ELSEVIER

Artificial Intelligence 133 (2001) 35–85

Artificial Intelligence

www.elsevier.com/locate/artint

An argument-based approach to reasoning with specificity

Phan Minh Dung [a], Tran Cao Son [b,*]

[a] *Department of Computer Science and Information Management, School of Advanced Technology, Asian Institute of Technology, PO Box 2754, Bangkok 10501, Thailand*

[b] *Department of Computer Science, New Mexico State University, PO Box 30001, MSC CS, Las Cruces, NM 88003, USA*

Received 17 December 1999; received in revised form 6 March 2001

Abstract

We present a new priority-based approach to reasoning with specificity which subsumes inheritance reasoning. The new approach differs from other priority-based approaches in the literature in the way priority between defaults is handled. Here, it is conditional rather than unconditional as in other approaches. We show that any unconditional handling of priorities between defaults as advocated in the literature until now is not sufficient to capture general defeasible inheritance reasoning. We propose a simple and novel argumentation semantics for reasoning with specificity taking the conditionality of the priorities between defaults into account. Since the proposed argumentation semantics is a form of stable semantics of nonmonotonic reasoning, it inherits a common problem of the latter where it is not always defined for every default theory. We propose a class of stratified default theories for which the argumentation semantics is always defined. We also show that acyclic and consistent inheritance networks are stratified. We prove that the argumentation semantics satisfies the basic properties of a nonmonotonic consequence relation such as deduction, reduction, conditioning, and cumulativity for well-defined and stratified default theories. We give a modular and polynomial transformation of default theories with specificity into semantically equivalent Reiter default theories. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Default reasoning; Specificity; Argumentation; Reasoning with specificity

* Corresponding author.
E-mail addresses: dung@cs.ait.ac.th (P.M. Dung), tson@cs.nmsu.edu (T.C. Son).

0004-3702/01/$ – see front matter
PII: S0004-3702(01)00134-5

1. Introduction

Default reasoning is a form of reasoning which is often employed by humans to make conclusions using commonsense knowledge even if some conclusions will turn out to be incorrect when new information is available. For instance, if all we know about *Tweety* is that it is a bird, we will conclude that *Tweety* flies because birds normally fly. If we later learn that *Tweety* is a penguin, we will withdraw that conclusion and infer that *Tweety* does not fly because (i) normally, penguins do not fly and (ii) *normally, conclusions supported by more specific information prevail over those supported by less specific ones*. While (i) represents a part of our common knowledge about penguins, (ii) does not. It is one of the generally accepted principles, often referred to as *the specificity principle*, used in default reasoning to resolve the conflict between contradictory conclusions. Default reasoning with specificity refers to default reasoning approaches which use the specificity principle as one of their conflict resolution strategies.

General approaches to nonmonotonic reasoning such as Reiter's default logic [42], McCarthy's circumscription [32], Moore's autoepistemic logic [36], or McDermott and Doyle's nonmonotonic logic [34] do not take specificity into consideration, i.e., the reasoning process in these approaches does not admit the specificity principle. For example, a naive representation of the above information about *Tweety* by the following default theory in Reiter's default logic notation

$$\left(\left\{penguin(Tweety), penguin(X) \supset bird(X)\right\},\right. \left.\left\{\frac{bird(X) : fly(X)}{fly(X)}, \frac{penguin(X) : \neg fly(X)}{\neg fly(X)}\right\}\right) \tag{$*$}$$

would not yield the intuitive conclusion that *Tweety* does not fly, i.e., $\neg fly(Tweety)$ cannot be concluded since the theory has two extensions and $\neg fly(Tweety)$ holds in one extension and does not hold in the other. The reason for this is the interaction between the two defaults in $(*)$, first noticed by Reiter and Criscuolo [43]. They discussed various situations, in which the interaction between defaults of a normal default theory can be compiled into the original theory to create a new default theory whose semantics yields the intuitive results. In the case of *Tweety*, their method yields the following default theory

$$\left(\left\{penguin(Tweety), penguin(X) \supset bird(X)\right\},\right. \left.\left\{\frac{bird(X) : \neg penguin(X) \wedge fly(X)}{fly(X)}, \frac{penguin(X) : \neg fly(X)}{\neg fly(X)}\right\}\right)$$

which entails $\neg fly(Tweety)$ because it has only one extension that contains $\neg fly(Tweety)$.

