Compiling reasoning with and about preferences into default logic

James P. Delgrande
School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
jim@cs.sfu.ca

Torsten H. Schaub*
Institut fur Informatik
Universitat Potsdam
Postfach 60 15 53, D-14415 Potsdam
torsten@cs. uni-potsdam. de

Abstract

We address the problem of introducing preferences into default logic. Two approaches are given, one a generalisation of the other. In the first approach, an *ordered default theory* consists of a set of default rules, a set of world knowledge, and a set of fixed preferences on the default rules. This theory is transformed into a second, standard default theory, where, via the naming of defaults, the given preference ordering on defaults is respected. In the second approach, we begin with a default theory where preference information is specified as part of an overall default theory. Here one may specify preferences that hold by default, or give preferences among preferences. Again, such a theory is translated into a standard default theory. The approach differs from previous work in that we obtain standard default theories, and do not rely on prioritised versions, as do other approaches. In practical terms this means we can immediately use existing default logic theorem provers for an implementation. From a theoretical point of view, this shows that the explicit representation of priorities adds nothing to the overall expressibility of default logic.

1 Introduction

In many situations in nonmonotonic reasoning the application of one default is preferred to another. Perhaps the best known example is inheritance of properties, where an individual is assumed to have properties by default according to the most specific class(es) to which it belongs. Hence an individual that is a penguin (and so a bird) does not fly by default, since penguins typically don't fly, even though birds do typically fly. Preferences are also found in decision making and in scheduling. For example, in scheduling not all deadlines may be simultaneously satisfiable; preferences then may allow some compromise solution. In legal reasoning laws may apply by default, but the laws themselves may conflict; such conflicts may be adjudicated by higher-level principles.

*Previously at LERIA, Universite d'Angers, France.

Our goal in this paper is to explore preference orderings in nonmonotonic reasoning, specifically in default logic [Reiter, 1980]. In the next section we examine the general notion of preference orderings on defaults. We note that there is not a single way in which preferences should be applied. Rather what we will call *specificity* orderings, as are used in property inheritance, can be distinguished from *preference* or *priority* orderings. These notions have frequently been conflated previously in the literature; here our concerns lie solely with preference orderings.

In considering how preference orderings may be enforced in default logic, we consider first where a default theory consists of world knowledge and a set of default rules, together with (external) preference information between default rules. We show how such a default theory can be translated into a second theory wherein preference information is now incorporated into the theory. So with this translation we obtain a theory in "standard" default logic, rather than requiring machinery external to default logic, as is found in previous approaches. We subsequently generalise this approach so that preferences may appear arbitrarily as part of a default theory and, specifically, preferences among default rules may (via the naming of default rules) themselves be part of a default rule. We again show how such a generalised default theory can be translated into a "standard" default theory where preference information is incorporated into the theory.

Previous approaches have generally added machinery to an extant approach to nonmonotonic reasoning. We remain within the framework of standard default logic, rather than building a scheme on top of default logic, for several reasons. First, there exist theorem provers for default logic. Consequently our approach can be immediately incorporated in such a prover. Second, it is easier to compare differing approaches to handling such orderings. Third, by "compiling" preferences into default logic, and in using the standard machinery of default logic, we obtain insight into the notion of preference orderings. Thus for example we implicitly show that explicit priorities provide no real increase in the expressibility of default logic.

2 Preference Orderings

This section discusses preference orderings in general; while we employ default logic, the discussion is independent of any particular approach to nonmonotonic reasoning. We can use default rules[1] to express that B follows by default from a by $\frac{\alpha:\beta}{\beta}$. Then we can write $\frac{\alpha:\beta}{\beta} < \frac{\gamma:\delta}{\delta}$ to express a preference between two defaults. Assume that we have an *ordered default theory.* The exact details are given in the next section; for the time being assume that we have a triple $(D, W, <)$, where D is a set of default rules, W a set of formulas, and < is a strict partial order on the default rules in D.

Informally a higher-ranked default should be applied or considered before a lower-ranked default. But what exactly does this mean? Consider for example the defaults concerning primary means of locomotion: "animals normally walk", "birds normally fly", "penguins normally swim":

$$\frac{Animal : Walk}{Walk} < \frac{Bird : Fly}{Fly} < \frac{Penguin : Swim}{Swim}. \qquad (1)$$

If we learn that some thing is penguin (and so a bird and animal), then we would want to apply the highest-ranked default, if possible, and only the highest-ranked default. Significantly, if the penguins-swim default is blocked (say the penguin in question has a fear of water) we *don't* try to apply the next default to see if it might fly. This then is standard inheritance of default properties.

