How to Prefer More Specific Defaults in Terminological Default Logic*

Franz Baader and Bernhard Hollunder
German Research Center for AI (DFKI)
Stuhlsatzenhausweg 3, 6600 Saarbriicken 11, Germany
E-mail: *(last* na/ne)@dfki.uni-sb.de

Abstract

In a recent paper we have proposed terminological default logic as a formalism which combines both means for structured representation of classes and objects, and for default inheritance of properties. The major drawback which terminological default logic inherits from general default logic is that it does not take precedence of more specific defaults over more general ones into account. The present paper addresses the problem of modifying terminological default logic such that more specific defaults are preferred. It turns out that the existing approaches for expressing priorities between defaults do not seem to be appropriate for this purpose. Therefore we shall consider an alternative approach for dealing with prioritization in the framework of Heifer's default logic. The formalism is presented in the general setting of default logic where priorities are given by an arbitrary partial ordering on the defaults. We shall exhibit some interesting properties of the new formalism, compare it with existing approaches, and describe an algorithm for computing extensions.

1 Introduction

Early knowledge representation formalisms such as semantic networks and frames comprise both means for structured representation of classes and objects, and for default inheritance of properties. However, these formalisms did not have a well-defined formal semantics, and subsequent formalisms trying to overcome this problem usually concentrated on one of these two means of representation. Nonmonotonic inheritance networks are concerned with defeasible inheritance, sometimes in combination with strict inheritance, but the nodes in these networks are unstructured objects or classes.[1]

Terminological representation formalisms, on the other hand, can be used to define the relevant concepts of a problem domain in a structured and well-formed way. This is done by building complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates) with the help of operations provided by the concept language of the particular formalism. In addition, objects can be described with respect to their relation to concepts and their interrelation with each other. The concept descriptions are interpreted as universal statements, which means that they do not allow for exceptions. As a consequence, the terminological system can use descriptions to automatically insert concepts at the proper place in the concept hierarchy (classification), and it can use the facts stated about objects to deduce to which concepts they must belong, but objects cannot inherit properties by default.

The problem addressed in this paper is how to bring together both means of representation originally present in semantic networks and frames, without losing the advantages of terminological formalisms, such as being equipped with a formal and well-understood semantics and providing for automatic concept classification. An integration of defaults would often greatly enhance applicability of terminological systems, or would at least make their use more convenient in most applications (see, e.g., [15] which shows that embedding defaults into terminological systems is an important item on the wish list of users of such systems). For this reason, several existing terminological systems, such as BACK [13], CLASSIC [4j, K-Rep [II], or LOOM [12], have been or will be extended to provide the user with some kind of default reasoning facilities. As the designers of these systems themselves point out, however, these approaches usually have an ad hoc character, and thus do not satisfy the requirement of having a formal semantics.

As a first attempt to give a formally well-founded solution to this problem, an integration of Reiter's default logic into a terminological formalism was proposed in [2j. One reason for selecting default logic, out of the wide range of nonmonotonic formalisms, was that Reiter's default rule approach fits well into the philosophy of terminological systems. Most of these systems already provide their users with a form of "monotonic" forward

*This work has been supported by the German Ministry for Research and Technology (BMFT) under research contract IT W 9201.

[1] There are some attempts to generalize this approach to structured classes, but they work in a very restricted setting, and it is not clear how to obtain more general results in this direction (see, e.g., [14]).

rules, and it turned out that these rules can be viewed as specific default rules where the justifications are absent. A second pleasant feature of terminological default logic, as introduced in [2], is that it becomes decidable provided that applicability of default rules is restricted to objects explicitly present in the knowledge base. It should be noted that this constraint is also imposed on the monotonic rules in terminological systems.