It has been recognized relatively early that priorities between defaults can help in dealing with specificity. Priorities can be used to remove unintuitive models. In prioritized circumscription, first defined by McCarthy [33], a priority order between predicates is added into each circumscription theory. Lifschitz [29] later proved that prioritized circumscription is a special case of parallel circumscription. A similar approach has been taken by Konolige [27] in using autoepistemic logic to reason with specificity. He defined hierarchical autoepistemic theories in which a preference order between sub-theories and a syntactical condition on the sub-theories ensure that higher priority conclusions will be concluded. Brewka [6]—in defining prioritized default logic—also adds a preference order between defaults into a Reiter's default theory and modifies the semantics of default logic in such a way that guarantees that default of higher priority is preferred. Baader and Hollunder [2] develops prioritized default logic to handle specificity in terminological systems. All of the approaches in [2,6,27,29,33] assume that priorities between defaults are given by the users. For this reason, these approaches are sometime called *reasoning with explicit specificity*.

Contrary to approaches to reasoning with explicit specificity are approaches to *reasoning with implicit specificity* in which a mechanism for computing the priority order between defaults is provided. Poole [40] is an early attempt to extract the preference between defaults from the theory. Poole defined a notion of more specific between pairs consisting of a conclusion and an argument supporting this conclusion. Moinard [35] pointed out that Poole's definition yields unnecessary priority, for example, it can arise even in consistent default theories. He also suggested five principles for establishing the priority between defaults. Simari and Loui [46] noted that Poole's definition does not take into consideration the interaction between arguments. To overcome this problem they combined Poole's approach and Pollock's theory [39] to define an approach that unifies various approaches to argument-based defeasible reasoning. Geffner and Pearl [20] also used an implicit priority order to define conditional entailment, an approach that exhibits the advantages of both conditional and extensional approaches to default reasoning. Conditional entailment, however, is too weak in that it does not capture inheritance reasoning. Pearl [38] also discussed how a preference relation between defaults can be established. In a later session, we will discuss Pearl's proposal in more details.

Obtaining specificity information is one problem, using specificity is another critical problem in reasoning with specificity. It can be used *directly* to define a new formalism that accounts for specificity. Examples of these systems can be found in [2,5,19,20, 22,38,40,46]. Specificity can also be used *indirectly*. The idea is to compile it into a general nonmonotonic reasoning approach thus avoiding the burden of introducing a new semantics. In the recent years, these approaches seem to get more attention than those using specificity directly [7–9,15,21]. Regardless of whether specificity is used directly or indirectly, in many approaches [2,5,9,19,20,22,38,40,46], the priority order is used unconditionally, independent of the concrete context. As we will show in Section 2, priority order should be used conditionally, if a general approach to reasoning with specificity were to capture nonmonotonic inheritance reasoning.

Argumentation has been recognized lately as an important and natural approach to nonmonotonic reasoning [1,3,10–13,15,20,24,25,39–41,46,50]. Dung [13] introduced a simple and abstract argumentation framework. Central to an argumentation framework is a notion of an argument and a binary relation, called the attack relation, between arguments. The semantics of an argumentation framework specifies what set of arguments is acceptable. Like other nonmonotonic logics, argumentation also has different types of semantics such as the *preferred, stable*, or *well-founded* semantics. Dung also proved that well-known nonmonotonic logics like autoepistemic logic, Reiter's default logic and logic programming represent different forms of a simple system of argumentation reasoning.

Based on the results in [13], a simple argument-based logical system has been developed in [3] which captures well-known nonmonotonic logics like autoepistemic logic, Reiter's default logic and logic programming as special cases.