The situation is very different in the next example. We have the defaults that "Canadians speak English by default" , "Quebecois speak French by default", "residents of the north of Quebec speak Cree by default":

$$\frac{Can : English}{English} < \frac{Que : French}{French} < \frac{NQue : Cree}{Cree}. \qquad (2)$$

Now if a resident of the north of Quebec didn't speak Cree, it would be reasonable to assume that that person spoke French, and if they didn't speak French, then English.

Assume that we have a chain of defaults $\delta_1 < \delta_2 < \ldots < \delta_m$. Informally, we have the following possibilities with respect to how the defaults may be applied.

[1] Default logic is introduced shortly.

PI For the maximum i for which si is applicable, and for $j > 0$, no s_{i+j} is denied,[2] apply S_i if possible." No other default is considered. This is the situation in (1).

P2 Apply s_m if possible; apply S_{m-i} if possible, continue in this fashion until no more than k (for fixed k where $1 < k < m$) defaults have been applied. This is the situation in (2) with $k = 1$.

An example of a general instance of P2 is where a student wishes to take $k = 3$ computing courses, out of m = 10 possible courses, and so provides a list of preferences over the courses. There are two important subcases of P2 corresponding to $k = 1$ and ib = m. In the first case a maximum of one default is applied. In the second case one attempts to apply every default.

PI is essentially (default property) inheritance. The ordering on defaults reflects a relation of *specificity;* one attempts to apply the most specific default possible. In approaches such as [Touretzky *et a*/., 1987; Pearl, 1990; Geffner & Pearl, 1992] specificity is determined implicitly, emerging as a property of the underlying system. [Reiter & Criscuolo, 1981; Etherington & Reiter, 1983; Delgrande & Schaub, 1994] have addressed adding specificity information in default logic. For incorporating preferences (as given in P2), [Boutilier, 1992; Brewka, 1994a; Baader & Hollunder, 1993] consider adding preferences in default logic while [McCarthy, 1986; Lifschitz, 1985; Grosof, 1991] do the same in circumscription. We note however that some of these latter papers include examples best interpreted as dealing with specificity (as given in (1)), and so would appear to conflate PI and P2.

Our concerns in this paper are with specifying preferences and priorities, as given in P2. We assume only that we are given a set of defaults and a priority policy on defaults, along with other world knowledge. We observe that the framework as given in P2 is significantly more general that that of P1. For example it seems to be an intrinsic property of inheritance as given in P1 that the ordering on defaults is determined by relative specificity of the prerequisites. So if $S_1 < S_2$ then the antecedent of S_1 is less specific than the antecedent of S_2. This is not the case for P2 though. Consider a variation on (2) where in the north of Quebec the first language is French, then English, then Cree: The resulting preference ordering is as follows.

$$\frac{NQue : Cree}{Cree} < \frac{Can : English}{English} < \frac{Que : French}{French}. \qquad (3)$$

Indeed for preferences, one need not have any antecedent information. That one prefers something (say, a car) that is red, then green might be expressed as $\frac{:Green}{Green}$ <

[2] i.e. we don't have that the prerequisite is true and the consequent false.

$\frac{:Red}{Red}$. In the most general case, we might have two defaults, with no relation between them, except for some given preference relation.

3 Default Logic and Ordered Default Logic

Default logic [Reiter, 1980] augments classical logic by *default rules* of the form $\frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma}$. For the most part we deal with *singular* defaults for which $n = 1$. [Marek & Truszczyński, 1993] show that any default rule can be transformed into a set of singular defaults; hence our one use of a non-singular rule in Section 5 is for notational convenience only. A singular rule is *normal* if β is equivalent to γ; it is *semi-normal* if β implies γ. We sometimes denote the *prerequisite* α of a default δ by $Prereq(\delta)$, its *justification* β by $Justif(\delta)$ and its *consequent* γ by $Conseq(\delta)$. Empty components, such as no prerequisite or even no justifications, are assumed to be tautological. 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 or multiple *extensions* in the following way.

Definition 3.1 *Let (D, W) be a default theory and let E be a set of formulas. Define $E_0 = W$ and for $i \geq 0$:*

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

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

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[3] $\Gamma = \bigcup_{i=0}^{\infty} \Gamma_i$ as the set of default rules *generating* extension E.