The major drawback which terminological default logic inherits from general default logic is that it does not take precedence of more specific defaults over more general ones into account. For example, assume that we have a default which says that penguins cannot fly, [2] and another one which says that birds can fly, and that classification shows that penguins are a subconcept of birds. Intuitively, for any penguin the more specific first default should be preferred, which means that there should be only one default extension in which the penguin cannot fly. However, in default logic the first default has no priority over the second one, which means that one also gets a second extension where the penguin can fly. This behaviour has already been criticized in the general context of default logic, but it is all the more problematic in the terminological case where the emphasis lies on the hierarchical organization of concepts.

In the present paper we shall consider the problem of modifying terminological default logic such that more specific defaults are preferred. After a short recapitulation of default logic and its specialization, terminological default logic, in Section 2, we shall consider the existing approaches for expressing priorities between defaults, and shall point out why they do not seem to be appropriate for our purpose (see Section 3). For this reason we present in Section 4 an alternative approach for dealing with prioritization in the framework of Reiter's default logic. The formalism is presented in the general setting of default logic where priorities are given by an arbitrary partial ordering on the defaults. For terminological default theories the priorities between defaults will be induced by the position of their prerequisites in the concept hierarchy. We shall exhibit some interesting properties of the new formalism, and shall compare it with existing approaches. It turns out that every extension according to our definition (S-extension) is an extension according to Reiter's definition (R-extension); however, R-extensions which are not compatible with the partial ordering on defaults are excluded by our formalism. Not all default theories with an R extension have an S-extension, but every normal default theory has an S-extension. If the defaults are further restricted to prerequisite-free normal defaults then our approach coincides with the one of Brewka and Junker [5; 9]. In Section 5 the problem of how to compute S-extensions will be addressed.

2 Default Logic

This section briefly reviews Reiter's default logic and its specialization, terminological default logic.

[2]The reader who is surprised that this is only taken as a default property should have a look at the cover of [8].

Reiter's default logic Reiter [17] deals with the problem of how to formalize nonmonotonic reasoning by introducing nonstandard, nonmonotonic inference rules, which he calls default rules. A default rule is any expression of the form

$$\frac{\alpha : \beta}{\gamma},$$

where α, β, γ are first-order formulae.[3] Here a is called the prerequisite of the rule, β is its justification, and γ its consequent. For a set of default rules V, we denote the sets of formulae occurring as prerequisites, justifications, and consequents in V by Pre(V), Jus(D), and Con(D), respectively.

A default rule is closed iff α, β, γ do not contain free variables. It is semi-normal iff its justification implies the consequent, and it is normal if its justification and consequent are identical. A default theory is a pair (W,V) where W is a set of closed first-order formulae (the world description) and V is a set of default rules. A default theory is close.d iff all its default rules are closed.

Intuitively, a closed default rule can be applied, i.e., its consequent is added to the current set of beliefs, if its prerequisite is already believed and its justification is consistent with the set of beliefs. Formally, the consequences of a closed default theory are defined with reference to the notion of an extension (called R-cxtenswn in this paper), which is a set of deductively closed first-order formulae defined by a fixed point construction (see [17], p.89). In general, a closed default theory may have more than one R-extension, or even no extension. Depending on whether one wants to employ skeptical or credulous reasoning, a closed formula 6 is a consequence of a closed default theory iff it is in all R-extensions or if it is in at least one R-extension of the theory.

To generalize the notion of an R-extension to arbitrary default theories one just assumes that a default with free variables stands for all its ground instances. In Roller's original semantics the world description and the consequents of all defaults have to be Skolernized before building ground instances (over the enlarged signature). As shown in [2] Skolemization leads to both semantic and algorithmic problems, which is the reason why we shall dispense with it in the case of terminological default theories.

