How to reason credulously and skeptically within a single extension

James P. Delgrande[1] and Torsten Schaub[2]*

[1] School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, jim@cs.sfu.ca
[2] Institut für Informatik, Universität Potsdam, Postfach 60 15 53, D–14415 Potsdam, Germany, torsten@cs.uni-potsdam.de

Abstract. Consistency-based approaches in nonmonotonic reasoning may be expected to yield multiple sets of default conclusions for a given default theory. Reasoning about such extensions is carried out at the meta-level. In this paper, we show how such reasoning may be carried out at the object level for a large class of default theories. Essentially we show how one can translate a (normal) default theory Δ, obtaining a second Δ', such that Δ' has a single extension that encodes every extension of Δ. Moreover, our translated theory is only a constant factor larger than the original (with the exception of unique names axioms). We prove that our translation behaves correctly. In the approach we can now encode the notion of *extension* from within the framework of standard default logic. Hence one can encode notions such as skeptical and credulous conclusions, and can reason about such conclusions within a single extension. This result has some theoretical interest, in that it shows how multiple extensions of normal default theories are encodable with manageable overhead in a single extension.

1 Introduction

In nonmonotonic reasoning, in so-called *consistency-based* approaches such as default logic [9] and autoepistemic logic [6], one typically obtains not just a single set of default conclusions, but rather multiple sets of candidate default conclusions. Consider the by-now hackneyed example wherein Quakers are normally pacifist, republicans are normally not, along with adults are normally employed. Assume as well that someone is a Quaker, republican, and an adult. In default logic (see Section 2) this can be encoded by: $(\{\frac{Q:P}{P}, \frac{R:\neg P}{\neg P}, \frac{A:E}{E}\}, \{Q, R, A\})$. This theory has two *extensions* or sets of default conclusions, one containing $\{Q, R, A, E, P\}$ and the other $\{Q, R, A, E, \neg P\}$. In autoepistemic logic the same example appropriately encoded yields two analogous *expansions* or possible belief sets.

Reasoning about these extensions (resp. expansions) is carried out at the meta-level: a default conclusion that appears in some extension (such as P) is called a *credulous* (or *brave*) default conclusion, while one that appears in every extension (such as E) is called a *skeptical* conclusion. Intuitively it might seem that skeptical inference is the more useful notion. However, this is not necessarily the case. In diagnosis from first

* Affiliated with Simon Fraser University, Burnaby, Canada.

principles [10] for example, in one encoding there is a 1-1 correspondence between diagnoses and extensions of the (encoding) normal default theory. Hence one may want to carry out further reasoning to determine which diagnosis to pursue. More generally there may be reasons to prefer some extensions over others, or to somehow synthesize the information found in several extensions.

In this paper, we show how such reasoning can be carried out at the object level. For a default theory $\Delta = (D, W)$, we translate Δ to obtain a second theory $\Delta' = (D', W')$, such that Δ' has a single extension that encodes every extension of Δ. Given this, one can express in the theory what it means for something to be a skeptical or credulous default conclusion. Our result isn't completely general; however it applies to *normal* default theories. The translation has several desirable properties. The translated theory Δ' is only a constant factor larger than the original Δ, with the exception of introduced unique names axioms. As well, we *prove* that our translation behaves correctly.

We first show for a set of defaults D_m how, using an encoding, we can detect the case wherein all defaults in D_m apply. From this, for a default theory $(D \cup D_m, W)$ we show how to obtain a second theory wherein (informally) either all of the defaults in D_m are applied en masse (if possible) or none of them are. This is done by naming each of the defaults in D_m, and then expressing in default logic the applicability conditions for the defaults. We develop this in Section 3. In Section 4 we present our main result, where we show how a default theory can be translated into a second theory whose extension encodes the extensions of the original. Roughly we provide an axiomatisation that "locates" maximal sets of applicable defaults; for such a set, the set of default conclusions is "tagged" with the set name, to distinguish it from other instances. For example, in our original example, let $m_{1,3}$ be the name of the set $\{\frac{Q:P}{P}, \frac{A:E}{E}\}$ and $m_{2,3}$ be the name of $\{\frac{R:\neg P}{\neg P}, \frac{A:E}{E}\}$. These are maximal applicable sets of defaults, and from our translation we would obtain a single extension containing $\{Q(m_{1,3}), Q(m_{2,3}), R(m_{1,3}), R(m_{2,3}), A(m_{1,3}),\ A(m_{2,3}),\ E(m_{1,3}),$ $E(m_{2,3}), P(m_{1,3}), \neg P(m_{2,3})\}$. As mentioned, we are able to prove that our translations in fact accomplish what is claimed.

The advantage of this approach is that we can encode the notion of extension within the framework of standard default logic. Hence one can reason about (skeptical and credulous) conclusions within the framework of a single extension of a default theory. Thus for example, in a diagnosis setting one could go on and axiomatise notions of preference among diagnoses having to do with, perhaps, number of faulty components, or based on components expected to fail first. This result has some theoretical interest, in that it shows (for theories that we consider) how multiple extensions are encodable, with no significant overhead in a single extension. The overall approach is similar to that of [2].

2 Default Logic

Default logic [9] augments classical logic by *default rules* of the form $\frac{\alpha : \beta}{\gamma}$. A default 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 *PRE*(δ), its *justification* β by *JUS*(δ), and its *consequent* γ by *CON*(δ). Accordingly, *PRE*(D) is the set of prerequisites of all defaults in D; *JUS*(D) and *CON*(D) are defined analogously. Empty components, such as no prerequisite or even no justifications, are assumed to be tautological. Semantically, 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 [9].