For adding preferences among default rules, a default theory is usually extended with an ordering on the set of default rules. In analogy to [Baader & Hollunder, 1993; Brewka, 1994a], an *ordered default theory* $(D, W, <)$ is a finite set D of default rules, a set W of formulas, and a strict partial order $< \subseteq D \times D$ on the default rule. For simplicity we assume the existence of a default $\delta_\top = \frac{\top:\top}{\top} \in D$ where for every rule $\delta_i \neq \delta_\top$ we have $\delta_i < \delta_\top$. This gives us a (trivial) maximally preferred default that is always applicable to "start things off".

4 Static Preferences on Defaults

We show here how ordered default theories can be translated into standard default theories. Our strategy is to add sufficient "hooks" to a default rule in a theory to enable the control of rule application. We begin with an ordered default theory $(D, W, <)$ which is then translated into a standard default theory (D', W') such that the explicit preferences in $<$ are "compiled" into D' and W'.

To this end, we associate a unique name with each default rule. This is done by extending the original language by a set of constants N[4] such that there is a bijective mapping $n : D \rightarrow N$. We write n_δ instead of $n(\delta)$ (and we often abbreviate n_{δ_i} by n_i to ease notation). For default rule δ along with its name n, we sometimes write $n : \delta$ to render naming explicit. To reflect the fact that we deal with a finite set of distinct default rules, we adopt a unique names assumption (UNA) and domain closure assumption (DCA) with respect to N. That is, for a name set $N = \{n_1, \ldots, n_m\}$, we add axioms $\forall x.\ name(x) \equiv (x = n_1 \vee \ldots \vee x = n_m)$ and $(n_i \neq n_j)$ for all $n_i, n_j \in N$ with $i \neq j$. For convenience, we then write $\forall x \in N.\ P(x)$ instead of $\forall x.\ name(x) \supset P(x)$. In the sequel, we use DCA_N and UNA_N to abbreviate domain closure and unique names axioms for N.

The use of names allows the expression of preference relations between default rules in the object language. So we can assert that default $n_j : \frac{\omega:\phi}{\varphi}$ is preferred to $n_i : \frac{\alpha:\beta}{\gamma}$ by $n_i \prec n_j$, where $\prec$ is a (new) predicate in the object language. Finally, in discussions of preference relations, we sometimes write $\delta_i < \delta_j$ or $\frac{\alpha:\beta}{\gamma} < \frac{\omega:\phi}{\varphi}$, to show a preference between two defaults; however it should be kept in mind that these latter expressions are not expressions *within* a default theory (as are given by $\prec$), but expressions *about* a default theory.

If we are given $\delta_i < \delta_j$, then we want to ensure that before δ_i is applied, that δ_j be applied or found to be inapplicable.[5] We do this by first translating default rules so that rule application can be explicitly controlled. 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 bl(·). Similarly we introduce a special-purpose predicate ap(·) to detect the case where a rule has been applied. For controlling application of a rule we introduce predicate ok(·). Then, a default rule $\delta = \frac{\alpha:\beta}{\gamma}$ is

[3] For simplicity, we refrain from parameterizing Γ by D and E.

[4] This is done also in [Brewka, 1994b]. Theorist [Poole, 1988] uses atomic propositions to name defaults.

[5] That is, we wish to exclude the case where $\delta_i \in \Gamma_n$ and $\delta_j \in \Gamma_m$ for $n \leq m$.

mapped to

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

These rules are sometimes abbreviated by $\delta_a, \delta_{b_1}, \delta_{b_2}$, resp.

None of the three rules in the translation can be applied unless ok(n) is true. Since ok(-) 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 ok(n) is true, the first rule of the translation may potentially be applied. If a rule has been applied, then this is "recorded" by assertion ap(n). The last two rules give conditions under which the original rule is inapplicable: either the negation of the original antecedent a is consistent (with the extension) or the justification β is known to be false; in either such case $\mathsf{bl}(n)$ is concluded.

This translation clearly says nothing about which default precedes what in order of application. However, for $\delta_i < \delta_j$ we can now fully control the order of rule application: 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. So we would have something like $(\mathsf{ap}(n_j) \vee \mathsf{bl}(n_j)) \supset \mathsf{ok}(n_i)$, but adjusted to allow for the fact that we might have other rules with higher priority than δ_i. Further, given that $\delta_1 < \delta_2 < \delta_3$ we would want the order of application to be δ_3 then δ_2 then δ_1. Given the predicates $\mathsf{bl}(\cdot)$ and $\mathsf{ap}(\cdot)$ it is a straightforward matter to also assert that a maximum of one default in a priority order can be applied, or in the general case that *k* rules can be applied.