Terminological default logic For lack of space we shall not formally introduce a particular terminological language (see e.g. [2] for details). Instead we shall just mention the features of terminological languages which will be important for the following. The terminological part of such languages allows one to build complex concept descriptions out of atomic concepts (unary predicates) and roles (binary predicates). For our purposes it suffices to know that a concept description C can be regarded as a first-order formula C{x) with one free variable x. The subsumption hierarchy between concepts

[3] For the sake of simplicity we consider only defaults with one justification. However, our results can easily be extended to the general case of defaults with finitely many justifications.

corresponds to implication of formulae: C is subsumed by D iff $\forall x: C(x) \rightarrow D(x)$ is valid.

The assertioal part of the language can be used to state that an object is an instance of a concept (7, or that two individuals are connected by a role R. Logically, this means that one has constant symbols a, b as names for objects, and can build formulae C(a) and R(a,b) by respectively substituting a for the free variable in C(r) and applying the binary predicate R to the constants a, 6. A finite set of such formulae is called an A Box. Important inference problems for ABoxes are whether a given A Box is consistent, and whether an object a is an instance of a concept C, i.e., whether C(a) is a logical consequence of the given ABox. It should be noted that the formulae C(x) obtained as concept descriptions of a terminological language belong to a restricted subclass of all first-order formulae with one free variable. For this reason the subsumption, consistency and instantiation problems are usually decidable for these languages.

A terminological default theory is a pair (A, D) where A is an ABox and V is a finite set of default rules whose prerequisites, justifications, and consequents are concept descriptions. Obviously, since ABoxes can be seen as sets of closed formulae, and since concept descriptions can be seen as formulae with one free variable,[4] terminological default theories are subsumed by Reiter's notion of an open default theory. However, as motivated in Section 3 and 4 of [2], we do not Skolemize before building ground instances. This means that an open default of a terminological default theory is interpreted as representing all closed defaults which can be obtained by instantiating the free variable by all object names occurring in the ABox. With this interpretation, it is possible to compute all R-extensions of terminological default theories (see [2], Section 5 and 6).

3 Approaches to Prioritization

When conflicts occur in reasoning with defaults it is quite obvious that the more specific information should prevail over the more general one. In the context of terminological default theories this means that for an instance of the concepts C and D a default with prerequisite C should be preferred if C is subsumed by D. As mentioned in the introduction this requirement is not taken into account by Reiter's approach. If we assume that P, B, and F are concept descriptions defining penguins, birds, and flying objects, where P is subsumed by B, then the terminological default theory consisting of the world description {/'(Danny)) and the defaults

$$\frac{P(x) : \neg F(x)}{\neg F(x)} \quad \text{and} \quad \frac{B(x) : F(x)}{F(x)}$$

has two R-extensions. One of them contains F{Danny) and the other one $\neg F$(Danny), and the semantics gives no reason for preferring the second one, in which the more specific default was applied.

[4]The formulae occurring in one rule are assumed to have identical free variables.

To overcome this kind of problem several approaches for realizing priorities among defaults have been proposed in the literature. The priorities may be induced by specificity of prerequisites (as described above), but may also come from other sources (such as reliability of defaults).

Reiter and Criscuolo show how some kind of prioritization between defaults can be achieved without changing the formalism by encoding the priority information into the justifications of semi-normal defaults [18]. Although our simple example from above could be handled with this approach, it is not clear how to treat more complex situations. Reiter and Criscuolo do not describe a general method for solving these problems; they just "focus on certain fairly simple patterns of default rules." Another problem is that, even if one starts with normal defaults (as in our example), one ends up with semi-normal defaults when realizing priorities this way. But this means that one has to face the undesirable properties of non-normal defaults, such as non-existence of extensions.

In order to avoid the introduction of semi-normal defaults Brewka [6] takes the ideas underlying prioritized circumscription [10]] and defines an iterated version of default logic, which he calls prioritized default logic. As pointed out by Brewka himself, this approach makes sense only if it is restricted to prerequisite-free normal defaults. In this restricted case, prioritized default logic yields a prioritized version of Poole's approach to default reasoning [16], and it seems to exhibit a quite reasonable behaviour. One reason why this is nevertheless not an appropriate formalism for treating specificity in terminological default theories is that the defaults have to be put into levels of priorities which are totally ordered. However, subsumption between prerequisites only gives us a partial ordering on defaults.