Definition 1. *Let $\Delta = (D, W)$ be a default theory. For any set of formulas S, let $\Gamma_\Delta(S)$ be the smallest set of formulas S' such that*

1. $W \subseteq S'$,
2. $\mathit{Th}(S') = S'$,
3. *For any* $\frac{\alpha : \beta}{\gamma} \in D$, *if* $\alpha \in S'$ *and* $\neg\beta \notin S$ *then* $\gamma \in S'$.

A set of formulas E is an extension of Δ iff $\Gamma_\Delta(E) = E$.

Any such extension represents a possible set of beliefs about the world at hand. Further, define for a set of formulas S and a set of defaults D, the *set of generating default rules* as $\mathit{GD}(D, S) = \{\delta \in D \mid \mathit{PRE}(\delta) \in S \text{ and } \neg\mathit{JUS}(\delta) \notin S\}$.

3 Applying All, or None, of a Set of Defaults

In this section we consider the problem of how to apply all defaults in some set, or none in the set. We will thus work with default theories (D, W) having some distinguished finite subset $D_m \subseteq D$. For making the set D_m explicit, we denote such theories by $(D \cup D_m, W)$. The idea is that we wish to obtain extensions of $(D \cup D_m, W)$ subject to the constraint that *all* defaults in D_m are applied, or *none* are. For example, in the theory $\left(\left\{\frac{:A}{A}\right\} \cup \left\{\frac{:B}{B}, \frac{C:D}{D}\right\}, \emptyset\right)$ we would want to obtain an extension containing A, but not B (since both defaults in $\left\{\frac{:B}{B}, \frac{C:D}{D}\right\}$ cannot be jointly applied). For $\left(\left\{\frac{:A}{A}\right\} \cup \left\{\frac{:B}{B}, \frac{C:D}{D}\right\}, \{C\}\right)$ we would want to obtain an extension containing A, B, and D.

We begin by associating a unique name with each default. This is done by extending the original language by a set of constants[1] N such that there is a bijective mapping $n : D \to N$. We write n_δ instead of $n(\delta)$ (and we often abbreviate n_{δ_i} by n_i to ease notation). Also, for default δ along with its 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 assertion (UNA$_N$) and domain closure assertion (DCA$_N$) with respect to N. So, for a name set $N = \{n_1, \ldots, n_k\}$, we add axioms

$$\begin{array}{l} \text{UNA}_N : \neg(n_i = n_j) \;\text{ for all }\; n_i, n_j \in N \;\text{ with }\; i \neq j \\ \text{DCA}_N : \forall x.\, name(x) \equiv (x = n_1 \vee \cdots \vee x = n_k)\ . \end{array}$$

We write $\forall x \in N.\, P(x)$ for $\forall x.\, name(x) \supset P(x)$.

We introduce a new constant m as the name of the designated rule set D_m. We relate the name of the rule set denoted by m with the names of its members by introducing a binary predicate in where $in(x, y)$ is true just if the default named by x is a member

[1] [5] first suggested naming defaults using a set of *aspect* functions. See also [8, 1].

of the set named by y. In this section, instances of in will be of the form $in(\cdot, m)$. While we could get away with not using in (and m) here, this additional machinery is required in Section 4, and it is most straightforward to introduce it here. Note that we do not need a full axiomatization of in, representing set membership, since we use it in a very restricted fashion.

For applying all, or none, of the defaults in D_m, we need to be able to, first, detect when a rule has been applied or is blocked and, second, control the application of a rule based on other prerequisite conditions. There are two cases for a default $\frac{\alpha:\beta}{\gamma}$ to not be applied: the prerequisite is not known to be true (and so its negation $\neg\alpha$ is consistent), or the justification is not consistent (and so its negation $\neg\beta$ is derivable). For detecting this case, we introduce a new, special-purpose predicate $\mathsf{bl}/1$. Similarly we introduce a special-purpose predicate $\mathsf{ap}/1$ to detect when a rule has been applied. For controlling application of a rule we introduce predicates $\mathsf{ok}/1$ and $\mathsf{ko}/1$.

We are given a default theory $(D \cup D_m, W)$ over language $\mathcal{L}$ and its set of associated default names $N \dot\cup \{m\}$. [2] Let

$$D_m = \left\{ n_j : \tfrac{\alpha_j : \beta_j}{\gamma_j} \;\middle|\; j = 1..k \right\}.$$

(For simplicity, we reuse the symbols j, k, m, n_j, α_j, etc. below.) We define $\mathcal{S}_m((D \cup D_m, W)) = (D', W')$ over $\mathcal{L}^*$, obtained by extending $\mathcal{L}$ to $\mathcal{L}^*$ with new predicates symbols $\mathsf{ok}/1$, $\mathsf{ko}/1$, $\mathsf{bl}/1$, $\mathsf{ap}/1$, and names $N \dot\cup \{m\}$, as follows

$$\begin{aligned} D' &= D \cup D_N \cup D_M \\ W' &= W \cup W_M \cup \{\mathrm{DCA}_N, \mathrm{UNA}_N\} \end{aligned}$$

where

$$\begin{aligned}
D_N &= \left\{ \tfrac{\alpha_j \wedge \mathsf{ok}(n_j) : \beta_j}{\gamma_j \wedge \mathsf{ap}(n_j)} \;\middle|\; j = 1..k \right\} && (1)\\
D_M &= \left\{ \tfrac{: \neg\mathsf{ko}(m)}{\mathsf{ok}(n_1) \wedge \cdots \wedge \mathsf{ok}(n_k)} \right\} && (2)\\
&\quad \cup \left\{ \tfrac{: \neg\alpha_j}{\mathsf{bl}(m)}, \tfrac{(\gamma_1 \wedge \cdots \wedge \gamma_k) \supset \neg\beta_j :}{\mathsf{bl}(m)} \;\middle|\; j = 1..k \right\} && (3)\\
W_M &= \{\forall x \in N. in(x, m) \equiv (x = n_1 \vee ... \vee x = n_k)\} && (4)\\
&\quad \cup \{\mathsf{bl}(m) \supset \mathsf{ko}(m)\} && (5)\\
&\quad \cup \left\{ \big(\forall x \in N.\, in(x, m) \supset \mathsf{ap}(x)\big) \supset \mathsf{ap}(m) \right\} && (6)
\end{aligned}$$