Taking all this into account, 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 *Given an ordered default theory $(D, W, <)$ over $\mathcal{L}$ and its set of default names $N = \{n_\delta \mid \delta \in D\}$, define $T((D, W, <)) = (D', W')$ over $\mathcal{L}^+$ by*

$$D' = \left\{ \frac{\alpha \wedge \mathsf{ok}(n) : \beta}{\gamma \wedge \mathsf{ap}(n)}, \frac{\mathsf{ok}(n) : \neg\alpha}{\mathsf{bl}(n)}, \frac{\neg\beta \wedge \mathsf{ok}(n) :}{\mathsf{bl}(n)} \,\middle|\, n : \frac{\alpha : \beta}{\gamma} \in D \right\} \cup D_\prec$$

$$W' = W \cup W_\prec \cup \{DCA_N, UNA_N\}$$

where

$$D_\prec = \left\{ \frac{: \neg(x \prec y)}{\neg(x \prec y)} \right\}$$

$$W_\prec = \{n_\delta \prec n_{\delta'} \mid (\delta, \delta') \in <\} \;\cup\; \{\mathsf{ok}(n_\top)\}$$
$$\cup\; \{\forall x \in N. \, [\forall y \in N. \, (x \prec y) \supset (\mathsf{bl}(y) \vee \mathsf{ap}(y))] \supset \mathsf{ok}(x)\}$$

W' contains prior world knowledge, together with assertions for managing the priority order $<$ on defaults. The first part of $W_\prec$ specifies that $\prec$ is a predicate drawn from 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.

This alone however gives necessary but not sufficient conditions for rendering δ_i potentially applicable. If $(\delta_i, \delta_j) \notin <$ then $(n_i \prec n_j) \notin W_\prec$; however, for the last formula in $W_\prec$ to work properly we must be able to conclude (in the extension) that $\neg(n_i \prec n_j)$. This is addressed by adding the default rule in $D_\prec$ that renders the resulting theory complete with respect to priority statements. That is, for all resulting extensions E we have that $(n_i \prec n_j) \in E$ or $\neg(n_i \prec n_j) \in E$. We also have $(n_\delta \prec n_\top) \in W'$ for every rule $\delta \neq \delta_\top$ by the definition of ordered default theories. Since $<$ is a strict partial order, W' also includes the transitive closure of $\prec$ and no reflexivities, such as $n \prec n$.

As an example, consider the defaults:

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

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

$$\frac{A_i \wedge \mathsf{ok}(n_i) : B_i}{C_i \wedge \mathsf{ap}(n_i)} \quad \frac{\mathsf{ok}(n_i) : \neg A_i}{\mathsf{bl}(n_i)} \quad \frac{\neg B_i \wedge \mathsf{ok}(n_i) :}{\mathsf{bl}(n_i)}$$

and analogously for $\delta_\top$ where A_i, B_i, C_i are $\top$. Given that $\delta_1 < \delta_2 < \delta_3$, 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 $(\mathsf{ap}(n_3) \vee \mathsf{bl}(n_3)) \supset \mathsf{ok}(n_2)$, and $((\mathsf{ap}(n_2) \vee \mathsf{bl}(n_2)) \wedge (\mathsf{ap}(n_3) \vee \mathsf{bl}(n_3))) \supset \mathsf{ok}(n_1)$ is contained in E_2 in Definition 3.1; from this we get that n_3 must be applied first, followed by n_2 and then n_1.

Now, given A_1, A_2, A_3, we obtain subsequently $C_3 \wedge \mathsf{ap}(n_3) \in E_3$, $\mathsf{ok}(n_2) \in E_4$, $C_2 \wedge \mathsf{ap}(n_2) \in E_5$, $\mathsf{ok}(n_1) \in E_6$, $C_1 \wedge \mathsf{ap}(n_1) \in E_7$. Given additionally $C_3 \supset \neg B_2$ and $C_2 \supset \neg B_3$, we get $\mathsf{bl}(n_2) \in E_5$ instead of $C_2 \wedge \mathsf{ap}(n_2) \in E_5$ because $\neg B_2 \in E_4$. Suppose there is an extension containing $C_2 \wedge \neg B_3$ as opposed to $C_3 \wedge \neg B_2$. Then however we have neither $\mathsf{ap}(n_3) \in E_3$ nor $\mathsf{bl}(n_3) \in E_3$, which makes it impossible to derive $\mathsf{ok}(n_2)$ and thus $\neg B_3$ cannot belong to such an extension.