In [5; 9] prioritized prerequisite-free normal default theories are generalized to ordered default theories which allow for an arbitrary partial ordering on defaults. We shall describe this approach in more detail because—in combination with an approach for approximating defaults with prerequisites by prerequisite-free normal defaults—it yields a first solution to our problem of treating specificity in terminological default theories, even though we shall argue that it still exhibits some undesirable properties. In addition, the default theories with specificity we shall propose in the next section turn out to be a generalization of ordered default theories to defaults with prerequisites.

An *ordered default theory* is a triple $(\mathcal{W}, \mathcal{D}, <)$, where $\mathcal{W}$ is a set of closed first-order formulae, $\mathcal{D}$ is a set of closed prerequisite-free normal defaults, and $<$ is a strict partial ordering on $\mathcal{D}$ such that $\{d' \in \mathcal{D} \mid d' < d\}$ is finite for every $d \in \mathcal{D}$.

The principal idea is to consider total extensions of the partial ordering when computing extensions of the ordered default theory (which we shall call B-extensions in the following). Any enumeration $d_1, d_2, \ldots$ of $\mathcal{D}$ that is compatible with the partial ordering (i.e., $i < k$ if $d_i < d_k$) defines a *B-extension* as follows. One starts with $\mathcal{W}$, and in the i-th step of the iteration, the consequent β_i of the default $d_i = \ :\beta_i/\beta_i$ is added if β_i is consistent with the set of formulae obtained after step 7—1. Otherwise, the current set of formulae remains unchanged. The limit of this process is the extension.

Even though ordered default theories allow for priorities given by a partial ordering, this approach cannot directly be used to realize specificity in terminological default theories. The reason is that the restriction to prerequisite-free defaults is too severe. In fact, for terminological default theories the priorities we wanted to consider were induced by subsumption relationships between the concept descriptions in the prerequisites. But this means that for prerequisite-free terminological defaults we no longer have a need for prioritization.

The situation is, however, not as bad as it seems. As shown in [3; 7], the closed normal default $\alpha : \beta/\beta$ can be approximated by the closed prerequisite-free normal default $\alpha \rightarrow \beta/\alpha \rightarrow \beta$. Thus one could start with a normal terminological default theory, determine the priorities between defaults from their prerequisites, and then transform the defaults into the corresponding ones without prerequisites. This way one ends up with an ordered default theory which approximates the terminological default theory, and which handles priorities induced by specificity of prerequisites in the terminological default theory.

However, we claim that this approach is still not satisfactory because it gives us a lot more than we bargained for. As pointed out in [7], the approximation not only gets rid of prerequisites, but also equips the defaults with properties of classical implication, such as reasoning by cases and reasoning using contrapositives of the original defaults. For example, assume that, in addition to the concept descriptions for penguins, birds, and flying objects, we have a description *W* for objects having wings, and that the only subsumption relation is the one between penguins and birds. If we consider the terminological default theory consisting of the world description *{P(Danny)}* and the defaults

$$\frac{P(x) : \neg F(x)}{\neg F(x)}, \quad \frac{B(x) : W(x)}{W(x)}, \quad \frac{W(x) : F(x)}{F(x)},$$

then the preferred extension should be the one in which Danny has wings, but does not fly. The approach we have described yields this extension; but it also yields another one in which Danny does not have wings, because as soon as the (approximation of the) first default has fired, the contrapositive of the third one can be fired, which gives us *-W(Danny)*

This shows that in this approach the defaults no longer behave like simple forward rules. But the similarity of default rules with the monotonic forward rules of terminological systems was one of our reasons for choosing default logic in the first place.