Clearly, D_N contains the images of the original rules in D_m. Each rule $\delta_j \in D_N$ is applicable, if $\mathsf{ok}(n_j)$ is derivable. In fact, we assert $\mathsf{ok}(n_j)$ for every $\delta_j \in D_m$, *unless* we cannot jointly apply all rules of D_m. That is, before activating the constituent rules, we have to make sure that none of them will be blocked. This is accomplished through the justification $\neg\mathsf{ko}(m)$ in (2) together with Axiom (5). We block Rule (2) (and with it the derivability of all $\mathsf{ok}(n_j)$) when we detect that one of $\delta_1, \ldots, \delta_k$ is blocked. That is, $\mathsf{ko}(m)$ will be an immediate consequence of $\mathsf{bl}(m)$.

Now, we have that D_m is blocked ($\mathsf{bl}(m)$) just if some rule in D_m is blocked. However, since we must control a whole set of defaults, we must check for the blockage

[2] We let $\dot\cup$ stand for disjoint union.

of one of the constituent default rules in the context of all other rules in the set applying. For detecting the failure of consistency, we verify for D_m and some set of formulas S (cf. Definition 1), whether $S \cup \{\gamma_1, \ldots, \gamma_k\} \vdash \neg\beta_j$ rather than $S \vdash \neg\beta_j$. This motivates the prerequisite of the second rule in (3). This context, $(\gamma_1 \wedge \cdots \wedge \gamma_k)$, is not needed for detecting the failure of derivability by means of the first rule in (3), since this test is effectuated with respect to the final extension E via $\neg\alpha_j \notin E$.

Finally, as given in (6), D_m is applied ($\mathsf{ap}(m)$) just if every rule in D_m is applied; it is only in this last case that the consequents of the constituent rules in D_m are asserted.

Consider theory $(D \cup D_m, W)$, where

$$D = \left\{ \frac{:E}{E} \right\},\ D_m = \left\{ n_1 : \frac{:P}{P},\ n_2 : \frac{:S}{S} \right\}. \tag{7}$$

For D_N and D_M, we obtain (after simplifying and removing redundant defaults):

$$\frac{\mathsf{ok}(n_1) : P}{P \wedge \mathsf{ap}(n_1)}, \quad \frac{\mathsf{ok}(n_2) : S}{S \wedge \mathsf{ap}(n_2)}, \quad \frac{: \neg\mathsf{ko}(m)}{\mathsf{ok}(n_1) \wedge \mathsf{ok}(n_2)}, \quad \frac{(\neg P \vee \neg S) :}{\mathsf{bl}(m)}.$$

The in predicate has instances: $in(n_1, m)$ and $in(n_2, m)$. From (6) we can deduce $[\mathsf{ap}(n_1) \wedge \mathsf{ap}(n_2)] \supset \mathsf{ap}(m)$.

Let $W = \{\neg(P \wedge E \wedge S)\}$. We obtain two extensions, one containing $P, S, \neg E$ and the other containing $E, \neg(P \wedge S)$. For the first case, we obtain $\mathsf{ok}(n_1)$ and $\mathsf{ok}(n_2)$. If both δ_1 and δ_2 are applicable (which they are) then we conclude $P \wedge \mathsf{ap}(n_1)$ and $S \wedge \mathsf{ap}(n_1)$ as well as $\mathsf{ap}(m)$. From this we get P and S and so $\neg E$. For the other extensions, if the default $\frac{:E}{E}$ is applied, then $\neg P \vee \neg S$ is derivable, and so $\frac{(\neg P \vee \neg S) :}{\mathsf{bl}(m)}$ is applicable, from which we obtain $\mathsf{bl}(m)$, and so $\mathsf{ko}(m)$, blocking application of $\frac{: \neg\mathsf{ko}(m)}{\mathsf{ok}(n_1) \wedge \mathsf{ok}(n_2)}$. Consequently neither $\frac{\mathsf{ok}(n_1) : P}{P \wedge \mathsf{ap}(n_1)}$ nor $\frac{\mathsf{ok}(n_2) : S}{S \wedge \mathsf{ap}(n_2)}$ can be applied.

In the next example, defaults inside a set depend upon each other. Consider $(\emptyset \cup D_m, \emptyset)$ with

$$D_m = \left\{ n_1 : \frac{:Q}{Q},\ n_2 : \frac{Q : R}{R} \right\}.$$

We get for D_N and D_M the following rules.

$$\frac{\mathsf{ok}(n_1) : Q}{Q \wedge \mathsf{ap}(n_1)}, \ \frac{Q \wedge \mathsf{ok}(n_2) : R}{R \wedge \mathsf{ap}(n_2)}, \qquad \frac{: \neg\mathsf{ko}(m)}{\mathsf{ok}(n_1) \wedge \mathsf{ok}(n_2)}, \ \frac{(\neg Q \vee \neg R) :}{\mathsf{bl}(m)}, \ \frac{: \neg Q}{\mathsf{bl}(m)}.$$

We obtain $\mathsf{ok}(n_1)$, and $\mathsf{ok}(n_2)$, which allow us to apply default δ_1, yielding in turn $Q \wedge \mathsf{ap}(n_1)$. Given Q, we can now apply default δ_2, yielding $R \wedge \mathsf{ap}(n_2)$. This allows us to deduce $\mathsf{ap}(m)$. We thus get an extension containing Q and R.