Early attempt in using argumentation in default reasoning with specificity can be attributed to Poole [40] in which a more specific relation between pairs of arguments and conclusions is defined. Although the semantics provided by this approach is rather weak, it has inspired others to use argumentation in default reasoning with specificity. Geffner and Pearl [20] employed argumentation to give a proof procedure for conditional entailment. Simari and Loui [46] developed an argumentation system for reasoning with specificity. Both systems are rather weak in that it does not capture inheritance reasoning. On the other hand, reasoning based on arguments represented as paths, has been studied in nonmonotonic inheritance reasoning, a special field of nonmonotonic reasoning, from the very first day [51] and then in [23,26,44,47–49]. Path-based approaches to nonmonotonic inheritance networks are widely accepted because they are intuitive and easy to implement. In [14], we proved that argument-based approaches to inheritance reasoning could be viewed as a simple form of argumentation. In a later work [15], we extended this result and showed that argumentation offers a natural and intuitive framework for dealing with specificity. However, the expressibility of default theories in [15] is rather limited in that the language for representing default theories does not admit material implication and disjunction.

This paper is concerned with approaches in which an implicit priority order between defaults is used to resolve conflicts. We concentrate on two important questions of reasoning with specificity:

(1) How to compute specificity?
(2) How to use specificity?

We propose a novel method to assign priority order to defaults which can be seen as a generalized version of Touretzky's specificity principle in inheritance reasoning [51]. We also show that specificity must be applied conditionally if a general approach to reasoning with specificity were to capture general defeasible inheritance reasoning. Instead of developing a new system for reasoning with specificity, we compile specificity into an argumentation system and develop a simple and novel argumentation semantics for reasoning with specificity taking the conditionality of the priorities between defaults into account. The new framework improves our previous work [15] in two aspects. It eliminates the syntactical restrictions on default theories and the more specific relation is much simpler than the previously defined more specific relation. [1] We will show that our method overcomes the shortcoming of the existing proposals in the literature by proving that our formalism captures general inheritance reasoning. Since the proposed argumentation semantics is a form of stable semantics of nonmonotonic reasoning, it inherits a common problem of the latter where it is not always defined for every default theory. We propose a large class of stratified default theories for which the argumentation semantics is always defined. We also show that acyclic and consistent inheritance networks [23] are stratified. We prove that the argumentation semantics satisfies the basic properties of nonmonotonic consequence relations such as deduction, reduction, conditioning, and cumulativity for

[1] Section 6.1 provides a detailed comparison between the two approaches.

well-defined and stratified default theories. To compute the newly defined entailment relation, we transform default theories with specificity into semantically equivalent Reiter's default logic (for a collection of algorithms for Reiter's default logic, the reader can consult [30]). The translation is *modular* with respect to the extension of a default theory. Most importantly, it is *polynomial* in the size of the original default theory.

The paper is organized as follows. We first argue that in inheritance reasoning, *specificity between defaults is conditional thus can not be used unconditionally* (Section 2). In Section 3, we present our approach to reasoning with specificity and define an argumentation semantics for it. We then study the existence of the proposed semantics. In Section 4, we introduce the class of stratified default theories and study the general properties of the newly defined semantics. We show that acyclic inheritance networks are stratified default theories. In Section 5, we give a polynomial transformation of our framework into Reiter's default logic. We relate our approach to other approaches in Section 6. Finally, we conclude in Section 7.

2. Why should specificity be conditional?

Formally a default theory T could be defined as a pair (E, K) where E is a set of evidence or facts representing what we call the concrete context of T, and $K = (D, B)$ constitutes the domain knowledge consisting of a set of default rules D and a first-order theory B representing the background knowledge. In the literature [2,5,9,19,20,38] the principle of reasoning with specificity is "enforced" by first determining a set of priority orders between defaults in D using the information given by the domain knowledge K. Based on these priorities between defaults and following some sensible and intuitive criteria, the semantics of T is then defined either model-theoretically by selecting a subset of the set of all models of $E \cup B$ as the set of preferred models of T or proof-theoretically by selecting certain extensions as preferred extensions. The problem of these approaches is that their semantics is rather weak: *they do not capture general defeasible inheritance reasoning*. There are many intuitive examples of reasoning with specificity (one of them is given below) that cannot be handled in these approaches. The reason is that the priorities between defaults are conditional thus cannot be used unconditionally.

