W \cup \mathit{Conseq}(\{\delta_0, \ldots, \delta_{i-1}\}) \vdash \mathit{Prereq}(\delta_i)$.

2.1. Applications of Default Logic

The most obvious and direct use of Default Logic is in the "brute force" encoding of default information. Thus "penguins normally do not fly" may be represented by the rule $\frac{P : \neg F}{\neg F}$ and "birds normally fly" may be represented by $\frac{B : F}{F}$. Interactions between defaults are most directly handled by encoding explicitly what happens in various cases (Reiter and Criscuolo, 1981). Thus, we replace our second default above with $\frac{B : F \wedge \neg P}{F}$ so that if we know that some bird is also a penguin, only the first default can be applied.

In (Etherington and Reiter, 1983) inheritance networks are translated into default theories. Strict links are mapped to universally quantified material implications; default links are mapped to open normal defaults. Exception links have the effect of providing exceptions to a default link. Thus a default link from α to β, and with an exception link from γ to this link maps onto the rule $\frac{\alpha : \beta \wedge \neg \gamma}{\beta}$, asserting that γ blocks the default inference from α to β. Consequently this approach provides a semantics for these inheritance networks. In so doing it enables the development of provably correct inference procedures, as well as proofs of the existence of extensions in certain cases.

Default logic has also found application in a wide range of diverse fields. It has been applied in natural language understanding, in speech act theory (Perrault, 1987) and for deriving natural language presuppositions (Mercer, 1988), and in database systems (Cadoli *et al.*, 1994). It has been used to extend the expressive power of terminological logics to handle default information (Baader and Hollunder, 1992; Baader and Hollunder, 1993). In (Turner, 1997), in an approach similar to that given here, a high-level language for describing the effects of actions is defined, following which a provably-correct translation into Default Logic is provided.

In (Gelfond and Lifschitz, 1990), *extended logic programs* under the *answer set semantics*, are defined; it is also shown there that extended logic programs correspond to the fragment of Default Logic in which the justifications and consequents of default rules are literals and the prerequisites are conjunctions of literals. We don't discuss logic programming here (but see (Lifschitz, 1996) for an introduction and survey); however this correspondence has a number of important results for our purposes. First, the mapping of a logic program into a Default Logic theory, essentially provides an alternative semantics for this approach to logic programming. Secondly, one can regard results involving extended logic programs as applying to Default Logic. Thus, (Wang *et al.*, 1999) presents a compilation of defeasible inheritance networks into (a subset of) extended logic programs and so into a fragment of Default Logic. On the other hand, (Gelfond and Son, 1998) define a language for strict and defeasible rules that (among other things) can encode inheritance networks; they show how this language can be "implemented" using extended logic programs. Lastly, this correspondence suggests an even stronger thesis than that presented here, that extended logic programs (or the corresponding fragment of Default Logic) provides an appropriate language for encoding commonsense notions. We return to this point in the concluding section.

Finally, it should be noted that there are now comprehensive implementations of Default Logic available. DeReS (Cholewiński *et al.*, 1999) is a full implementation of Default Logic. XRay (Schaub and Nicolas, 1997) provides an implementation platform for query-answering in semi-monotonic default logics. Smodels (Niemelä and Simons, 1997) implements the stable model semantics and well-founded semantics, while DLV (Eiter *et al.*, 1997) is able to handle disjunctive logic programming with the answer set semantics.

3. A Methodology for the Use of Default Logic

In this section we outline the general approach. Our methodology involves the appropriate "deconstruction" of default rules, so that one can detect the application of default rules and control their application. Fundamentally, we show how one can detect whether a default rule is applicable or not, and how one can control the invocation of a default rule *within* a (first-order) default theory. This is accomplished, first, by associating a unique name with each default rule, so that it can be referred to within a theory. Second, special-purpose predicates are introduced for detecting conditions in a default rule, and for controlling rule invocation.

This in turn allows a fine-grained control over what default rules are applied and in what cases. By means of these named rules and special-purpose predicates, one can formalise various phenomena of interest. This is done as follows. We begin with a theory $\mathcal{D}$ expressed in some language. In the examples presented here $\mathcal{D}$ is a (regular) default theory but with an external order on default rules. In the first example, the static order represents a notion of preference among the defaults and, in the second, it represents a simplified version of default property inheritance. One then provides a translation of the given theory into a standard theory in Default Logic, $\mathcal{T}(\mathcal{D})$. Assuming that things have been set up correctly, one can then show that the translated "standard" default theory $\mathcal{T}(\mathcal{D})$ provably captures the phenomenon of interest expressed in $\mathcal{D}$.

3.1. The Approach

We begin by extending the original language by a set of constants (as is done in (Brewka, 1994b) for example), such that there is a bijective mapping between these constants and the default rules.[1] Assume then that we have a default theory (D, W) where each default rule $\frac{\alpha : \beta}{\gamma}$ has an associated name n. This can be written $n : \frac{\alpha : \beta}{\gamma}$; the next section describes naming in more detail. Consider though where we augment our original language with three predicate symbols $\mathsf{ap}(\cdot)$, $\mathsf{bl_P}(\cdot)$, and $\mathsf{bl_J}(\cdot)$. Moreover each default rule $n : \frac{\alpha : \beta}{\gamma}$ is replaced by three rules:

$$\frac{\alpha \,:\, \beta}{\gamma \wedge \mathsf{ap}(n)}, \quad \frac{:\, \neg\alpha}{\mathsf{bl_P}(n)}, \quad \frac{\neg\beta \,:}{\mathsf{bl_J}(n)} \tag{1}$$

Call the resulting default theory (D', W). Any extension E of (D, W) will have a corresponding extension E' of (D', W), where $E \subseteq E'$ and E' informally consists of E along with ground instances of the introduced predicates.

Moreover, we will have $\mathsf{ap}(n) \in E'$ just if the default named n is one of the *generating defaults* of the extension. We will have $\mathsf{bl_P}(n) \in E'$ if the default named by n failed to be applied by virtue of its prerequisite being unproven, and we will have $\mathsf{bl_J}(n) \in E'$ if the default named by n failed to be applied by virtue of its justification being inconsistent with E'. We can go further (see the following section) and prove that for every name n

$$\mathsf{ap}(n) \in E' \quad \text{iff} \quad (\mathsf{bl_P}(n) \notin E' \text{ and } \mathsf{bl_J}(n) \notin E').$$

This is all relatively straightforward and of limited use. However, it can be seen from (1) that via these introduced predicates we can detect when a rule is or is not applied.