The following theorems summarize the major properties of our approach, and demonstrate that rules are applied in the desired order:

Theorem 4.1 *Let E be an extension of $T((D, W, <))$ for ordered default theory $(D, W, <)$. We have for all $\delta, \delta' \in D$*

1. $n_\delta \prec n_{\delta'} \in E$ *or* $\neg(n_\delta \prec n_{\delta'}) \in E$

2. $\mathsf{ok}(n_\delta) \in E$

3. *either* $\mathsf{ap}(n_\delta) \in E$ *or* $\mathsf{bl}(n_\delta) \in E$

4. $\mathsf{ok}(n_\delta) \in E_i$ *and* $Prereq(\delta) \in E_j$ *and* $\neg Justif(\delta) \notin E$ *implies* $\mathsf{ap}(n_\delta) \in E_{\max(i,j)+1}$

5. $\mathsf{ok}(n_\delta) \in E_i$ *and* $Prereq(\delta) \notin E$ *implies* $\mathsf{bl}(n_\delta) \in E_{i+1}$

6. $\mathsf{ok}(n_\delta) \in E_i$ *and* $\neg Justif(\delta) \in E$ *implies* $\mathsf{bl}(n_\delta) \in E_j$ *for some* $j > i$

7. $\mathsf{ok}(n_\delta) \notin E_{i-1}$ *and* $\mathsf{ok}(n_\delta) \in E_i$ *implies* $\mathsf{ap}(n_\delta), \mathsf{bl}(n_\delta) \notin E_j$ *for* $j \leq i$

Notably, Theorem 4.1.5 allows us to detect blockage due to non-derivability of the prerequisite immediately after having the "ok" for the default at hand. For an extension E and its generating default rules T, we trivially have $\delta_a \in \Gamma$ iff $\mathsf{ap}(n_\delta) \in E$, and $\delta_{b_1} \in \Gamma$ or $\delta_{b_2} \in \Gamma$ iff $\mathsf{bl}(n_\delta) \in E$.

Theorem 4.2 *Let E be an extension of $\mathcal{T}((D,W,<))$ for ordered default theory $(D,W,<)$ and Γ, Γ_i be defined wrt E. Then, we have for all $\delta \in D$*

8. $\delta_a \in \Gamma$ *or* $\delta_{b_1} \in \Gamma$ *or* $\delta_{b_2} \in \Gamma$

9. $\delta_a \in \Gamma$ *iff* $(\delta_{b_1} \notin \Gamma$ *and* $\delta_{b_2} \notin \Gamma)$

For all default rules $\delta, \delta' \in D$ such that $\delta < \delta'$, we have

10. $\delta'_a, \delta'_{b_1}, \delta'_{b_2} \notin \Gamma_i$ *implies* $\delta_a, \delta_{b_1}, \delta_{b_2} \notin \Gamma_j$ *for* $j < i+3$

11. $\delta'_a \in \Gamma_i$ *or* $\delta'_{b_1} \in \Gamma_i$ *or* $\delta'_{b_2} \in \Gamma_i$ *implies* $\delta_a \in \Gamma_j$ *or* $\delta_{b_1} \in \Gamma_j$ *or* $\delta_{b_2} \in \Gamma_j$ *for some* $j > i+1$

12. $\delta_a \in \Gamma_i$ *or* $\delta_{b_1} \in \Gamma_i$ *or* $\delta_{b_2} \in \Gamma_i$ *implies* $\delta'_a \in \Gamma_j$ *or* $\delta'_{b_1} \in \Gamma_j$ *or* $\delta'_{b_2} \in \Gamma_j$ *for some* $j < i-1$.

The minimum two-step delay between rules stemming from δ and those originated by δ' is due to to the fact that in Definition 3.1 the deductive closure of E_i is determined at E_{i+1}. The important overall consequence of this series of propositions is that we have full control over default application.

Using the above properties, we can show that any extension of a translated default theory is a regular extension of the underlying *unordered* default theory:

Theorem 4.3 *Let E be an extension of $\mathcal{T}((D,W,<))$ for ordered default theory $(D,W,<)$ over $\mathcal{L}$. Then $E \cap \mathcal{L}$ is an extension of (D,W).*

The approach is equivalent (modulo the original language) to standard default logic if there are no preferences:

Theorem 4.4 *For a default theory (D,W) over $\mathcal{L}$ and a set of formulas E, we have that E is an extension of $\mathcal{T}((D,W,\emptyset))$ iff $E \cap \mathcal{L}$ is an extension of (D,W).*

5 Dynamic Preferences on Defaults

We now consider situations where the presence of preferences is context-dependent. We deal with standard default theories (D,W) over a language already including a predicate $\prec$ expressing a preference relation by means of default names. In order to keep a finite domain closure axiom, we restrict ourselves to a finite set of default rules D being in one-to-one correspondence with a finite name set N.