Priority orders are strict partial orders [2] between defaults in D. Let PO_K be the set of all such priority orders. For each priority order $\alpha \in PO_K$, where $(d, d') \in \alpha$ means that d is of lower priority than d', a priority order $<_\alpha$ between the sets of defaults in D is defined where $S <_\alpha S'$ means that S is preferred to S'. There are many ways to define $<_\alpha$ [2,5, 9,19,20,22,38,40]. But whatever the definition of $<_\alpha$ is, $<_\alpha$ has to satisfy the following property.

Let d, d' be two defaults in D such that $(d, d') \in \alpha$. Then $\{d'\} <_\alpha \{d\}$.

$<_\alpha$ can be extended into a partial order between models of $B \cup E$ as follows:

$$M <_\alpha M' \quad \text{iff} \quad D_M <_\alpha D_{M'}$$

[2] Strict partial orders are transitive, irreflexive and antisymmetric relations.

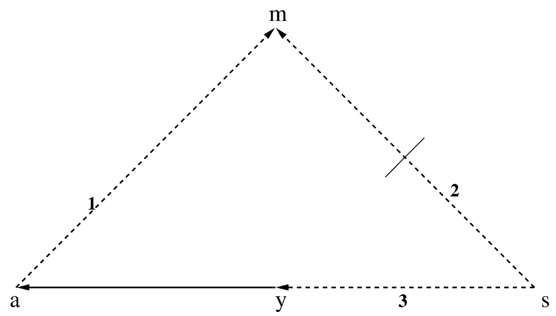

Fig. 1. Student-adult-married network.

where D_M is the set of all defaults in D which are true in M whereas a default $p \to q$ is said to be true in M iff the material implication $p \Rightarrow q$ is true in M.

A model M of $B \cup E$ is defined as a preferred model of T if there exists a partial order α in PO_K such that M is minimal with respect to $<_\alpha$. We then say that a formula β is defeasibly derived from T if β holds in each preferred model of T.

Now we want to show that any preferential semantics based on $<_\alpha$ cannot account in full for general inheritance reasoning.

Example 2.1. Let us consider the following inheritance network [3] (Fig. 1), where the links $s \not\to m$, $a \to m$, and $s \to y$ represent the normative sentences “normally, students are not married”, “normally, adults are married”, and “normally, students are young adults”, respectively, and, the strict link $y \Rightarrow a$ represents the subclass relation “young adults are adults”.

This defeasible inheritance network represents the domain knowledge (B, D) with $B = \{y \Rightarrow a\}$, and $D = \{d_1 : a \to m,\ d_2 : s \to \neg m,\ d_3 : s \to y\}$.

Consider now the marital status of a young adult who is also a student. The problem is represented by the default theory $T = (E, B, D)$ with $E = \{s, y, a\}$. The desirable semantics here is represented by the model $M = \{s, y, a, \neg m\}$. To deliver this semantics, all priority-based approaches in the literature [2,5,9,19,38] assigns default 1 a lower priority than default 2.

Let us consider now the marital status of another student who is an adult but not a young one. Let $T' = (E', B, D)$ with $E' = \{s, \neg y, a\}$. Now, since y does not hold, default 2 cannot be considered more specific than default 1. Hence, it is intuitive to expect that neither m nor $\neg m$ should be concluded in this case. This is also the result sanctioned by all semantics of defeasible inheritance networks [23,26,44,47–49]. In any priority-based system employing the same priorities between defaults with respect to E' as with respect to E, we have $M = \{\neg m, s, \neg y, a\} <_\alpha M' = \{m, s, \neg y, a\}$ since $D_M = \{2\} <_\alpha D'_M = \{1\}$ (due to $(1, 2) \in \alpha$). That means priority-based approaches in the literature conclude $\neg m$ given (E', K) which is not the intuitive result we expect.

[3] Throughout the paper, solid lines and dash lines represent strict rules and default rules, respectively.

To produce a correct semantics, 1 should have lower priority than 2 only when default 3 can be applied and hence making default 2 more specific than default 1. In general, the example shows that *specificity-induced priorities between defaults is conditional*.