[1] In the case of open defaults, we associate names with individual rule instances.

Consider next another introduced predicate $\mathsf{ok}(\cdot)$. If we replace a rule $n : \frac{\alpha : \beta}{\gamma}$ by

$$\frac{\alpha \wedge \mathsf{ok}(n) \ : \ \beta}{\gamma}$$

then clearly the transformed rule can (potentially) be applied only if $\mathsf{ok}(n)$ is asserted. More generally we can combine this with the preceding mapping and so have $\mathsf{ok}(n)$ appear in the prerequisite of each rule in (1). We use a similar translation in the next section where, in axiomatising preferences among default rules, we employ ok to "force" a given order on default rules. That is, we ensure that, with respect to the set of generating defaults, the most preferred rules are first flagged as "ok", then the next most preferred, and so on.

Similarly, one could imagine replacing a default $n : \frac{\alpha : \beta}{\gamma}$ by

$$\frac{\alpha \ : \ \beta \wedge \neg \mathsf{ko}(n)}{\gamma}.$$

Now the transformed rule behaves exactly as the original, except that we can "knock out" this rule by asserting $\mathsf{ko}(n)$. This means of blocking a rule's application has of course appeared earlier in the literature, where ko was most commonly called Ab (for "abnormal").

There are obviously many other possibilities, and we don't mean to suggest that the above constitutes a complete survey. In (Delgrande and Schaub, 1997c) for example we map a rule $n : \frac{\alpha : \beta}{\gamma}$ onto (among other rules) $\frac{\alpha \wedge \mathsf{ok}(n) : \beta}{\mathsf{ap}(n)}$. Notably the consequent of the original rule, γ, is gone, replaced by $\mathsf{ap}(n)$, recording just the fact that the rule is applicable. We use this in (Delgrande and Schaub, 1997c) to axiomatise preferences over *sets* of default rules, where the idea is to apply rules en masse. For a set of rules, if all are found to be applicable, then and only then are all the consequents asserted.

This use of introduced predicates is central to our approach, and provides a means for modelling a very broad class of domains and applications. In the following sections we provide two examples, or "modelling exercises", that illustrate this approach. In each case, we fix the meaning of the phenomena in question by providing appropriate translations (or compilations) that can be performed automatically by appeal to this "tagging" technique. That is, we formalise preference and inheritance by essentially providing an axiomatisation in (standard, Reiter) Default Logic. Being a formalisation in Default Logic, we can *prove* that things work out as expected, that for example, preference among default rules works out correctly, that rules are applied in the specified order, and so on.

4. Preference

The notion of *preference* is pervasive in AI. In reasoning about default properties, one wants to apply defaults pertaining to a more specific class. In decision making, one may have various desiderata, not all of which can be simultaneously satisfied; in such a situation, preferences among desiderata may allow one to come to an appropriate compromise solution. In legal reasoning, laws may conflict. Conflicts may be resolved by principles such as ruling that newer laws will have priority over less recent ones, and laws of a higher authority have priority over laws of a lower authority. For a conflict among these principles one may further decide that the "authority" preference takes priority over the "recency" preference. (Boutilier, 1992; Brewka, 1994a; Baader and Hollunder, 1993) consider adding preferences in Default Logic while (McCarthy, 1986; Lifschitz, 1985; Grosof, 1991) and (Brewka, 1996; Zhang and Foo, 1997; Brewka and Eiter, 1999) do the same in circumscription and logic programming, respectively. In (Delgrande and Schaub, 1997a) we address preference in the context of Default Logic.

For adding preferences between default rules, a default theory is usually extended with an ordering on the set of default rules. In analogy to (Baader and Hollunder, 1993; Brewka, 1994a), an *ordered default theory* $(D, W, <)$ is a finite set D of default rules, a finite set W of formulas, and a strict partial order $< \subseteq D \times D$ on the default rules. That is, $<$ is a binary irreflexive and transitive relation on D. For simplicity, in the following development we assume the existence of a default $\delta_\top = \frac{\top : \top}{\top} \in D$ where for every rule $\delta \in D$, we have $\delta < \delta_\top$ if $\delta \neq \delta_\top$. This gives us a (trivial) maximally preferred default that is always applicable.

Consider an example, where in the north of Québec the first language is French, then English, then Cree. A useful preference ordering is as follows.

$$\frac{NQue : Cree}{Cree} < \frac{Can : English}{English} < \frac{Que : French}{French}. \tag{2}$$

In this case we obtain the correct result for the north of Québec; also we obtain the correct result for the non-north of Québec (where the first language is French followed by English), and for the rest of Canada, where the first language is English. For a second example, one might prefer something (say, a car) that is red, then green; this might be expressed as[2]

$$\frac{: Green}{Green} < \frac{: Red}{Red}. \tag{3}$$

[2] To be sure, this is a naïve encoding; see (Brewka and Gordon, 1994) for a more realistic formalisation.

In (Delgrande and Schaub, 1997a) we show how an ordered default theory $(D, W, <)$ can be translated into a regular default theory (D', W') using the methodology outlined in Section 3 such that the explicit preferences in $<$ are "compiled" into D' and W'. In the next subsection we briefly review this approach.

4.1. Static Preferences on Defaults

We begin with an ordered default theory $(D, W, <)$. The relation $\delta_1 < \delta_2$ has the informal interpretation that for $\delta_1, \delta_2 \in D$, δ_2 is to be considered for application before δ_1. This theory is translated into a regular default theory (D', W') such that the explicit preferences in $<$ are "compiled" into D' and W', in the following manner.

First, a unique name is associated with each default rule. This is done by extending the original language by a set of constants[3] N such that there is a bijective mapping $n : D \to N$. We write n_δ instead of $n(\delta)$ (and abbreviate n_{δ_i} by n_i to ease notation). Also, for default rule δ with name n, we sometimes write $n : \delta$ to render naming explicit. To encode the fact that we deal with a finite set of distinct default rules, we adopt a unique names assumption (UNA_N) and domain closure assumption (DCA_N) with respect to N. That is, for a name set $N = \{n_1, \ldots, n_m\}$, we add axioms

UNA_N: $\quad (n_i \neq n_j)$ for all $n_i, n_j \in N$ with $i \neq j$.

DCA_N: $\quad \forall x.\ name(x) \equiv (x = n_1 \vee \cdots \vee x = n_m)$.

For convenience, we write $\forall x \in N.\ P(x)$ instead of $\forall x.\ name(x) \supset P(x)$.