The last example also shows why we cannot avoid the translation by replacing D_m by $\frac{\bigwedge_{\delta \in D_m} PRE(\delta) : \bigwedge_{\delta \in D_m} JUS(\delta)}{\bigwedge_{\delta \in D_m} CON(\delta)}$. As well, in Section 4, this replacement would result in an exponential blowup in the encoding.

The next theorem summarizes properties of our approach, and shows that rules are applied either en masse, or not at all.

Theorem 1. *Let E be a consistent extension of $\mathcal{S}_m((D \cup D_m, W))$ for default theory $(D \cup D_m, W)$. We have that:*

1. $\mathsf{ap}(m) \in E$ *iff* $\{\mathsf{ap}(n_\delta) \mid \delta \in D_m\} \cup CON(D_m) \subseteq E$

2. $\mathsf{bl}(m) \in E$ *iff* $\{\mathsf{ap}(n_\delta) \mid \delta \in D_m\} \not\subseteq E$
3. $\mathsf{ok}(n_\delta) \in E$ *iff* $\mathsf{ap}(n_\delta) \in E$
4. $\mathsf{ok}(n_\delta) \in E$ *for all* $\delta \in D_m$ *iff* $\mathsf{ko}(m) \notin E$
5. $\mathsf{ap}(n_\delta) \in E$ *implies* $(\mathsf{ap}(m) \wedge in(n_\delta, m)) \in E$ *for some* $\delta \in D_m$
6. $\mathsf{ap}(n_\delta) \in E$ *for* $\delta \in D_m$ *iff* $\{\mathsf{ap}(n_\delta) \mid \delta \in D_m\} \subseteq E$.

Theorem 2. *For default theory* $(\emptyset \cup D, W)$*, we have that* $\mathcal{S}_m((\emptyset \cup D, W))$ *has extension* E *where either* $E \cap \mathcal{L} = \mathit{Th}(W \cup \mathit{CON}(D))$ *or else* $E \cap \mathcal{L} = \mathit{Th}(W)$.

The default theory $\left(\emptyset \cup \left\{\frac{: B}{\neg B}\right\},\ \emptyset\right)$ has an extension E where $E \cap \mathcal{L} = \mathit{Th}(\emptyset)$.

Theorem 3. *Let* (D, W) *be a (standard) default theory over* $\mathcal{L}$ *with extension* E *and (respective) set of generating defaults* $\mathit{GD}(D, E)$*. Then* $\mathcal{S}_m((\emptyset \cup \mathit{GD}(D, E), W))$ *has extension* E' *where* $E = E' \cap \mathcal{L}$.

4 Encoding extensions using sets

For encoding extensions of a normal default theory (D, W), we use the machinery developed in the previous section to determine maximal (with respect to set inclusion) sets of applicable defaults. Names are introduced for each subset of D, and for each instance of a rule in each subset of D. As well, new predicate symbols are introduced to further control application of sets of rules. We then give a translation that yields a second default theory (D', W'). Viewed algorithmically, this second theory carries out the following: If the original set of defaults D constitutes the set of generating defaults of an extension, then a corresponding "ap"-literal is derived; all default consequences are obtained; and all subsets of the defaults are rendered inapplicable. If this isn't the case (and D isn't a set of generating defaults), we proceed along the partial order induced by set inclusion and consider every set $D \setminus \{\delta\}$ for every $\delta \in D$ to see whether it is a set of generating defaults. Crucially, default conclusions are "tagged" with the name of the set in which they appear so as to eliminate possible side effects.

To name sets of defaults, we take some fixed enumeration $\langle n_1, \ldots, n_k\rangle$ of N, and define m as a k-ary function symbol. Then, for $n_\perp \notin N$, define

$$\begin{aligned} &\mathrm{DCA}_M : \forall x_1, \ldots, x_k.\ \textit{set-name}(m(x_1, \ldots, x_k)) \equiv \\ &\quad (x_1 = n_1 \vee x_1 = n_\perp) \wedge \cdots \wedge (x_k = n_k \vee x_k = n_\perp). \end{aligned}$$

Intuitively, $x_i = n_\perp$ tells us that n_i does not belong to the set at hand.

Accordingly, for $\boldsymbol{x} = x_1..x_k$ and $\boldsymbol{x}' = x'_1..x'_k$ define

$$\begin{aligned} &\mathrm{UNA}_M : \forall \boldsymbol{x}, \boldsymbol{x}'.\ \textit{set-name}(m(\boldsymbol{x})) = \\ &\quad \textit{set-name}(m(\boldsymbol{x}')) \equiv x_1 = x'_1 \wedge \cdots \wedge x_k = x'_k\ . \end{aligned}$$

The advantage of this "vector-oriented" representation over a dynamic one including a binary function symbol (as with lists) is that each set has a unique representation. We write $\forall x \in M.\ P(x)$ instead of $\forall x.\ \textit{set-name}(x) \supset P(x)$. Further, we use M for denoting the set of all valid set-names, that is,

$$M = \{m \mid \mathrm{DCA}_M \models \textit{set-name}(m)\}\ .$$

In order to ease notation, we write $m_{1,3}$ instead of $m(n_1, n_\perp, n_3, n_\perp, \ldots, n_\perp)$ when representing the set $\{\delta_1, \delta_3\}$. Also, we abbreviate $m(n_\perp, \ldots, n_\perp)$ by $m_\emptyset$ and $m(n_1, \ldots, n_k)$ by m_D. Note the difference between names n_i and m_i, induced by our notational convention.