4 Default Theories with Specificity

To overcome the problems pointed out in the previous section we shall now propose a new approach for handling priorities among defaults with prerequisites. The semantics will be very close to Reiter's semantics, and the properties of our theory will also resemble those of Reiter's theory.

A ***default theory with specificity*** is a triple $(\mathcal{W}, \mathcal{D}, <)$ consisting of a closed default theory $(\mathcal{W}, \mathcal{D})$ and a strict partial ordering $<$ on $\mathcal{D}$ such that $\{d' \in \mathcal{D} \mid d' < d\}$ is finite for every $d \in \mathcal{D}$.

In the terminological case, $\mathcal{W}$ is an ABox, and $\mathcal{D}$ is obtained by instantiating the terminological default rules by all constants occurring in the ABox. For two instantiated terminological default rules d_1, d_2 with prerequisites $C_1(a_1), C_2(a_2)$ we have $d_1 < d_2$ iff they are concerned with the same object (i.e., $a_1 = a_2$) and C_1 is more specific than C_2 (i.e., C_1 is subsumed by C_2 but not vice versa). The restriction on the ordering is satisfied since $\mathcal{D}$ is finite by definition of terminological default theories.

Our definition of an extension for a default theory with specificity is modelled on Reiter's iterative characterization of R-extensions (see [17], Theorem 2.1). The main idea for treating priorities is that the consequent of a default can only be added during an iteration step if the default is not blocked by a preferred default, i.e., there does not exist a smaller default that is currently active. For a set E of closed formulae, and a closed default $d = \alpha : \beta/\gamma$ we say that *d is active in E* iff its prerequisite is a consequence of E (i.e., $\alpha \in Th(E)$[5]), its justification is consistent with E (i.e., $\neg\beta \notin Th(E)$), and its consequent is not a consequence of E (i.e., $\gamma \notin Th(E)$).

Definition 4.1 *Let $(\mathcal{W}, \mathcal{D}, <)$ be a default theory with specificity, and let $\mathcal{E}$ be a set of closed formulae. We define $E_0 := \mathcal{W}$, and for all $i \geq 0$*

$$E_{i+1} := E_i \cup \{\gamma \mid \exists d \in \mathcal{D}\colon d = \alpha : \beta/\gamma,\ \alpha \in Th(E_i),\ \neg\beta \notin \mathcal{E},\ \text{and all } d' < d \text{ are not active in } E_i\}.$$

Then $\mathcal{E}$ is an S-extension of iff $\mathcal{E} = \bigcup_{i \geq 0} Th(E_i)$.

The only difference to Reiter's characterization is the additional requirement that smaller defaults must not be active in the current state of the iteration. With this definition of an extension we get the intuitively correct result in our example with the three defaults concerning penguins, birds, and objects with wings. In fact, for any penguin the second default (asserting that birds normally have wings) can only fire after the more specific default (asserting that penguins normally cannot fly) has been applied. But this means that the third default (asserting that winged objects normally can fly) will never become applicable for a penguin (before its prerequisite becomes derivable, the negation of its justification must have been added). This means that our definition of an S-extension chooses from the two existing R-extensions the one which respects priorities.

Our first theorem states that this will always be the case, i.e., that the set of all S-extensions is always a subset of the set of all R-extensions.

Theorem 4.2 *Let $\mathcal{E}$ be an S-extension of the default theory with specificity $(\mathcal{W}, \mathcal{D}, <)$. Then $\mathcal{E}$ is an R-extension of $(\mathcal{W}, \mathcal{D})$.*

Proof idea. Even though the definition of an S-extension closely resembles Reiter's characterization of

[5] By $Th(E)$ we denote the deductive closure of the set of formulae E.