Given $\delta_i < \delta_j$, we want to ensure that, before δ_i is applied, δ_j can be applied or found to be inapplicable. More formally, we wish to exclude the case where $\delta_i \in D_n$ and $\delta_j \in D_m$ for $n \leq m$ in Definition 2.1. For this purpose, we need to be able to, first, detect when a rule has been applied or when a rule is blocked, and, second, control the application of a rule based on other antecedent conditions. For a default rule $\frac{\alpha : \beta}{\gamma}$ there are two cases for it to not be applied: it may be that the antecedent is not known to be true (and so its negation is consistent), or it may be that the justification is not consistent (and so its negation is known to be true). For detecting this case, we introduce a new, special-purpose predicate $\mathsf{bl}(\cdot)$. Similarly we introduce a predicate $\mathsf{ap}(\cdot)$ to detect when a rule has been applied. To control application of a rule we introduce

[3] (McCarthy, 1986) effectively first suggested the naming of defaults using a set of *aspect* functions. Theorist (Poole, 1988) uses atomic propositions to name defaults.

predicate $\mathsf{ok}(\cdot)$. Then, a default rule $\delta = \frac{\alpha:\beta}{\gamma}$ is mapped to

$$\frac{\alpha \wedge \mathsf{ok}(n_\delta) \ : \ \beta}{\gamma \wedge \mathsf{ap}(n_\delta)}, \quad \frac{\mathsf{ok}(n_\delta) \ : \ \neg\alpha}{\mathsf{bl}(n_\delta)}, \quad \frac{\neg\beta \wedge \mathsf{ok}(n_\delta) \ :}{\mathsf{bl}(n_\delta)} \ . \tag{4}$$

These rules are sometimes abbreviated by $\delta_a, \delta_{b_1}, \delta_{b_2}$, respectively. While δ_a is more or less the image of the original rule δ, rules δ_{b_1} and δ_{b_2} capture the non-applicability of the rule.

None of the three rules in the translation can be applied unless $\mathsf{ok}(n_\delta)$ is true. Since $\mathsf{ok}(\cdot)$ is a new predicate symbol, it can be expressly made true in order to potentially enable the application of the three rules in the image of the translation. If $\mathsf{ok}(n_\delta)$ is true, the first rule of the translation may potentially be applied. If a rule has been applied, then this is indicated by assertion $\mathsf{ap}(n_\delta)$. The last two rules give conditions under which the original rule is inapplicable: either the negation of the original antecedent α is consistent (with the extension) or the justification β is known to be false; in either such case $\mathsf{bl}(n_\delta)$ is concluded.

We can assert that default $n_j : \frac{\alpha_j : \beta_j}{\gamma_j}$ is preferred to $n_i : \frac{\alpha_i : \beta_i}{\gamma_i}$ in the object language by introducing a new predicate $\prec$ and then asserting that $n_i \prec n_j$. However, this translation so far does nothing to control the order of rule application. Nonetheless, for $\delta_i < \delta_j$ we can now control the order of rule application: we can assert that if δ_j has been applied (and so $\mathsf{ap}(n_j)$ is true), or known to be inapplicable (and so $\mathsf{bl}(n_j)$ is true), then it's ok to apply δ_i. The idea is thus to *delay* the consideration of less preferred rules until the applicability question has been settled for the higher ranked rules.

We obtain the following translation, mapping ordered default theories in some language $\mathcal{L}$ onto standard default theories in the language $\mathcal{L}^+$ obtained by extending $\mathcal{L}$ by new predicates symbols $(\cdot \prec \cdot)$, $\mathsf{ok}(\cdot)$, $\mathsf{bl}(\cdot)$, and $\mathsf{ap}(\cdot)$, and a set of associated default names:

DEFINITION 4.1. (Delgrande and Schaub, 1997a) *Given an ordered default theory $(D, W, <)$ over $\mathcal{L}$ and its set of default names $N = \{n_\delta \mid \delta \in D\}$, define $\mathcal{T}((D, W, <)) = (D', W')$ over $\mathcal{L}^+$ by*

$$\begin{aligned} D' &= \left\{ \tfrac{\alpha \wedge \mathsf{ok}(n) : \beta}{\gamma \wedge \mathsf{ap}(n)}, \ \tfrac{\mathsf{ok}(n) : \neg\alpha}{\mathsf{bl}(n)}, \ \tfrac{\neg\beta \wedge \mathsf{ok}(n) :}{\mathsf{bl}(n)} \ \middle| \ n : \tfrac{\alpha : \beta}{\gamma} \in D \right\} \cup D_\prec \\ W' &= W \cup W_\prec \cup \{DCA_N, UNA_N\} \end{aligned}$$

where

$$\begin{aligned} D_{\prec} &= \left\{ \tfrac{:\neg(x \prec y)}{\neg(x \prec y)} \right\} \\ W_{\prec} &= \{ n_{\delta} \prec n_{\delta'} \mid (\delta, \delta') \in < \} \\ &\cup \{ \mathsf{ok}(n_{\top}) \} \\ &\cup \{ \forall x \in N. \left[\forall y \in N. \, (x \prec y) \supset (\mathsf{bl}(y) \vee \mathsf{ap}(y)) \right] \supset \mathsf{ok}(x) \}. \end{aligned}$$

W' contains the prior world knowledge W, together with assertions for managing the priority order $<$ on defaults. The first part of $W_{\prec}$ specifies that $\prec$ is a predicate whose positive instances mirror those of the strict partial order $<$. $\mathsf{ok}(n_{\top})$ asserts that it is ok to apply the maximally preferred (trivial) default. The third formula in $W_{\prec}$ controls the application of defaults: for every n_i, we derive $\mathsf{ok}(n_i)$ whenever for every n_j with $n_i \prec n_j$, either $\mathsf{ap}(n_j)$ or $\mathsf{bl}(n_j)$ is true. This axiom allows us to derive $\mathsf{ok}(n_i)$, indicating that δ_i may potentially be applied whenever we have for all δ_j with $\delta_i < \delta_j$ that δ_j has been applied or cannot be applied.

For the last formula in $W_{\prec}$ to work properly we must have complete information about $\prec$. This is addressed by the default rule in $D_{\prec}$. With this rule we can generate the complement of $W_{\prec}$ with a single open default; an explicit encoding would likely have much larger size. We also have $(n_{\delta} \prec n_{\top}) \in W_{\prec}$ for every rule $\delta \neq \delta_{\top}$ by the definition of ordered default theories. Since $<$ is a strict partial order, $W_{\prec}$ also includes the transitive closure of $\prec$ and no reflexivities, such as $n \prec n$.