Since preferences are now available dynamically by inferences from W and D, we lack a priori complete information about the ordering predicate $\prec$, which was available in the rigid case by appeal to the explicit order $<$ between rules and the "closed world default" $\frac{:\neg(x\prec y)}{\neg(x\prec y)}$. This however leads to a problem. Consider where our only preference is given by $\frac{:n\prec m}{n\prec m}$. Either the default would apply or it would not; in either case we would expect one extension only. However we also have the "closed world" default for preferences, given in $D_\prec$, that asserts that if there is no known or derived preference between rules, then no preference exists. An instance of $D_\prec$ is $\frac{:\neg(n\prec m)}{\neg(n\prec m)}$. So if we simply have these two defaults then we run the risk of potentially having an unwanted extension where $\frac{:\neg(n\prec m)}{\neg(n\prec m)}$ applies over $\frac{:n\prec m}{n\prec m}$. Obviously we can't solve the problem by asserting that $\frac{:\neg(n\prec m)}{\neg(n\prec m)} < \frac{:n\prec m}{n\prec m}$ since our approach would now be circular. We address this issue by adding a new binary predicate $\not\prec$ indicating that for defaults δ and δ' neither $(n_\delta \prec n_{\delta'}) \in E$ nor $\neg(n_\delta \prec n_{\delta'}) \in E$. We add the following rule, where x, y are variables ranging over default names:

$$\frac{:\neg(x\prec y), (x\prec y)}{(x\not\prec y)} \tag{4}$$

This rule accounts for situations where neither $(x \prec y)$ nor $\neg(x \prec y)$ is derivable. That is for names n and n', the only time this rule will apply is when $n \prec n' \notin E$ and $\neg(n \prec n') \notin E$. So, since $\not\prec$ is an introduced predicate, the only time we have $n \not\prec n' \in E$ is when the default theory has no information on whether the two defaults are in a preference relation or not.

We now consider standard default theories in a language C including the set of default names and propositions formed by binary predicate $\prec$ applied to variables and default names only; these are mapped onto theories in the language $\mathcal{L}^*$ obtained by extending $\mathcal{L}$ with new predicate symbols $(\cdot \not\prec \cdot)$, $\mathsf{ok}(\cdot)$, $\mathsf{bl}(\cdot)$, and $\mathsf{ap}(\cdot)$:

Definition 5.1 *Given a default theory (D,W) over $\mathcal{L}$ and its set of default names $N = \{n_\delta \mid \delta \in D\}$, we define $\mathcal{T}((D,W)) = (D',W')$ over $\mathcal{L}^*$ by*

$$D' = \left\{ \frac{\alpha \wedge \mathsf{ok}(n) : \beta}{\gamma \wedge \mathsf{ap}(n)}, \frac{\mathsf{ok}(n) : \neg\alpha}{\mathsf{bl}(n)}, \frac{\neg\beta \wedge \mathsf{ok}(n) :}{\mathsf{bl}(n)} \,\middle|\, n : \frac{\alpha : \beta}{\gamma} \in D \right\} \cup D_\prec$$

$$W' = W \cup W_\prec \cup \{DCA_N, UNA_N\}$$

where

$$
\begin{aligned}
D_{\prec} &= \left\{ \frac{:\neg(x \prec y), (x \prec y)}{(x \not\prec y)} \right\} \\
W_{\prec} &= \{\forall x \in N. \neg(x \prec x)\} \\
&\cup \{\forall xyz \in N. ((x \prec y) \wedge (y \prec z)) \supset (x \prec z)\} \\
&\cup \{\forall x \in N. (x \neq n_{\top}) \supset (x \prec n_{\top})\} \cup \{\mathsf{ok}(n_{\top})\} \\
&\cup \{\forall x \in N. (\forall y \in N. (x \not\prec y) \vee \\
&\qquad [(x \prec y) \supset (\mathsf{bl}(y) \vee \mathsf{ap}(y))]) \supset \mathsf{ok}(x)\}
\end{aligned}
$$

In contrast to Definition 4.1, D and W now may contain preference information expressed by $\prec$ applied to default names. The first three axioms in $W_{\prec}$ account for information that was explicitly provided by ordered default theories in the rigid case. The last axiom is a straightforward extension of that found in the rigid case, now also accounting for the information provided by the default rule in $D_{\prec}$.