We also rely on the "vector-oriented" representation for capturing set membership, denoted by $in/2$. Consider for instance $N = \{n_1, n_2\}$. Membership is then axiomatized through the formulas

$$\forall x_1, x_2.\ in(n_1, m(x_1, x_2)) \equiv (n_1 = x_1)$$
$$\forall x_1, x_2.\ in(n_2, m(x_1, x_2)) \equiv (n_2 = x_2).$$

While this validates $in(n_1, m_{1,2})$, it falsifies $in(n_1, m_2)$. See (15) for the general case.

We need to be able to refer to separate instances of the same default appearing in different sets. For this we introduce a function-symbol $\cdot/2$. For $\delta_j \in D_i$ we write $n_{\delta_j}\cdot m_i$ or $n_j\cdot m_i$ to name the instance of δ_j appearing in D_i. This results in name set $N\cdot M = \{n\cdot m \mid n \in N, m \in M\}$. Corresponding axioms, as DCA$_{N\cdot M}$ and UNA$_{N\cdot M}$, are obtained in a straightforward way. In what follows, we refer to the various domain closure and unique names axioms pertaining to N, M, and $N\cdot M$ as $Ax(N)$.[3]

Given language $\mathcal{L}$, we define a family of languages $\mathcal{L}(m)$ for $m \in M$ as follows. If P is an i-ary predicate symbol then $P(\cdot)$ is a distinct $(i+1)$-ary predicate symbol. If $\gamma \in \mathcal{L}$ then $\gamma(m) \in \mathcal{L}(m)$ is the formula obtained by replacing all predicate symbols in γ with predicate symbols extended as described, and with term m as the $(i+1)^{st}$ argument. This extra argument is used to index formulas by the (names of) sets in which they are used.

Lastly, we introduce special-purpose predicates for controlling the application of sets of defaults. These are summarised in the following table:

Name	Use/meaning
$m \sqsubset m'$	$D_m \subset D_{m'}$
$\mathsf{ok}(e)$	It is ok to try to apply set/rule e
$\mathsf{ap}(e)$	Set/rule e is applied
$\mathsf{bl}(m)$	Not all rules in set m can be applied
$\mathsf{ovr}(m)$	Some set named m' is applied and $m \sqsubset m'$
$\mathsf{ko}(m)$	For set m, $\mathsf{bl}(m) \vee \mathsf{ovr}(m)$ is true

Taking all this into account, we obtain the following translation, mapping default theories in language $\mathcal{L}$ onto default theories in the language $\mathcal{L}^+$ obtained by unioning all languages $\mathcal{L}(m)$ for $m \in M$ and using the aforementioned names and introduced predicates and functions:

Definition 2. *Given a finite default theory (D, W) over $\mathcal{L}$ and its set of associated default names N, define $\mathcal{E}((D, W)) = (D', W')$ over $\mathcal{L}^+$ by*

$$D' = D_N \cup D_M \cup D_\neg$$
$$W' = W_D \cup W_W \cup W_M \cup W_\sqsubset \cup Ax(N)$$

[3] Note that names in M and $N\cdot M$ are obtained from those in N.

where

$$D_N = \left\{ \frac{\alpha(x)\wedge in(n,x)\wedge \mathsf{ok}(n\cdot x) : \beta(x)}{\gamma(x)\wedge \mathsf{ap}(n\cdot x)} \;\middle|\; n : \frac{\alpha : \beta}{\gamma} \in D \right\} \tag{8}$$

$$D_M = \left\{ \frac{\mathsf{ok}(x) : \neg \mathsf{ko}(x)}{\forall y \in N.\ in(y,x) \supset \mathsf{ok}(y\cdot x)} \right\} \tag{9}$$

$$\cup \left\{ \frac{in(n,x)\wedge \mathsf{ok}(x) : \neg\alpha(x)}{\mathsf{bl}(x)} \;\middle|\; n : \frac{\alpha : \beta}{\gamma} \in D \right\} \tag{10}$$

$$\cup \left\{ \frac{([\forall y \in N.\ in(y,x) \supset c(y,x)] \supset \neg\beta(x))\wedge \mathsf{ok}(x) :}{\mathsf{bl}(x)} \;\middle|\; n : \frac{\alpha : \beta}{\gamma} \in D \right\} \tag{11}$$

$$D_\neg = \left\{ \frac{: \neg(x \sqsubset y)}{\neg(x \sqsubset y)}, \frac{: \neg in(x,y)}{\neg in(x,y)} \right\} \tag{12}$$

$$W_W = \{\forall x \in M.\ \alpha(x) \mid \alpha \in W\} \tag{13}$$

$$W_D = \{\forall x \in M.\, c(n_\delta, x) \equiv \mathit{CON}(\delta)(x) \mid \delta \in D\} \tag{14}$$

$$W_M = \{\forall x_1, \ldots, x_k.\ in(n_i, m(x_1, \ldots, x_k)) \equiv (n_i = x_i) \mid n_i \textit{ in } \langle n_1, \ldots, n_k\rangle\} \tag{15}$$

$$\cup \{\forall x, x' \in M.[\exists y \in N. \neg in(y,x) \wedge in(y,x')] \wedge [\forall y.\ in(y,x) \supset in(y,x')] \supset x \sqsubset x' \} \tag{16}$$

$$W_\sqsubset = \{\mathsf{ok}(m_D)\} \tag{17}$$

$$\cup \{\forall x \in M\big[\forall y \in M.\, x \sqsubset y \supset \mathsf{bl}(y)\big] \supset \mathsf{ok}(x)\} \tag{18}$$

$$\cup \{\forall x \in M.\ \big[\mathsf{bl}(x) \vee \mathsf{ovr}(x)\big] \supset \mathsf{ko}(x)\} \tag{19}$$

$$\cup \{\forall x \in M\ \big[\forall y \in N.\, in(y,x) \supset \mathsf{ap}(y\cdot x)\big] \supset \mathsf{ap}(x)\} \tag{20}$$

$$\cup \{\forall x, x' \in M.\, \mathsf{ap}(x) \supset (x' \sqsubset x \supset \mathsf{ovr}(x'))\} \tag{21}$$