Note that the translation results in a manageable increase in the size of the default theory. For ordered theory $(D, W, <)$, the translation $\mathcal{T}((D, W, <))$ is only a constant factor larger than $(D, W, <)$ (assuming that we count the default in $D_{\prec}$ as a single default).

As an example, consider the defaults:

$$n_1 : \tfrac{A_1 : B_1}{C_1}, \quad n_2 : \tfrac{A_2 : B_2}{C_2}, \quad n_3 : \tfrac{A_3 : B_3}{C_3}, \quad n_{\top} : \tfrac{\top : \top}{\top}.$$

We obtain for $i = 1, 2, 3$:

$$\tfrac{A_i \wedge \mathsf{ok}(n_i) : B_i}{C_i \wedge \mathsf{ap}(n_i)}, \qquad \tfrac{\mathsf{ok}(n_i) : \neg A_i}{\mathsf{bl}(n_i)}, \qquad \tfrac{\neg B_i \wedge \mathsf{ok}(n_i) :}{\mathsf{bl}(n_i)},$$

and analogously for $\delta_{\top}$ where A_i, B_i, C_i are $\top$. Given $\delta_1 < \delta_2 < \delta_3$, we obtain $n_1 \prec n_2$, $n_2 \prec n_3$, $n_1 \prec n_3$ along with $n_k \prec n_{\top}$ for $k \in \{1, 2, 3\}$ as part of $W_{\prec}$. From $D_{\prec}$ we get $\neg(n_i \prec n_j)$ for all remaining combinations of $i, j \in \{1, 2, 3, \top\}$. It is instructive to verify that $\mathsf{ok}(n_3)$, along with

$$\begin{aligned} (\mathsf{ap}(n_3) \vee \mathsf{bl}(n_3)) &\supset \mathsf{ok}(n_2), \qquad \text{and} \\ ((\mathsf{ap}(n_2) \vee \mathsf{bl}(n_2)) \wedge (\mathsf{ap}(n_3) \vee \mathsf{bl}(n_3))) &\supset \mathsf{ok}(n_1) \end{aligned}$$

are obtained after a few iterations in Definition 2.1 (see below); from this we get that n_3 must be taken into account first, followed by n_2 and then n_1.

The following theorem summarises the major technical properties of our approach, and demonstrates that rules are applied in the desired order:

THEOREM 4.1. (Delgrande and Schaub, 1997a) *Let E be a consistent extension of $\mathcal{T}((D,W,<))$ for ordered default theory $(D,W,<)$. We have for all $\delta, \delta' \in D$ that*

1. $n_\delta \prec n_{\delta'} \in E$ *iff* $\neg(n_\delta \prec n_{\delta'}) \notin E$
2. $\mathsf{ok}(n_\delta) \in E$
3. $\mathsf{ap}(n_\delta) \in E$ *iff* $\mathsf{bl}(n_\delta) \notin E$
4. $\mathsf{ok}(n_\delta) \in E_i$ *and* $\mathit{Prereq}(\delta) \in E_j$ *and* $\neg \mathit{Justif}(\delta) \notin E$ *implies* $\mathsf{ap}(n_\delta) \in E_{\max(i,j)+3}$
5. $\mathsf{ok}(n_\delta) \in E_i$ *and* $\mathit{Prereq}(\delta) \notin E$ *implies* $\mathsf{bl}(n_\delta) \in E_{i+1}$
6. $\mathsf{ok}(n_\delta) \in E_i$ *and* $\neg \mathit{Justif}(\delta) \in E$ *implies* $\mathsf{bl}(n_\delta) \in E_j$ *for some* $j > i+1$
7. $\mathsf{ok}(n_\delta) \notin E_{i-1}$ *and* $\mathsf{ok}(n_\delta) \in E_i$ *implies* $\mathsf{ap}(n_\delta) \notin E_j$ *for* $j < i+2$ *and* $\mathsf{bl}(n_\delta) \notin E_j$ *for* $j < i+1$

Moreover, it turns out that our translation $\mathcal{T}$ amounts to selecting those extensions of the original default theory that are in accord with the provided ordering. This can be expressed in the following way.

DEFINITION 4.2. (Delgrande and Schaub, 2000) *Let (D,W) be a default theory and let $< \subseteq D \times D$ be a strict partial order. An extension E of (D,W) is $<$-preserving if there exists a grounded enumeration $\langle \delta_i \rangle_{i \in I}$ of $GD(D,E)$ such that for all $i, j \in I$ and $\delta \in D \setminus GD(D,E)$, we have that*

1. *if* $\delta_i < \delta_j$ *then* $j < i$ *and*
2. *if* $\delta_i < \delta$ *then* $\mathit{Prereq}(\delta) \notin E$ *or* $W \cup \mathit{Conseq}(\{\delta_0, \ldots, \delta_{i-1}\}) \vdash \neg \mathit{Justif}(\delta)$.

In the first condition above, applied rules are applied in the order specified by $<$. Second, if a rule δ is not applied but a less-highly ranked rule is, then it must be the case that either the prerequisite of δ is not derivable (at all) or its justification is refuted by other, higher-ranked rules. In any case, the applicability issue must first be settled for higher-ranked default rules before it is for lower-ranked rules.

THEOREM 4.2. (Delgrande and Schaub, 2000) *Let (D,W) be a default theory and let $< \subseteq D \times D$ be a strict partial order. Let E be a set of formulas.*

E is an extension of $\mathcal{T}((D,W,<))$ iff $E \cap \mathcal{L}$ is a $<$-preserving extension of (D,W).

Consequently, the notion of $<$-preservation can be seen as providing an informal semantics for our approach.

One might expect that ordered default theories would enjoy the same properties as standard Default Logic. This indeed is the case, but with one important exception: in the instance of our approach described here, normal ordered default theories do not guarantee the existence of extensions. For example, the image of the ordered default theory (under our translation)

$$\left(\{n_1 : \tfrac{:B}{B},\ n_2 : \tfrac{B:C}{C}\}, \emptyset, \{\delta_1 < \delta_2\}\right) \tag{5}$$

has no extension. Informally the problem is that our preference $\delta_1 < \delta_2$ conflicts with the normal order of rule application. If $W = \emptyset$, only δ_1 is applicable, but once it has applied, δ_2 becomes applicable. Thus we have an ordering implicit in the form of the defaults and world knowledge, but where this implicit ordering is contradicted by the assertion $\delta_1 < \delta_2$. Not surprisingly then there is no extension.