3. A general framework

We assume a propositional language $\mathcal{L}$. For convenience, we use variables in our representation and a formula with variables is viewed as shorthand of the set of its ground instantiations. The set of ground literals of $\mathcal{L}$ is denoted by $lit(\mathcal{L})$. Literals of $\mathcal{L}$ will be called hereafter simply literals (or $\mathcal{L}$-literals) for short. Following the literature, a default theory is defined as follows:

Definition 3.1. A default theory T is a triple (E, B, D) where

(i) E is a set of ground literals representing the evidence of the theory,
(ii) B is a set of ground clauses,
(iii) D is a set of defaults of the form $l_1 \wedge \cdots \wedge l_n \rightarrow l_0$ where l_i's are ground literals, and
(iv) $E \cup B$ is a consistent theory.

Notice that in the above definition, we use $\rightarrow$ to denote a default implication. The material implication is represented by the $\Rightarrow$ symbol. Intuitively, $a \rightarrow b$ means that "typically, if a holds then b holds" while $a \Rightarrow b$ means that "whenever a holds then b holds". For a default $d \equiv l_1 \wedge \cdots \wedge l_n \rightarrow l_0$, we denote $l_1 \wedge \cdots \wedge l_n$ and l_0 by $bd(d)$ and $hd(d)$, respectively.

Example 3.1. Consider the famous penguin and bird example with $B = \{p \Rightarrow b\}$ (penguins are birds) and D consisting of two defaults $p \rightarrow \neg f$ (normally, penguins do not fly) and $b \rightarrow f$ (normally, birds fly).

The question is whether penguins fly. This problem is represented by the default theory $T = (E, B, D)$ where $E = \{p\}$.

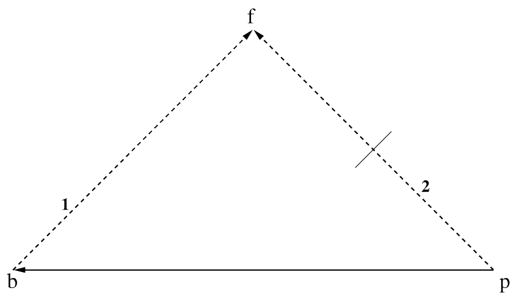

Fig. 2. Penguin-bird-fly network.

We next define the notion of *defeasible derivation* that will be used to draw conclusions given a default theory. Intuitively, a defeasible derivation represents a possible proof for a conclusion.

Definition 3.2. Let $T = (E, B, D)$ be a default theory and l be a ground literal.

- A sequence of defaults $d_1, \ldots, d_n$ $(n \geqslant 0)$ is said to be a *defeasible derivation of l* if following conditions are satisfied:
  (1) $n = 0$ and $E \cup B \vdash l$ where the relation $\vdash$ represents the first-order consequence relation, or
  (2) $n > 0$ and
    (a) $E \cup B \cup \{hd(d_1), \ldots, hd(d_i)\} \vdash bd(d_{i+1})$ for $i \in \{1, \ldots, n-1\}$, and
    (b) $E \cup B \cup \{hd(d_1), \ldots, hd(d_n)\} \vdash l$.
- We say l is a *possible consequence of E with respect to B and a set of defaults $K \subseteq D$*, denoted by $E \cup B \vdash_K l$, if there exists a defeasible derivation $d_1, \ldots, d_n$ of l such that for all $1 \leqslant i \leqslant n$, $d_i \in K$.

For a set of literals L we write $E \cup B \vdash_K L$ iff $\forall l \in L$: $E \cup B \vdash_K l$. We write $E \cup B \vdash_K \bot$ [4] iff there is an atom a such that both $E \cup B \vdash_K a$ and $E \cup B \vdash_K \neg a$ hold. For the default theory T from Example 2.1, it is easy to check that $E \cup B \vdash_{\{s \to \neg m\}} \neg m$ and $E \cup B \vdash_{\{s \to y,\ a \to m\}} m$. Hence $E \cup B \vdash_D \bot$. We say that a set of defaults K is *consistent in T* if $E \cup B \nvdash_K \bot$. K is *inconsistent* if it is not consistent. [5]

3.1. The "more specific" relation

We now define the notion of "more specific" between defaults which generalizes the specificity principle of Touretzky in inheritance reasoning. Consider for example the network from Example 2.1, it is clear that being a student is more specific than being a young adult. Since being a young adult is always more specific than being an adult, it follows that being a student is more specific than being an adult if the respective individual is a young adult. This stipulates us to say that the default $s \to \neg m$ (students are normally not married) is more specific than the default $a \to m$ (adults are normally married) provided that the default $s \to y$ (students are normally young adults) can be applied. Similarly, in Example 3.1, since penguins are birds we can conclude that the default $p \to \neg f$ (penguins do not fly) is always more specific than $b \to f$ (birds fly). This discussion leads to the following definition.