The rules in D_N and D_M directly generalise those in (1–3), from treating a single set named m to an arbitrary set referenced by variable x. The specific consequents used in the second rule in (3) are dealt with via the axioms in (W_D/14) that allows us to quantify over default consequents (via predicate c). This trick avoids the exponential blowup that would occur in (11) if we were to explicitly give the consequences of the rules.

The rules in ($D_\neg$/12) provide us with complete knowledge on predicates $\sqsubset$ and in. The axioms in $(W_W/13)$ propagate the information in W to all possible contexts.

W_M takes care of what we need wrt set operations. That is, (15) formalises set membership, while (16) formalises strict set inclusion. $W_\sqsubset$ axiomatises the control flow along the partial order induced by $\sqsubset$. Axioms (17) and (18) tell us when it is ok to consider a certain set: we always consider the maximum set D; otherwise, via (18), we consider a set just when every superset is known to be blocked (and so inapplicable). (19) tells us when the consideration of a set is cancelled. This either happens because a set is inapplicable (given by bl) or because it has been explicitly cancelled (given by ovr). (20) asserts that a set is applied just if all of its member rules are. Once we have found an applicable set of rules (and hence a set of generating defaults) we need not consider any subset; (21) annuls the consideration of all such subsets.

For example, consider the following normal default theory:

$$\mathit{\Delta}_{22} = \left(\left\{n_1 : \tfrac{:A}{A}, n_2 : \tfrac{:B}{B}, n_3 : \tfrac{:\neg B}{\neg B}, n_4 : \tfrac{B:D}{D}\right\}, \emptyset\right). \tag{22}$$

From $\mathcal{E}(\mathit{\Delta}_{22})$ we get an extension, where the only "ap-literals" are $\mathsf{ap}(m_{1,2,4})$ and $\mathsf{ap}(m_{1,3})$. That is, $\mathit{\Delta}_{22}$ has two extensions with generating defaults, the first with δ_1, δ_2, δ_4, and the second with δ_1, δ_3. Among formulas in the extension of $\mathcal{E}(\mathit{\Delta}_{22})$ are $A(m_{1,2,4})$, $A(m_{1,3})$, $B(m_{1,2,4})$, $\neg B(m_{1,3})$, and $D(m_{1,2,4})$. To see this, let us take a closer look at the image of $\mathit{\Delta}_{22}$, namely $\mathcal{E}(\mathit{\Delta}_{22})$. For D_N, we get

$$\frac{in(n_1,x)\wedge\mathsf{ok}(n_1\cdot x) : A(x)}{A(x)\wedge\mathsf{ap}(n_1\cdot x)} \qquad \frac{in(n_2,x)\wedge\mathsf{ok}(n_2\cdot x) : B(x)}{B(x)\wedge\mathsf{ap}(n_2\cdot x)} \tag{23}$$

$$\frac{in(n_3,x)\wedge\mathsf{ok}(n_3\cdot x) : \neg B(x)}{\neg B(x)\wedge\mathsf{ap}(n_3\cdot x)} \qquad \frac{B(x)\wedge in(n_4,x)\wedge\mathsf{ok}(n_4\cdot x) : D(x)}{D(x)\wedge\mathsf{ap}(n_4\cdot x)} \tag{24}$$

We get a single nontrivial rule in (10), namely

$$\frac{in(n_4,x)\wedge\mathsf{ok}(x) : \neg B(x)}{\mathsf{bl}(x)} \tag{25}$$

and four rules in (11)

$$\frac{([\forall y\in N.\ in(y,x)\supset c(y,x)]\supset\neg A(x))\wedge\mathsf{ok}(x) :}{\mathsf{bl}(x)} \tag{26}$$

$$\frac{([\forall y\in N.\ in(y,x)\supset c(y,x)]\supset\neg B(x))\wedge\mathsf{ok}(x) :}{\mathsf{bl}(x)} \tag{27}$$

$$\frac{([\forall y\in N.\ in(y,x)\supset c(y,x)]\supset\ \ B(x))\wedge\mathsf{ok}(x) :}{\mathsf{bl}(x)} \tag{28}$$

$$\frac{([\forall y\in N.\ in(y,x)\supset c(y,x)]\supset\neg D(x))\wedge\mathsf{ok}(x) :}{\mathsf{bl}(x)} \tag{29}$$

Given $\mathsf{ok}(m_D)$, we may consider any rule in D_M. However, given that $\forall y \in N.\ in(y, m_D)$ is true, we obtain that (14) and $\forall y \in N.\ in(y, m_D) \supset c(y, m_D)$ are inconsistent and thus imply any formula. Consequently, rules (26) to (29) are applicable and provide $\mathsf{bl}(m_D)$, yielding $\mathsf{ko}(m_D)$, which in turn blocks (9) for $x = m_D$. From (16), we obtain (among other relations) $m_{1,2,3} \sqsubset m_D$, $m_{1,2,4} \sqsubset m_D$, $m_{1,3,4} \sqsubset m_D$, and $m_{2,3,4} \sqsubset m_D$. From (18), we then get $\mathsf{ok}(m_{1,2,3})$, $\mathsf{ok}(m_{1,2,4})$, $\mathsf{ok}(m_{1,3,4})$, and $\mathsf{ok}(m_{2,3,4})$.