4.2. Extensions

In (Delgrande and Schaub, 1997a) we also show that standard default theories (D,W) over a language including a predicate expressing a preference over (named) default rules can similarly be translated into a default theory where no such mention of preferences is made. Thus for example, let *bf* be the name of default $\frac{B:F}{F}$, asserting that birds fly by default and *bnf* be the name of default $\frac{B:\neg F}{\neg F}$. Further, let $loc(NZ)$ assert that the location is New Zealand. Then if we believed that birds normally fly, but New Zealand birds don't, we could encode this in the object theory by

$$\frac{:\neg loc(NZ)}{bf \prec bnf}, \quad \frac{loc(NZ):}{bnf \prec bf}.$$

This extension allows the expression of preference in a particular context (as above), preferences applying by default, and preferences among preferences (as in the legal reasoning example mentioned at the start of this section).

Second, we show how a default theory, with an attendant ordering on *sets* of defaults, can similarly be translated into a standard default theory where no mention of preferences is made. In this case, we can express that in buying a car one may rank price (E) of a car model over safety features (S) over power (P), but safety features together with power is ranked over price, as follows:

$$\left\{\frac{:P}{P}\right\} < \left\{\frac{:S}{S}\right\} < \left\{\frac{:E}{E}\right\} < \left\{\frac{:P}{P}, \frac{:S}{S}\right\} \qquad (6)$$

If we were given only that not all desiderata can be satisfied (and so W contains $\neg P \vee \neg S \vee \neg E$) then intuitively we would want to apply the defaults in the set $\left\{\frac{:P}{P}, \frac{:S}{S}\right\}$ and conclude that P and S can be met. On the other hand if we know that P and S are mutually exclusive (and so W contains $\neg P \vee \neg S$) then intuitively we would want to apply the defaults in the sets $\left\{\frac{:S}{S}\right\}$ and $\left\{\frac{:E}{E}\right\}$ and so conclude that S and E can be met. Again we show how this can be expressed in a standard default theory.

4.3. Application to model-based diagnosis

In (Reiter, 1987), Default Logic is used to provide an account of a theory of diagnosis from first principles. Roughly, in this framework one has an axiomatisation of a domain, or *system description*, in which there is a distinguished set of *components*, given by a set of constants *COMPS* that may be normal or abnormal. These components are assumed to be normal by default, expressed by the rule $\frac{:\neg Ab(x)}{\neg Ab(x)}$. For example, in the circuit domain the system description would include a description of a circuit, while the set *COMPS* could represent specific gates. We can express that an AND gate that is not abnormal has output *on* when its inputs are *on* by:

$$\begin{aligned}(AndG(x) \wedge \neg Ab(x) \wedge value(in(1,x), on) \wedge value(in(2,x), on))\\ \supset value(out(x), on).\end{aligned}$$

As well there is a set of *observations OBS*, for example expressing that both inputs of AND gate a_1 are *on* but the output is not *on*. A diagnosis can be expressed by an extension of the resulting theory. In an extension, one has complete information concerning all instances of Ab; moreover the extension of Ab is minimal.

We can use our approach to incorporate further assumptions into a theory of diagnosis. The original approach appeals to a *principle of parsimony*, wherein a diagnosis is a conjecture that some minimal set of components are faulty. This can be strengthened by preferring a single-fault diagnosis over two-fault diagnosis, over three-fault diagnosis, etc. Suppose we have three components whose normal behaviour is modelled[4] by the rule $\frac{:\neg Ab(x)}{\neg Ab(x)}$. In our extension to preference that allows

[4] Note that we haven't addressed the problem of preferences on open defaults. We skirt any difficulties here by assuming complete knowledge of the (circuit) domain.

preferences among sets of defaults, we can model the strengthened principle of parsimony by

$$\left\{\frac{:\neg Ab(c_1)}{\neg Ab(c_1)}\right\} < \left\{\frac{:\neg Ab(c_1)}{\neg Ab(c_1)}, \frac{:\neg Ab(c_2)}{\neg Ab(c_2)}\right\} < \left\{\frac{:\neg Ab(c_1)}{\neg Ab(c_1)}, \frac{:\neg Ab(c_2)}{\neg Ab(c_2)}, \frac{:\neg Ab(c_3)}{\neg Ab(c_3)}\right\}$$

for every c_1, c_2, $c_3 \in \mathit{COMPS}$. Suppose our system description entails $Ab(a) \vee (Ab(b) \wedge Ab(c))$. In standard Default Logic, we obtain two extensions, which violates the strengthened principle of parsimony. With the given preferences, however, we obtain only the single-fault extension, containing $Ab(a)$ along with $\neg Ab(b)$ and $\neg Ab(c)$.

In a second extension, we can model preferences for faults over types of components. In the extension to our approach where preferences can be expressed in the language, we can express the fact that an OR gate is expected to fail over an AND gate as follows. For $c \in \mathit{COMPS}$ let the name of the rule $\frac{:\neg Ab(c)}{\neg Ab(c)}$ be given by predicate $AbRule(c)$. We assert:

$$\forall x, y.(OrG(x) \wedge AndG(y)) \supset (AbRule(y) \prec AbRule(x)).$$

Clearly other elaborations can be addressed within this framework. For example, it would be an elementary extension to allow different types of faults, and then assert, say, that an AND gate that is stuck on is to be expected over an OR gate that negates its correct output.

5. Inheritance of Properties

5.1. Preference and Inheritance of Properties

A common problem in Knowledge Representation is the inheritance of (default) properties. Informally, individuals may be expected to have (or inherit) properties by virtue of being instances of particular classes. Thus by default Sue will be assumed to be employed since Sue is an adult and adults are normally employed. The principle of *specificity* says (for our purposes) that properties are inherited from more specific classes in preference to less specific classes. Thus if Sue is a student also, and we know that students normally are not employed, then we now can conclude nothing about Sue's employment status. If we have that adult students are normally employed, then this would be applied in preference to the students-are-not-employed default.