Definition 3.3. Let $d_1,\ d_2$ be two defaults in D. We say that d_1 is *more specific* than d_2 with respect to a set of defaults $K \subseteq D$, denoted by $d_1 \prec_K d_2$, if

(i) $B \cup \{hd(d_1), hd(d_2)\}$ is inconsistent,
(ii) $bd(d_1) \cup B \vdash_K bd(d_2)$, and
(iii) $bd(d_1) \cup B \nvdash_K \bot$.

[4] Throughout the paper, we use $\top$ and $\bot$ to denote *True* and *False*, respectively.
[5] If there is no possibility for misunderstanding, we often simply say consistent instead of consistent in T.

In the above definition (i) guarantees that a priority is defined between two defaults only if they are in conflict, (ii) ensures that being $bd(d_1)$ is a special case of being $bd(d_2)$ provided that the defaults in K can be applied, and (iii) guarantees that K is a consistent set of defaults. *We could say that this is a generalization of Touretzky's specificity principle to general propositional default theories*. In [15], the more specific relation is defined based on the notion of minimal conflict set, which in turn is defined based on the notion of defeasible derivation. As it can be seen, the above definition is much simpler than that was proposed in [15]. Besides, it allows us to deal with default theories with nonempty background knowledge. In a later section, we will discuss this in more details. When $K = \emptyset$ we say that d_1 is *strictly more specific* than d_2 and write $d_1 < d_2$ instead of $d_1 \prec_\emptyset d_2$.

Example 3.2. In Example 2.1, $d_2 \prec_{\{d_3\}} d_1$ holds, i.e., d_2 is more specific than d_1 if d_3 is applicable.

In Example 3.1, it is obvious that $d_2 < d_1$, i.e., d_2 is strictly more specific than d_1.

Notice that for the default theory in Example 2.1, even though $bd(d_3) \cup B \vdash bd(d_1)$, the relation $d_3 < d_1$ does not hold because d_3 and d_1 are not in conflict, i.e., $B \cup \{hd(d_3), hd(d_1)\} \nvdash \bot$. That is, instead of saying that a default is more specific than another default if its body is more specific than that of the other's one, we employ a stronger notion of more specific here. Thus, our approach to specificity could be referred to as *specificity-with-conflict*. [6] This allows us to combine both specificity and inconsistency into a simple, but central to argumentation reasoning, notion of 'attack' (defined below) which will be used for conflict resolution. Further, the stable semantics defined for default theories in this paper is a kind of credulous semantics that admits maximal set of conclusions when no conflict arises. Therefore, a more specific relation among non-conflicting pairs of defaults would be spurious.

3.2. Stable semantics of default reasoning with specificity

The semantics of a default theory is defined by determining which defaults can be applied to draw new conclusions from the evidence. For example, the semantics of the network in Example 2.1 is defined by determining that the defaults which could be applied are 2 and 3.

In the following, we will see that an argumentation-theoretic notion of attack between a set of defaults K and a default d lies at the heart of the semantics of reasoning with specificity.

Suppose that $K \subseteq D$ is a set of defaults we can apply. Further let d be a default such that $E \cup B \vdash_K \neg hd(d)$. It is obvious that d should not be applied together with K. In this case, we say that *K attacks d by conflict*. For illustration of attack by conflict, consider the default theory T in Example 2.1. Let $K = \{d_3, d_2\}$. Since $E \cup B \vdash_K \neg m$, K attacks d_1 by conflict. Similarly, $K' = \{d_3, d_1\}$ attacks d_2 by conflict because $E \cup B \vdash_{K'} m$.