Now, consider $\mathsf{ok}(m_{1,2,4})$. From (9), we obtain

$$\forall y \in N.\ in(y, m_{1,2,4}) \supset \mathsf{ok}(y\cdot m_{1,2,4})$$

yielding $\mathsf{ok}(n_1\cdot m_{1,2,4})$, $\mathsf{ok}(n_2\cdot m_{1,2,4})$, and $\mathsf{ok}(n_4\cdot m_{1,2,4})$. This allows us to apply three of the four rules in (23/24) and we obtain $A(m_{1,2,4}) \wedge \mathsf{ap}(n_1\cdot m_{1,2,4})$, $B(m_{1,2,4}) \wedge \mathsf{ap}(n_2\cdot m_{1,2,4})$, and $D(m_{1,2,4}) \wedge \mathsf{ap}(n_4\cdot m_{1,2,4})$. From (20), we obtain $\mathsf{ap}(m_{1,2,4})$, from which we deduce with (21) in turn $\mathsf{ovr}(m_{1,2,4})$, $\mathsf{ovr}(m_{2,4})$, ..., $\mathsf{ovr}(m_4)$, and $\mathsf{ovr}(m_\emptyset)$.

Next, consider $\mathsf{ok}(m_{1,2,3})$. As with $\mathsf{ok}(m_D)$, we obtain an inconsistency among $in(n_1, m_{1,2,3})$, $in(n_2, m_{1,2,3})$, $in(n_3, m_{1,2,3})$, $\forall y \in N.\ in(y, m_{1,2,3}) \supset c(y, m_{1,2,3})$, and (14). This validates the prerequisites of Rule (26), (27), and (28), thus yielding

$\mathsf{bl}(m_{1,2,3})$. As above, we then get from W_M that $\mathsf{ok}(m_{1,2})$, $\mathsf{ok}(m_{1,3})$, $\mathsf{ok}(m_{2,3})$. Note that we have already obtained $\mathsf{ovr}(m_{1,2})$ from $\mathsf{ap}(m_{1,2,4})$.

Given $\mathsf{ok}(m_{1,3})$, (9) provides us with $\mathsf{ok}(n_1 \cdot m_{1,3})$ and $\mathsf{ok}(n_3 \cdot m_{1,3})$. Using the two first rules in (23/24), we get $A(m_{1,3}) \wedge \mathsf{ap}(n_1 \cdot m_{1,3})$ and $\neg B(m_{1,3}) \wedge \mathsf{ap}(n_3 \cdot m_{1,3})$. From (20), we then get $\mathsf{ap}(m_{1,3})$, from which we deduce with (21) in turn $\mathsf{ovr}(m_1)$, $\mathsf{ovr}(m_3)$, and $\mathsf{ovr}(m_\emptyset)$ (again).

Given $\mathsf{ok}(m_{2,3})$, along with the fact that $in(n_2, m_{2,3})$, $in(n_3, m_{2,3})$, $\forall y \in N.\ in(y, m_{2,3}) \supset c(y, m_{2,3})$, and (14) imply $B(m_{2,3})$ and $\neg B(m_{2,3})$, Rule (27) and (28) fire and we get $\mathsf{bl}(m_{2,3})$.

The next results show that our default theories resulting from $\mathcal{E}$ have appropriate properties.

Theorem 4. *Let E be a consistent extension of $\mathcal{E}((D,W))$ for normal default theory (D,W). We have for all $\delta \in D$ and for all $D_m, D_{m'} \subseteq D$ that:*

1. $(m \sqsubset m') \in E$ *iff* $\neg(m \sqsubset m') \notin E$
2. $in(n_\delta, m) \in E$ *iff* $\neg in(n_\delta, m) \notin E$
3. $\mathsf{ok}(m) \in E$ *if* $\mathsf{ovr}(m) \notin E$
4. $\mathsf{ok}(m) \in E$ *if (*$\mathsf{ap}(m) \in E$ *or* $\mathsf{bl}(m) \in E$*)*
5. $\mathsf{ap}(m) \in E$ *iff* $\mathsf{ko}(m) \notin E$
6. $\mathsf{ko}(m) \in E$ *iff (*$\mathsf{bl}(m) \in E$ *or* $\mathsf{ovr}(m) \in E$*)*
7. $\mathsf{ovr}(m) \in E$ *iff* $\mathsf{ap}(m') \in E$ *and* $m \sqsubset m' \in E$ *for some* $m' \in M$.
8. *If* $\mathsf{ap}(m) \in E$ *then* $\mathsf{bl}(m') \in E$ *for all* $m' \in M$ *with* $m \sqsubset m' \in E$.
9. *If* $\mathsf{ap}(m) \in E$ *then* $\mathsf{ovr}(m') \in E$ *for all* $m' \in M$ *with* $m' \sqsubset m \in E$.
10. *If* $\mathsf{ap}(m), \mathsf{ap}(m') \in E$ *for then* $\neg(m \sqsubset m') \in E$

Theorem 5. *If (D,W) is a normal default theory then $\mathcal{E}((D,W))$ has a unique extension.*

The next two theorems show that our translation captures an encoding of extensions of a normal default theory.

Theorem 6. *Let (D,W) be a normal default theory and let E be the extension of $\mathcal{E}((D,W))$.*

Then for any $\mathsf{ap}(m) \in E$ with $m \in M$, we have that $Th(\{\gamma \mid \gamma(m) \in E\})$ is an extension of (D,W).