It might seem that we could use the approach of the previous section to implement inheritance of properties, and indeed many approaches implement inheritance via a preference ordering. However an example shows that this can lead to unfortunate results. Consider defaults concerning primary means of locomotion: "animals normally walk", "birds normally fly", "penguins normally swim". This can be expressed in an ordered default theory as follows:

$$n_1 : \frac{Animal : Walk}{Walk} \quad < \quad n_2 : \frac{Bird : Fly}{Fly} \quad < \quad n_3 : \frac{Penguin : Swim}{Swim}. \tag{7}$$

If we learn that some thing is penguin (and so a bird and animal), then we would want to apply the highest-ranked default and, all other things being equal, conclude that it swims. However, if the penguin in question is hydrophobic, and so doesn't swim, preference tells us that we should try to apply the next default and so, again all other things being equal, conclude that it flies. This situation then is very different from our example (2), and moreover in this instance gives us an undesirable conclusion.

We can characterise this difference as follows. Let $(D, W, <)$ be a normal (i.e. the defaults in D are normal) ordered default theory. For preference, as described in the previous section, we want to "apply" defaults as constrained by $<$. For inheritance, we want to apply the $<$-maximum defaults where the prerequisite is true, if possible. If a $<$-maximum default is inapplicable, then no less specific default is considered. In the next section we make these intuitions precise and provide an axiomatisation in Default Logic.

5.2. Expressing Inheritance

For default property inheritance, the ordering on defaults reflects a relation of *specificity* among the prerequisites. For example, in the preceding, the class of penguins is strictly narrower than the class of birds. As (2) and (3) illustrate, this isn't the case for preference. Informally, for adjudicating among conflicting defaults, one determines the most specific (with respect to rule prerequisite) defaults as candidates for application. In approaches such as (Pearl, 1990; Geffner and Pearl, 1992), among many others, specificity is determined implicitly, emerging as a property of an underlying formal system. (Reiter and Criscuolo, 1981; Etherington and Reiter, 1983; Delgrande and Schaub, 1997b) have addressed encoding specificity information in Default Logic. Here we briefly provide an account of how a mechanism of inheritance may be encoded.

To begin with, for simplicity, our account is incomplete. In particular we ignore the fact that, in formulating an ordered default theory, the specification of $<$ must take into account the relevant properties (i.e. consequents) of the default rules. For example, given defaults concerning flight, and a rule that birds normally have wings (viz. $\frac{Bird : Fly}{Fly}$, $\frac{Bird : Wing}{Wing}$, $\frac{Penguin : \neg Fly}{\neg Fly}$) we would assert $\frac{Bird : Fly}{Fly} < \frac{Penguin : \neg Fly}{\neg Fly}$.

Obviously we would not want to assert $\frac{Bird:Wing}{Wing} < \frac{Penguin:\neg Fly}{\neg Fly}$ even though $Bird$ subsumes $Penguin$. So here we have nothing to say about *how* the information concerning $<$ is obtained.[5] Rather, we assume that inheritance information *has been* appropriately captured in $<$, and our task is to provide a semantic account of property inheritance via an appropriate translation into Default Logic.

In the last section, our approach to preference used special purpose predicates to detect when a rule in D was applied or blocked. A more fine-grained approach would be to distinguish the source of blockage by replacing δ_{b_1} and δ_{b_2} by

$$\frac{\mathsf{ok}(n_\delta)\ :\ \neg\alpha}{\mathsf{bl}_{\mathsf{P}}(n_\delta)}, \quad \frac{\neg\beta \wedge \mathsf{ok}(n_\delta)\ :}{\mathsf{bl}_{\mathsf{J}}(n_\delta)}\ .$$

That is, we replace bl by two new predicate symbols, bl_{P} and bl_{J}. Accordingly, in Definition 4.1 the final formula in $W_\prec$ would be

$$\forall x \in N.\ \left[\forall y \in N.\ (x \prec y) \supset (\mathsf{bl}_{\mathsf{P}}(y) \vee \mathsf{bl}_{\mathsf{J}}(y) \vee \mathsf{ap}(y))\right] \supset \mathsf{ok}(x)\ .$$

Interestingly, a generalisation of this axiom, namely

$$\forall x \in N.\ \left[\forall y \in N.\ ((x \prec y) \wedge (y \neq n_\top)) \supset \mathsf{bl}_{\mathsf{P}}(y)\right] \supset \mathsf{ok}(x)\ , \qquad (8)$$

allows us to specify *inheritance*. Given a chain of defaults $\delta_1 < \delta_2 < \cdots < \delta_m$, we apply δ_i, if possible, where δ_i is the $<$-maximum default such that for every default δ_j, $j = i+1, \ldots, m$, the prerequisite of δ_j is not known to be true. Otherwise no default in the chain is applicable. Technically, the formula (8) allows lower ranked default rules to be applied only in case higher ranked rules are blocked because their prerequisite is not derivable. Otherwise, the propagation of $\mathsf{ok}(\cdot)$-predicates is interrupted so that no defaults below δ_i are considered.

DEFINITION 5.1. *Given an ordered default theory $(D, W, <)$ over $\mathcal{L}$ and its set of default names $N = \{n_\delta \mid \delta \in D\}$, define $\mathcal{I}((D, W, <)) = (D', W')$ over $\mathcal{L}^+$ by*

$$\begin{aligned} D' &= \left\{ \tfrac{\alpha \wedge \mathsf{ok}(n)\,:\,\beta}{\gamma},\ \tfrac{\mathsf{ok}(n)\,:\,\neg\alpha}{\mathsf{bl}_{\mathsf{P}}(n)} \ \middle|\ n : \tfrac{\alpha:\beta}{\gamma} \in D \right\} \cup D_\prec \\ W' &= W \cup W_\prec \cup \{DCA_N, UNA_N\} \end{aligned}$$

where

$$\begin{aligned} D_\prec &= \left\{ \tfrac{:\neg(x \prec y)}{\neg(x \prec y)} \right\} \\ W_\prec &= \{n_\delta \prec n_{\delta'} \mid (\delta, \delta') \in <\} \\ &\cup\ \{\mathsf{ok}(n_\top)\} \\ &\cup\ \{\forall x \in N.\ \left[\forall y \in N.\ ((x \prec y) \wedge (y \neq n_\top)) \supset \mathsf{bl}_{\mathsf{P}}(y)\right] \supset \mathsf{ok}(x)\}. \end{aligned}$$

[5] This issue is dealt with in (Delgrande and Schaub, 1997b).

Consider the ordered defaults theory $(D, W, <)$ where D and $<$ are as in (7), and where

$$W = \{Penguin \supset Bird, Bird \supset Animal\}.$$

We obtain $n_1 \prec n_2$, $n_2 \prec n_3$, $n_1 \prec n_3$ along with $n_k \prec n_\top$ for $k \in \{1, 2, 3\}$ as part of $W_\prec$. From $D_\prec$ we get $\neg(n_i \prec n_j)$ for all remaining combinations of $i, j \in \{1, 2, 3, \top\}$.