We note that Theorem 4.1 and Theorem 4.2 carry over to the general case except for Theorem 4.1.1. We get instead

$$1'.\ n_{\delta} \prec n_{\delta'} \in E \text{ or } \neg(n_{\delta} \prec n_{\delta'}) \in E \text{ or } n_{\delta} \not\prec n_{\delta'} \in E$$

In fact, ordered default theories are treated in the same way by our basic and general approach, except for different augmented languages:

Theorem 5.1 *Let $(D, W, <)$ be an ordered default theory over $\mathcal{L}$. For each extension E of $\mathcal{T}((D, W, <))$ there is an extension E' of $\mathcal{T}((D, W \cup \{n_{\delta} \prec n_{\delta'} \mid (\delta, \delta') \in <\}))$ such that $E \cap \mathcal{L} = E' \cap \mathcal{L}$ and vice versa.*

As a consequence, our general approach yields all regular extensions (modulo the original language) if (D, W) does not contain an occurrence of $\prec$.

Given Theorems 4.3 and 5.1, one would expect that ordered default theories would enjoy the same properties as standard default logic. This indeed is the case, but with one important exception: normal ordered default theories do not guarantee the existence of extensions. For example, the image of the ordered default theory (under our translation)

$$\left(\left\{n_1 : \frac{A : B}{B}, n_2 : \frac{B : C}{C}\right\}, \{A\}, n_1 < n_2\right) \qquad (5)$$

has no extension. The full paper describes formally why this is the case. Informally the problem is that we have a preference $n_1 < n_2$; however, given $W = \{A\}$ default n_1 applies first, and once it has applied, n_2 becomes applicable. Thus we have a preference ordering implicit in the form of the defaults and world knowledge, but where this implicit ordering is contradicted by the assertion $n_1 < n_2$. Not surprisingly then there is no extension. There are two immediate ways to resolve this problem. First, replace the rigid preference by $\frac{: n_1 \prec n_2}{n_1 \prec n_2}$; so we have this preference by default only. Second is to recognise that (5) is "buggy", in the same way that incorrect programs require modification. The lack of extension then indicates a problem in the specification of the original theory.

We conclude this section with the observation that our 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).[6]

6 Discussion and Related Work

We have presented a very general framework for incorporating preferences into default logic. Via the naming of defaults we allow preferences to appear arbitrarily in *D* and *W* in a default theory. This allows preferences among preferences, preferences by default, preferences holding only in certain contexts, and so on. Strictly speaking, such generality isn't required: [Doyle *k* Wellman, 1991], building on work by Arrow, argue that in any preference-based default theory, for coherence, one requires a "dictator" to adjudicate preferences. That is, there must be, essentially, some way of determining a unique, complete, priority ordering. So in this sense, all one needs is what we have called the rigid approach of Section 4. We provide the more general framework of Section 5 for two reasons. First, it allows the more flexible specification of preferences, leaving it up to the user to ensure that there is no ambiguity in preferences. In the case where there is ambiguity in preferences (for example we might have $D \supseteq \{\frac{: n_1 \prec n_2}{n_1 \prec n_2}, \frac{: n_2 \prec n_1}{n_2 \prec n_1}\}$) one typically obtains multiple extensions. Second, we feel that the general approach is of technical interest: arbitrary defaults may be "compiled" into standard default theories, and so in a certain sense the explicit representation of priorities adds nothing to the fundamental power or expressibility of default logic.

Of other work in default logic treating preferences, we have argued that [Reiter *k* Criscuolo, 1981; Etherington *k* Reiter, 1983; Delgrande *k* Schaub, 1994] treat a separate problem, that of specificity orderings, as exemplified by (1). [Baader *k* Hollunder, 1993] and [Brewka, 1994a] present prioritised variants of default logic in which the iterative specification of an extension is modified. In brief, a default is only applicable at an iteration step (cf. Definition 3.1) if no <-greater default is applicable.[7] In contrast we translate priorities into standard default theories. There is insufficient space to fully compare approaches; see [Delgrande *k* Schaub, 1994] for a full discussion of these approaches with regard to how they address specificity in a theory.

[6] This assumes we count the default in $D_{\prec}$ as a single default.

[7] These authors use $<$ in the reverse order from us.