The other case where d should not be applied together with K is where it is less specific than some default with respect to K. Formally, this means that if there exists $d' \in D$

[6] Suggested by a reviewer.

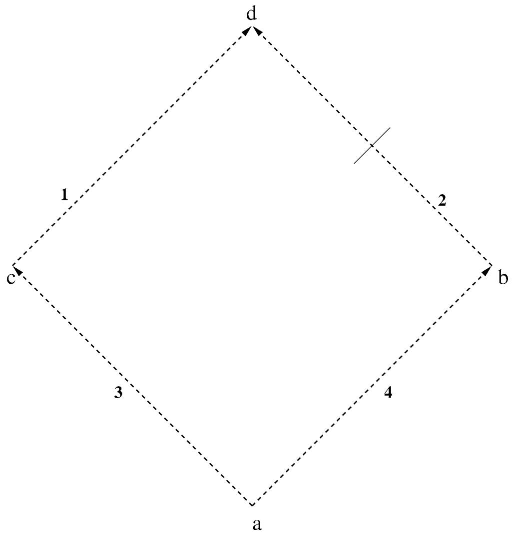

Fig. 3. Nixon-diamond.

such that $d' \prec_K d$ and $E \cup B \vdash_K bd(d')$ then d should not be applied together with the defaults in K. In this case we say that *K attacks d by specificity*. For illustration of attack by specificity, consider again the default theory T in Example 2.1. Let $K = \{d_3\}$. Because $d_2 \prec_{\{d_3\}} d_1$ and $E \cup B \vdash_{\{d_3\}} bd(d_2)$, K attacks d_1 by specificity.

The following definition summarizes what we have just discussed:

Definition 3.4. Let $T = (E, B, D)$ be a default theory. A set of defaults K is said to attack a default d in T [7] if:

(1) (*Attack by conflict*) $E \cup B \vdash_K \neg hd(d)$, or
(2) (*Attack by specificity*) There exists $d' \in D$ such that $d' \prec_K d$ and $E \cup B \vdash_K bd(d')$.

Note that there is an important difference between attack by conflict and inconsistency. It is possible that though K is consistent and $K \cup \{d\}$ is inconsistent but K does not attack d by conflict. It is also possible that K attacks some default d by conflict though $K \cup \{d\}$ is consistent. The Nixon diamond example (Fig. 3) illustrates these points.

Let $E = \{a\}$, $B = \emptyset$, and

$$D = \{d_1 : c \to d,\ d_2 : b \to \neg d,\ d_3 : a \to c,\ d_4 : a \to b\}.$$

Though $K = \{d_1, d_2, d_4\}$ is consistent and $K \cup \{d_3\}$ is inconsistent, K does not attack d_3 by conflict. Further, though $K' = \{d_2, d_4\}$ attacks d_1 by conflict, $K = K' \cup \{d_1\}$ is consistent.

[7] If there is no possibility for misunderstanding then T is often omitted.

K is said to attack some set $H \subseteq D$ if K attacks some default in H. K is said to *attack itself* if K attacks K. The stable semantics of argumentation is often defined as a set of arguments that does not attack itself and that attacks every argument not belonging to it. In the next definition, we employ this type of semantics in defining the semantics of a default theory with specificity.

Definition 3.5. Let $T = (E, B, D)$ be a default theory. A set of defaults S is called an *extension* of T if S does not attack itself and attacks every default not belonging to it.

Definition 3.6. Let $T = (E, B, D)$ be a default theory. Let l be a ground literal. We say T *entails* l, denoted by $T \mathrel{|\!\!\sim} l$, if for every extension S of T, $E \cup B \vdash_S l$.

Because the defeasible consequence relation $\vdash_K$ subsumes the first-order consequence relation (Definition 3.2), it is obvious that an inconsistent set of defaults attacks every default. Therefore it is clear that an extension is always consistent. We illustrate Definition 3.5 in the next examples.