For the theory $(D, W \cup \{Bird\}, <)$ it is useful to verify that one extension is obtained. Initially $\mathsf{ok}(n_3)$ is obtained (in Definition 2.1) as is $\mathsf{bl_P}(n_3)$. Since we can deduce $\mathsf{bl_P}(n_3) \supset \mathsf{ok}(n_2)$, the rule $\frac{Bird \wedge \mathsf{ok}(n_2) : Fly}{Fly}$ can be applied. Since the rule $\frac{\mathsf{ok}(n_2) : \neg Bird}{\mathsf{bl_P}(n_2)}$ clearly cannot be applied, we do not obtain $\mathsf{bl_P}(n_2)$ and since this is the only way in which $\mathsf{ok}(n_1)$ can be obtained (from the (reduced) formula $(\mathsf{bl_P}(n_2) \wedge \mathsf{bl_P}(n_3)) \supset \mathsf{ok}(n_1)$ in $W_\prec$) we do not obtain $\mathsf{ok}(n_1)$. If instead we have the theory $(D, W \cup \{Penguin, \neg Swim\}, <)$, we again obtain one extension, containing $Animal$, $Bird$, $Penguin$, and $\mathsf{ok}(n_3)$. However we do not obtain $\mathsf{ok}(n_1)$, $\mathsf{ok}(n_2)$, nor $\mathsf{bl_P}(n_1)$, $\mathsf{bl_P}(n_2)$.

The following theorem summarises the major technical properties of our approach, and demonstrates that rules are applied in the desired order:

THEOREM 5.1. *Let E be a consistent extension of $\mathcal{I}((D, W, <))$ for ordered default theory $(D, W, <)$. We have for all $\delta, \delta' \in D$ that*

1. $n_\delta \prec n_{\delta'} \in E$ *iff* $\neg(n_\delta \prec n_{\delta'}) \notin E$
2. $\mathsf{ok}(n_\delta) \in E_i$ *and* $Prereq(\delta) \in E_j$ *and* $\neg Justif(\delta) \notin E$ *implies* $Conseq(\delta) \in E_{\max(i,j)+3}$
3. $\mathsf{ok}(n_\delta) \in E_i$ *and* $Prereq(\delta) \notin E$ *implies* $\mathsf{bl_P}(n_\delta) \in E_{i+1}$
4. $Prereq(\delta) \in E$ *where* $\delta \neq \delta_\top$ *implies for every* δ' *where* $\delta' < \delta$ *we have* $\mathsf{ok}(n_{\delta'}) \notin E$
5. $\mathsf{ok}(n_\delta) \in E$ *iff for every* δ' *where* $\delta < \delta'$ *we have* $\mathsf{ok}(n_{\delta'}) \in E$ *and* $\mathsf{bl_P}(n_\delta) \in E$

It follows immediately from the last two parts above that for $\delta \in D$, where $\delta \neq \delta_\top$, if $\delta_a \in GD(D, E)$ then

- for every δ' where $\delta < \delta'$ we have $\mathsf{bl_P}(n_\delta) \in E$, and
- for every δ' where $\delta' < \delta$ we have $\mathsf{ok}(n_{\delta'}) \notin E$.

That is, if a non-trivial default is applied, then every $<$-greater default has an unprovable prerequisite, and every $<$-lesser default is not considered.

6. Discussion

We have proposed and illustrated a general methodology for using Default Logic as an analytical tool and as an underlying formalism for the representation of knowledge. This role for Default Logic extends that advocated for formal logic in Knowledge Representation. The methodology involves the naming of default rules and the introduction of special-purpose predicates, for detecting conditions for default rule applicability and controlling a rule's application. This allows the encoding of specific strategies and policies governing a set of default rules. Given this, we present two examples, wherein Default Logic is used to formalise preferences among properties and the inheritance of default properties. In earlier work (Delgrande and Schaub, 1999) we have also shown how a notion of "similar individuals" can be encoded so that default rules apply uniformly to such similar individuals.

Thus, in our examples we show how Default Logic can be employed to provide a semantics for such phenomena and, on the other hand, provide an encoding for reasoning with such phenomena. Given that there are now comprehensive implementations of Default Logic available, it also becomes a straightforward matter to implement, for example, preference or property inheritance: one needs just implement the translation into Default Logic, and feed the result into a Default Logic theorem prover.

These examples suggest that the general methodology proposed here provides a general and useful approach to analyse and axiomatise various diverse phenomena. For example, our translations demonstrate that there are distinct notions having to do with preference and priority on the one hand, and property inheritance on the other. Hence we suggest that, in addition to directly representing nonmonotonic theories, Default Logic is appropriate as a general Artificial Intelligence formalism in which specific phenomena may be encoded. In fact, a stronger, perhaps more pragmatic, thesis[6] can be advanced in view of implemented reasoning systems such as Smodels (Niemelä and Simons, 1997) (Eiter *et al.*, 1997) and DLV: that a subset of Default Logic, corresponding to extended logic programs under the answer set semantics, might provide just the appropriate approach for encoding and addressing such phenomena as advocated here.

[6] We thank a reviewer for suggesting this.

References

J. Allen and P.J. Hayes. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5(4):225–238, 1989.

F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning*, pages 306–317, Cambridge, MA, October 1992.

F. Baader and B. Hollunder. How to prefer more specific defaults in terminological default logic. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 669–674, Chambéry, France, 1993.

C. Boutilier. What is a default priority? In J. Glasgow and R. Hadley, editors, *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pages 140–147, Vancouver, B.C., 1992.

G. Brewka and T. Eiter. Preferred answer sets for extended logic programs. *Artificial Intelligence*, 109(1-2):297–356, 1999.

G. Brewka and T. Gordon. How to buy a porsche: An approach to defeasible decision making. In *AAAI-94 Workshop on Computational Dialectics*, pages 28–38, Seattle, WA, July 1994. AAAI Press.

G. Brewka. Adding priorities and specificity to default logic. In L. Pereira and D. Pearce, editors, *European Workshop on Logics in Artificial Intelligence (JELIA'94)*, Lecture Notes in Artificial Intelligence, pages 247–260. Springer Verlag, 1994.

G. Brewka. Reasoning about priorities in default logic. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, volume 2, pages 940–945. AAAI Press/The MIT Press, 1994.