We conclude with an example from [Gordon, 1993], discussed in [Brewka, 1994b]. A person wants to find out if her security interest in a certain ship is "perfected", or legally valid. This person has possession of the ship, but has not filed a financing statement. According to the code UCC, a security interest can be perfected by taking possession of the ship. However, the federal Ship Mortgage Act (SMA) states that a security interest in a ship may only be perfected by filing a financing statement. Both UCC and SMA are applicable; the question is which takes precedence here. There are two legal principles for resolving such conflicts. *Lex Posterior* gives precedence to newer laws; here we have that UCC is more recent than SMA. But *Lex Superior* gives precedence to laws supported by the higher authority; here SMA has higher authority since it is federal law. Apart from $\delta_\top$, we get:

$$ucc : \frac{possess : perf}{perf} \qquad sma : \frac{ship \wedge \neg finstmt : \neg perf}{\neg perf}$$

$$lp(x, y) : \frac{newer(y, x) : x \prec y}{x \prec y} \qquad ls(x, y) : \frac{stalaw(x) \wedge fedlaw(y) : x \prec y}{x \prec y}$$

To preserve finiteness, we restrict our attention to name set $N = N_0 \cup N_1$ where $N_0 = \{n_\top, ucc, sma\}$ and $N_1 = \{lp(x, y), ls(x, y) \mid x, y \in N_0\}$, and the corresponding default instances. We have the facts: $possess$, $ship$, $\neg finstmt$, $newer(ucc, sma)$, $fedlaw(sma)$, $stalaw(ucc)$, $\forall x, y, u, v \in N_0.\ lp(x, y) \prec ls(u, v)$. [Brewka, 1994b] solves this problem by first generating 4 *complete* extensions. In a second step he rules out three of these extensions since they do not satisfy a certain priority criterion. In contrast, we obtain one extension, $E \supseteq \{\neg perf, ucc \prec sma\}$, and no other extension; this is the extension that Brewka ultimately obtains after ruling out non-preferred extensions.

References

[Baader *k* Hollunder, 1993] F. Baader *k* B. Hollunder. How to prefer more specific defaults in terminological default logic. *Proc. Int'l Joint Conference on Artificial Intelligence,* p. 669-674, 1993.

[Boutilier, 1992] C. Boutilier. What is a default priority? *Canadian Conference on AI,* p. 140-147, 1992.

[Brewka, 1994a] G. Brewka. Adding priorities and specificity to default logic. *Proc. European Workshop on Logics in Artificial Intelligence,* p. 247-260. Springer, 1994.

[Brewka, 1994b] G. Brewka. Reasoning about priorities in default logic. *Proc. AAAI Conference on Artificial Intelligence,* p. 940-945, 1994.

[Delgrande *k* Schaub, 1994] J. Delgrande *k* T. Schaub. A general approach to specificity in default reasoning. *Proc. Fourth Int;Inference on Knowledge Representation and Reasoning,* p. 146-157. Morgan Kaufmann, 1994.

[Doyle *k* Wellman, 1991] J. Doyle *k* M.R Wellman. Impediments to universal preference-based default theories. *Artificial Intelligence,* 49(l-3):97-128, 1991.

[Etherington *k* Reiter, 1983] D.W. Etherington *k* R. Reiter. On inheritance hierarchies with exceptions. *Proc. AAAI Conference on Artificial Intelligence,* p. 104-108, 1983.

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

[Gordon, 1993] T. Gordon. *The pleading game: An Artificial Intelligence Model of Procedural Justice.* PhD thesis, University of Darmstadt, Germany, 1993.

[Grosof, 1991] B. Grosof. Generalizing prioritization. *Proc. Second InVl Conference on Knowledge Representation and Reasoning,* p. 289-300. Morgan Kaufmann, 1991.

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

[Marek *k* Truszczyiiski, 1993] W. Marek *k* M. Truszczynski. *Nonmonotonic logic: context-dependent reasoning.* Springer, 1993.

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

[Pearl, 1990] J. Pearl. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. *Proc. of the Third Conference on Theoretical Aspects of Reasoning About Knowledge,* p. 121-135, 1990.

[Poole, 1988] D.L. Poole. A logical framework for default reasoning. *Artificial Intelligence,* 36(l):27-48, 1988.

[Reiter *k* Criscuolo, 1981] R. Reiter *k* G. Criscuolo. On interacting defaults. *Proc. Int'l Joint Conference on Artificial Intelligence,* p. 270-276, 1981.

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

[Touretzky *et al,* 1987] D.S. Touretzky, J.F. Horty, *k* R.H. Thomason. A clash of intuitions: The current state of nonmonotonic multiple inheritance systems. *Proc. InVl Joint Conference on Artificial Intelligence,* p. 476-482, 1987.