Theorem 7. *Let (D,W) be a normal default theory with extensions $E_1, ..., E_n$ and E be the extension of $\mathcal{E}((D,W))$.*

Then, for any $i \in \{1, \ldots, n\}$, there is some $m \in M$ naming $GD(D, E_i)$ such that $\mathsf{ap}(m) \in E$.

Lastly, our claim that a translated theory is "almost" a constant factor larger than the original requires elaboration. UNA$_N$ yields a quadratic number of unique names assertions. In practice this is no problem, since any sensible implementation would not explicitly list such axioms. With the exception of unique names assertions, a translated theory is a constant factor larger than the original. To see this, it suffices to examine Definition 2. Each of (8, 10, 11, 14, 15) introduce $|D|$ axioms/rules; (13) introduces $|W|$ axioms. All remaining terms introduce a single axiom. Moreover, the size of individual axioms is similarly bounded. (For example, each instance of (8) is a constant factor larger than the original default.)

5 Discussion

We have shown how we can encode a normal default theory so that the extension from the encoding represents all extensions of the original theory. These results don't rely on the normal form of the defaults, but rather on the fact that normal default theories are *semi-monotonic*, that is on the fact that if E is an extension of (D, W), then there is an extension $E' \supseteq E$ of $(D \cup D', W)$. The results of the previous sections then extend to any such theory.

The fact that we encode all extensions of a theory within a single extension means that we can now encode phenomena of interest, usually dealt with at the metalevel, at the object level. Specifically we can now encode the notions of skeptical and credulous inference within a theory. In order to do this, we introduce two new constants $skep$ and $cred$, for "skeptical" and "credulous" respectively.

A formula is a skeptical inference if it is a member of every extension. In our approach, this means that it follows in every "ap-set". Hence we define skeptical inference within a theory, for a given formula γ, by

$$(\forall x \in M.\ \mathsf{ap}(x) \supset \gamma(x)) \supset \gamma(skep).$$

For credulous inference there are a number of possibilities. The simplest is to assert that a formula is a credulous inference if it is a member of some extension:

$$(\exists x \in M.\ \mathsf{ap}(x) \wedge \gamma(x)) \supset \gamma(cred).$$

With this definition, a formula and its negation may be credulous inferences. A stronger definition is to assert that a formula is a credulous inference if it is a member of some extension, and its negation is a member of no extension. We can define this notion of credulous inference (indicated by $cred'$) for a formula γ by means of the default:

$$\frac{\exists x \in M.\ \mathsf{ap}(x) \wedge \gamma(x)\ :\ \forall x \in M.\ \mathsf{ap}(x) \supset \gamma(x)}{\gamma(cred')}.$$

Hence in Example (22), we obtain that A is a $skep$tical inference, while D is a $cred'$ulous inference. B and $\neg B$ are $cred$ulous inferences.

We have suggested that the approach may be applicable in diagnosis programs, such as found in [10]. Similarly, the approach can be used to directly encode applications expressible in Theorist [8]. That is, there is a correspondence between so-called *Poole-type* theories and Theorist with constraints [3]. Since Poole-type theories are semi-monotonic, this means that our approach can encode any application encodable in Theorist.

Our approach relies on a first-order language. Despite this, the image of a theory over a finite language remains finite. As regards implementation, however, it is not advisable to use a bottom-up grounding approach, as done in many implementations of extended logic programming [4, 7]. Instead, a query-oriented approach seems to be advantageous, because it may rely on unification rather than ground instantiation.

In Definition 2, sets of defaults were ordered based on the partial order given by set containment. This order represents one example of a *preference* order on sets of defaults. A natural avenue for future work would be to generalise our approach to address arbitrary preference orders on sets of defaults. In an arbitrary preference order on sets, one could represent desiderata as found in configuration, scheduling, or (generally) decision-theoretic problems. This could also be combined with the present approach yielding an encoding of preferences on extensions. Hence, for our diagnosis example, we might want to prefer extensions (diagnoses) on the basis of an ordering based on reliability of components.

6 Conclusion

We have described an approach for encoding default extensions within a single extension. Using constants and functions for naming, we can refer to default rules, sets of defaults, and instances of a rule in a set. Via these names we can, first, determine whether a set of defaults is its own set of generating defaults and, second, consider the application of sets of defaults ordered by set containment. The translated theory requires a modest increase in space: except for unique names axioms, only a contant-factor increase is needed. The translated theory is a (regular, Reiter) default theory. Hence we essentially axiomatise the notion of "extensions" for a class of default theories in a single extension. Further, we are able to prove that our translation behaves correctly.

Using the approach we can now express notions such as skeptical and credulous inference within a theory. Arguably this will prove beneficial in expressing at the object level problems and approaches generally expressed at the metalevel. Areas of application range from specific areas such as diagnosis, to broadly-applicable approaches such as Theorist. Lastly, we suggest that the approach may be easily extended to address arbitrary preferences over sets of defaults.

References

1. G. Brewka. Reasoning about priorities in default logic. In *Proceedings AAAI'94*, pages 940–945. The AAAI Press, 1994.
2. J. Delgrande and T. Schaub. Expressing preferences in default logic. *Artificial Intelligence*, 123(1-2):41–87, 2000.
3. J. Dix. On cumulativity in default logic and its relation to Poole's approach. In B. Neumann, editor, *Proceedings ECAI'92*, pages 289–293. Wiley, 1992.
4. 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 LPNMR'97*, pages 363–374. Springer, 1997.
5. J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.
6. R. Moore. Semantical considerations on nonmonotonic logics. *Artificial Intelligence*, 25:75–94, 1985.
7. 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 LPNMR'97*, pages 420–429. Springer, 1997.
8. D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.
9. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.
10. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.