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

M. Cadoli, T. Eiter, and G. Gottlob. Default logic as a query language. In J. Doyle, P. Torasso, and E. Sandewall, editors, *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 99–108. Morgan Kaufmann Publishers, 1994.

P. Cholewiński, V. Marek, A. Mikitiuk, and M. Truszczyński. Computing with default logic. *Artificial Intelligence*, 112(1-2):105–146, 1999.

J.P. Delgrande and T. Schaub. Compiling reasoning with and about preferences into default logic. In T. Dean, editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 168–174, Nagoya, Japan, August 1997. Morgan Kaufmann Publishers.

J.P. Delgrande and T. Schaub. Compiling specificity into approaches to nonmonotonic reasoning. *Artificial Intelligence*, 90(1-2):301–348, 1997.

J.P. Delgrande and T. Schaub. Reasoning with sets of preferences in default logic. In *The Second IJCAI Workshop on Nonmonotonic Reasoning, Action and Change*, Nagoya, Japan, August 1997.

J.P. Delgrande and T. Schaub. The role of default logic in knowledge representation. In *Workshop on Logic-Based Artificial Intelligence*, Washington, D.C., June 1999.

J. Delgrande and T. Schaub. Expressing preferences in default logic. *Artificial Intelligence*, 123(1-2):41–87, 2000.

T. Eiter, N. Leone, C. Mateis, G. Pfeifer, , and F. Scarcello. A deductive system for nonmonotonic reasoning. In J. Dix, U. Furbach, and A. Nerode, editors, *Proceedings of the Fourth International Conference on Logic Program-*

ming and Non-Monotonic Reasoning, volume 1265 of *Lecture Notes in Artificial Intelligence*, pages 363–374. Springer Verlag, 1997.

D.W. Etherington and R. Reiter. On inheritance hierarchies with exceptions. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 104–108. Morgan Kaufmann Publishers, 1983.

H. Geffner and J. Pearl. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence*, 53(2-3):209–244, 1992.

M. Gelfond and V. Lifschitz. Logic programs with classical negation. In David H. D. Warren and Peter Szeredi, editors, *Proceedings of the International Conference on Logic Programming*, pages 579–597. MIT, 1990.

M. Gelfond and T. C. Son. Reasoning with prioritized defaults. In Jürgen Dix, Luís Moniz Pereira, and Teodor C. Przymusinski, editors, *Proceedings of the 3th International Workshop on Logic Programming and Knowledge Representation (LPKR-97)*, volume 1471 of *LNAI*, pages 164–223, Berlin, 1998. Springer.

B. Grosof. Generalizing prioritization. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning*, pages 289–300, San Mateo, CA, April 1991. Morgan Kaufmann.

P.J. Hayes. Naive physics I: Ontology for liquids. In J.R. Hobbs and R.C. Moore, editors, *Formal Theories of the Commonsense World*, pages 71–108. Ablex, 1985.

P.J. Hayes. The second naive physics manifesto. In J.R. Hobbs and R.C. Moore, editors, *Formal Theories of the Commonsense World*, pages 1–36. Ablex, 1985.

T. Janhunen. Classifying semi-normal default logic on the basis of its expressive power. In M. Gelfond, N. Leone, and G. Pfeifer, editors, *Proceedings of the Fifth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'99)*, volume 1730 of *Lecture Notes in Artificial Intelligence*, pages 19–33. Springer Verlag, 1999.

H.J. Levesque, F. Pirri, and R. Reiter. Foundations for the situation calculus. *Linköping Electronic Articles in Computer and Information Science*, 3(18), 1998.

V. Lifschitz. Closed-world databases and circumscription. *Artificial Intelligence*, 27:229–235, 1985.

V. Lifschitz. Foundations of logic programming. In G. Brewka, editor, *Principle of Knowledge Representation*, pages 69–127. CSLI, 1996.

J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In D. Michie and B. Meltzer, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

J. McCarthy. First order theories of individual concepts and propositions. In D. Michie, editor, *Machine Intelligence 9*, pages 129–147. Edinburgh University Press, 1979.

J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

R.E. Mercer. Using default logic to derive natural language suppositions. In R. Goebel, editor, *Proceedings of the Seventh Biennial Canadian Conference on Artificial Intelligence*, pages 14–21, 1988.

R.C. Moore. Reasoning about knowledge and action. Technical Report 181, SRI International, 1980.

R.C. Moore. The role of logic in knowledge representation and commonsense reasoning. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 428–433, Pittsburgh, PA, 1982.

I. Niemelä and P. Simons. Smodels: An implementation of the stable model and well-founded semantics for normal logic programs. In J. Dix, U. Furbach, and

A. Nerode, editors, *Proceedings of the Fourth International Conference on Logic Programing and Nonmonotonic Reasoning*, pages 420–429. Springer-Verlag, 1997.

J. Pearl. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In R. Parikh, editor, *Proc. of the Third Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 121–135, Pacific Grove, Ca., 1990. Morgan Kaufmann Publishers.

C.R. Perrault. An application of default logic to speech act theory. Technical Report CSLI-87-90, Stanford University, 1987.

D.L. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–48, 1988.

R. Reiter and G. Criscuolo. On interacting defaults. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 270–276, Vancouver, B.C., 1981.

R. Reiter and J. de Kleer. Foundations for assumption-based truth maintenance systems: Preliminary report. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 183–188. Morgan Kaufmann Publishers, 1987.

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

R. Reiter. Towards a logical reconstruction of relational database theory. In M.L. Brodie, J. Mylopoulos, and J.W. Schmidt, editors, *On Conceptual Modelling*, pages 191–233. Springer-Verlag, 1984.

R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

T. Schaub and P. Nicolas. An implementation platform for query-answering in default logics: The XRay system, its implementation and evaluation. In J. Dix, U. Furbach, and A. Nerode, editors, *Proceedings of the Fourth International Conference on Logic Programming and Non-Monotonic Reasoning*, volume 1265 of *Lecture Notes in Artificial Intelligence*, pages 442–453. Springer Verlag, 1997.

H. Turner. Representing actions in logic programs and default theories: A situation calculus approach. *Journal of Logic Programming*, 31(1–3):245–298, 1997.

X. Wang, J. You, and L. Yuan. Compiling defeasible inheritance networks to general logic programs. *Artificial Intelligence*, 113(1-2):247–268, 1999.

Y. Zhang and N. Foo. Answer sets for prioritized logic programs. In J. Maluszynski, editor, *Proceedings of the International Symposium on Logic Programming (ILPS-97)*, pages 69–84. MIT Press, 1997.