an R-extension, this theorem is not at all obvious. The idea for a proof is to take an S-extension $\mathcal{E}$ which has been obtained from the sequence $E_0, E_1, \ldots$, and to use it to construct a sequence $F_0, F_1, \ldots$ as in the characterization of R-extensions. It is easy to see that $E_i \subseteq F_i$ for all $i \geq 0$, but the converse is not true. In fact, the consequent γ of a default d may be added to F_i but not to E_i because d is blocked by a smaller default which is active. A straightforward way to prove that $F_i \subseteq \mathcal{E} = \bigcup_{i \geq 0} Th(E_i)$ would thus be to show that the set of active defaults blocking d decreases along our F-iteration. Unfortunately, the set of defaults blocking d may also increase because prerequisites of smaller defaults which have not been derivable at step i may become derivable in a later step of the iteration. For this reason, the proof of the theorem uses a more complex induction argument, first on i, and then on the size of the sets D_j^d of defaults potentially blocking d, where $D_j^d := \{d' = \alpha' : \beta'/\gamma' \mid d' < d$ and $(\alpha' \notin Th(E_j) \vee d'$ is active in $E_j)\}$ (see [1] for details). Our restriction on the partial ordering makes sure that the sets D_j^d are finite. □

Since not all default theories have R-extensions it follows that a default theory with specificity need not have an S-extension. But even if we have R-extensions there need not exist S-extensions of a default theory with specificity. This is demonstrated by the following example. Assume that $\mathcal{W}$ is empty, and consider the three defaults $:\beta/\beta$, $:\neg\beta/\neg\beta$ and $\beta : \alpha/\neg\alpha$. We assume that the first default is smaller than the second one, and that there are no other comparabilities with respect to $<$. This default theory has the R-extension $Th(\{\neg\beta\})$, but it does not have an S-extension. In fact, an S-extension would prefer the first default, which yields β; but then the third default (which is a modified version of the well-known one-rule example of a default theory having no R-extension) would become relevant.

As in the case without specificity, normal default theories with specificity have much nicer properties than arbitrary default theories with specificity.

Theorem 4.3 *Every closed normal default theory with specificity has an S-extension.*

The proof is a relatively straightforward adaptation of Reiter's proof for R-extensions (see [1] for details).

For normal default theories with specificity we also have "orthogonality of S-extensions" (see [17], Theorem 3.3), but we do not have "semi-monotonicity" (see [17], Theorem 3.2). For defaults with priorities, semi-monotonicity cannot be expected to hold since by adding a default of high priority one should of course be able to change the extensions considerably.

If we further restrict the attention to normal defaults without prerequisites then the notion of an S-extension coincides with that of a B-extension, which shows that our approach is a generalization of ordered default theories (see [1] for a proof).

Theorem 4.4 *Let $\mathcal{D}$ be a set of closed prerequisite-free normal defaults. Then $\mathcal{E}$ is an S-extension of the default theory with specificity $(\mathcal{W}, \mathcal{D}, <)$ iff $\mathcal{E}$ is a B-extension of the ordered default theory $(\mathcal{W}, \mathcal{D}, <)$.*

5 Computing S-extensions

Since all S-extensions are R-extensions, one could first generate all R-extensions of a default theory, and then for each R-extension $\mathcal{E}$ directly use the definition of S-extensions to check whether $\mathcal{E}$ is an S-extension. For terminological default theories this provides us with an effective procedure for computing all S-extensions. In fact, in [2] it is shown how to compute all R-extensions of a terminological default theory. Since one has only finitely many closed defaults, and since the instantiation problem for the terminological languages we use is decidable, the iteration in the definition of an S-extension is effective as well.

However, there may exist a lot more R-extensions than S-extensions, and computing R-extensions is rather expensive. For this reason, it would be preferable to have an algorithm for directly computing S-extensions. The idea behind the algorithm presented below is to make an iteration similar to the one in the definition of an S-extension, but without already having the final set $\mathcal{E}$ for controlling which consequents of defaults are added. After the iteration becomes stable (which will always be the case for finite sets of closed defaults) one has to check an additional condition to make sure that the result really is an S-extension.

The main problem is to determine which sets of consequents are candidates for being added in each step of the iteration. Of course there can be more than one correct choice because there may exist more than one S-extension. If we look at the definition of E_{i+1} in Definition 4.1 we see that the defaults whose consequents are added are defaults active in E_i that are minimal w.r.t. the priority order $<$. Which subset of their consequents is taken depends on the set $\mathcal{E}$ used for the iteration. Since our algorithm does not know the final $\mathcal{E}$ it has to consider arbitrary subsets, but we shall see that there are some constraints that reduce the number of possible choices. It should be noted that neither a greedy procedure (which takes maximal subsets that are consistent with what has already been computed) nor an overly modest procedure (which adds only one consequent in each step) would be complete (see [1] for examples). In the following (nondeterministic) algorithm, E_i will always be a subset of $\mathcal{W} \cup Con(\mathcal{D})$, and J_i will be a subset of $\neg Jus(\mathcal{D})$ (where, for a set $\mathcal{F}$ of formulae, $\neg\mathcal{F} := \{\neg\beta \mid \beta \in \mathcal{F}\}$).

Algorithm 5.1 *Let $(\mathcal{W}, \mathcal{D}, <)$ be a closed default theory with specificity. If $\mathcal{W}$ is inconsistent then $Th(\mathcal{W})$ is the only S-extension. Otherwise we define $E_0 := \mathcal{W}$ and $J_0 := \emptyset$. Now assume that E_i $(i \geq 0)$ is already defined. Consider*

$$\mathcal{D}_{i+1} := \{d \in \mathcal{D} \mid d \text{ is active in } E_i \text{ and no } d' < d \text{ is active in } E_i\},$$

and choose a nonempty subset $\widehat{\mathcal{D}}_{i+1}$ of $\mathcal{D}_{i+1}$ that satisfies

$$\neg\beta \notin Th(E_i \cup Con(\widehat{\mathcal{D}}_{i+1}) \cup J_i \cup \neg Jus(\mathcal{D}_{i+1} \setminus \widehat{\mathcal{D}}_{i+1}))$$

for all $\beta \in Jus(\widehat{\mathcal{D}}_{i+1})$.

If there is no such set, then $E_{i+1} := E_i$, $J_{i+1} := J_i$. Otherwise each choice yields new sets $E_{i+1} := E_i \cup Con(\widehat{\mathcal{D}}_{i+1})$ and $J_{i+1} := J_i \cup \neg Jus(\mathcal{D}_{i+1} \setminus \widehat{\mathcal{D}}_{i+1})$.

The set $\mathcal{E} := \bigcup_{i \geq 0} Th(E_i)$ is an S-extension iff

1. for all $d = \alpha : \beta/\gamma \in \bigcup_{i\geq 1} \widehat{\mathcal{D}}_i$ we have $\neg\beta \notin \mathcal{E}$, and
2. for all $\neg\beta \in \bigcup_{i\geq 1} J_i$ we have $\neg\beta \in \mathcal{E}$.

A proof of soundness and completeness of this algorithm can be found in [l]. The idea behind the sets J_i is as follows. If the consequent of a minimal active default is not included in E_{i+1}, then the reason must be that its justification is not consistent with the final extension. Thus, if we exclude such a default from $\widehat{\mathcal{D}}_{i+1}$, we know that the negation of its justification must belong to the extension. The condition on $\widehat{\mathcal{D}}_{i+1}$ corresponds to the fact that defaults whose consequents are added to an S-extension must have justifications that are consistent with the extension. The condition on $\widehat{\mathcal{D}}_{i+1}$ can only ensure local correctness of our choices. For this reason we have to check the two conditions on $\mathcal{E}$ to ensure global correctness.

For terminological default theories, all the steps of the algorithm are effective, provided that the consistency and instantiation problem for the underlying terminological language is decidable (an assumption which is usually satisfied). In addition, since one has only finitely many closed defaults, the iteration will become stable after finitely many steps.

6 Conclusion

We have addressed the question of how to prefer more specific defaults over more general ones. This problem is of general interest for default reasoning, but is even more important in the terminological case where the emphasis lies on the hierarchical organization of concepts. Of the existing approaches for handling priorities among defaults, Brewka's ordered default theories turned out to come nearest to what is needed for solving the specificity problem in terminological default theories. But its restriction to prerequisite-free normal defaults seems to be too severe to make it an adequate solution in the terminological case.

Therefore we have proposed a new approach, called default theories with specificity, for handling priorities among defaults with prerequisites. The properties we could prove for this formalism demonstrate that it is a quite reasonable extension of Reiter's default logic and of Brewka's ordered default theories. In addition it correctly handles examples for which the other approaches give unintuitive results. We have also described a method for generating the extensions of a default theory with specificity. This method is effective provided that the base logic is decidable, and one has only finitely many closed defaults. These restrictions are satisfied in the terminological case, which means that terminological default logic with specificity is decidable.

An interesting point for further research is to consider priorities on terminological defaults which not only take subsumption between prerequisites of defaults into account, but also the role relationships in ABoxes.

Acknowledgements We should like to thank Peter Patel-Schneider for interesting discussions on specificity of defaults, and Bernhard Nebel for helpful comments on a draft of this paper.

References

[1] F. Baader and B. Hollunder. How to prefer more specific defaults in terminological default logic. Research Report RR-92-58, DFKI Saarbrucken, 1992.

[2] F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. In Proceedings of KR'92, Cambridge, Mass., 1992.

[3] P. Besnard. An Introduction to Default Logic. Symbolic Computation Series. Springer, 1989.

[4] R. J. Brachman, D. L. McGuinness, P. F. Patel-Schneider, L. A. Resnick, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In J. Sowa, editor, Principles of Semantic Networks, pages 401-456. Morgan Kaufmann, San Mateo, Calif., 1991.

[5] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In Proceedings of 1JCAI'89, Detroit, Mich., 1989.

[6] G. Brewka. Nonmonotonic Reasoning: Logical Foundations of Commonsense. Cambridge University Press, Cambridge, 1991.

[7] J. P. Delgrande and W K Jackson. Default logic revisited. In Proceedings of KR'91, Cambridge, Mass., 1991.

[8] M L . Ginsberg, ed. Readings in Nonmonotonic Reasoning. Morgan Kaufmann, Los Altos, Cal., 1987.

[9] U. Junker and G. Brewka. Handling partially ordered defaults in TMS. In Proceedings of the 1st European Conference on Symbolic and Quantitative Approaches for Uncertainty, Marseille, France, 1991.

[10] V. Lifschitz. Computing circumscription. In Proceedings of IJCAI'85, Los Angeles, Calif., 1985.

[11] K. Mays and EL Dionne. Making KR systems useful. In [15], pages 11 12.

[12] R McGregor. Statement of interest. In K. von Luck, B. Nebel, and C. Peltason, editors, Statement of Interest for the 2nd International Workshop on Terminological Logics. Document D-91-13, DFKI Kaiserslautern, 1991.

[13] //BACK. System presentation. In [15], page 186.

[14] L. Padgharn and B. Nebel. Combining classification and nonmonotonic inheritance reasoning: A first step. In Z W. Ras and J. Komorowski, editors, Methodologies for Intelligent Systems (ISMIS'9S). North-Holland, Amsterdam, 1993. To appear.

[15] C. Peltason, K. v. Luck, and C. Kindermann (Org). Terminological logic users workshop - Proceedings. KIT Report 95, TU Berlin, 1991.

[16] D. L. Poole. A logical framework for default reasoning. Artificial Intelligence, pages 27 47, 1988.

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

[18] R. Reiter and G. Criscuolo. On interacting defaults. In Proceedings of IJCAP'81, Vancouver, BC, 1981.