Example 3.3. Consider the theory in Example 3.1. We have that $d_2 < d_1$, i.e., d_2 is strictly more specific than d_1. Let $K = \{d_1\}$ and $H = \{d_2\}$. Because $\{p\} \cup B \vdash_H \neg f$ and $\{p\} \cup B \vdash_K f$, we have that K and H attack d_2 and d_1 by conflict, respectively. Hence both K and H attack every default not belonging to it. But while H does not attack itself, K attacks itself by specificity because $d_1 \in K$, $d_2 < d_1$, and $\{p\} \cup B \vdash_K bd(d_2)$. Hence H is the unique extension of T. Therefore $T \mathrel{|\!\!\sim} \neg f$.

Example 3.4.

(1) Consider the theory T in Example 2.1. Let $H = \{d_3, d_2\}$. Because $\{s, y, a\} \cup B \vdash_H \neg m$, H attacks d_1 by conflict. Furthermore, since $\{s, y, a\} \cup B \nvdash_H m$ and $\{s, y, a\} \cup B \nvdash_H \neg y$, H does not attack itself by conflict. Because there is no default which is more specific than d_2 or d_3 with respect to H, H does not attack itself by specificity. Hence H does not attack itself and attacks every default not belonging to it. Therefore H is an extension of T.
Let $K = \{d_1, d_3\}$. Because $d_2 \prec_K d_1$ and $\{s, y, a\} \cup B \vdash_K bd(d_2)$, K attacks d_1 by specificity. Hence K is not an extension of T. It should be obvious now that H is the only extension of T. Hence, $T \mathrel{|\!\!\sim} \neg m$.

(2) Consider the theory T' in Example 2.1. Let $H = \{d_2\}$ and $K = \{d_1\}$. Since $\{s, \neg y, a\} \vdash_H \neg m$ and $\{s, \neg y, a\} \vdash_K m$, and $\{s, \neg y, a\} \vdash_\emptyset \neg y$, H attacks d_1, d_3 by conflict while K attacks d_2, d_3 by conflict. Due to the fact that there are no defaults d, d' such that $d \prec_H d'$ or $d \prec_K d'$, both H and K do not attack themselves. Thus, both H and K are extensions of T', and so, $T' \mathrel{|\!\!\not\sim} \neg m$ and $T' \mathrel{|\!\!\not\sim} m$.

Definition 3.5 of an extension of a default theory corresponds to the stable semantics of argumentation which has been first introduced in [13] and later further studied in [3]. There are also a number of other semantics for argumentation which could be applied to reasoning with specificity. But in this paper we will limit ourselves to the stable semantics.

3.3. Existence of extensions

A well-known problem of stable semantics in nonmonotonic reasoning is that it is not always defined for every nonmonotonic logic. For example, stable model semantics is not always defined for logic programs, i.e., there exist logic programs which do not possess a stable model. The same holds for autoepistemic logic, i.e., not every autoepistemic theory has a stable expansion. Similarly, there exists argumentation framework without stable extensions. As our semantics is a form of stable semantics of argumentation, it is expected that the same problem will be encountered in our framework. The following example originated from [9] confirms our expectation.

Example 3.5 [9]. Consider $T = (E, \emptyset, D)$ with $E = \{a, b, c\}$ and D consists of the following defaults

$$d_1 : a \wedge q \rightarrow \neg p,$$
$$d_2 : a \rightarrow p,$$
$$d_3 : b \wedge r \rightarrow \neg q,$$
$$d_4 : b \rightarrow q,$$
$$d_5 : c \wedge p \rightarrow \neg r,$$
$$d_6 : c \rightarrow r.$$

Here we have that $d_1 < d_2$, $d_3 < d_4$, and $d_5 < d_6$.

It is easy to see that for each $K \subseteq D$, there is no $d \in D$ such that $d \prec_K d_1$ or $d \prec_K d_3$ or $d \prec_K d_5$.

We will prove that T does not have an extension.

Assume the contrary that T has an extension S. We want to prove that $d_1 \notin S$. Assume the contrary that $d_1 \in S$. Since $E \vdash_{\{d_2\}} p$ and S does not attack itself, we conclude that $d_2 \notin S$. This implies that S attacks d_2. There are two cases:

(1) S attacks d_2 by conflict. This means that $E \vdash_S \neg p$, which implies that $E \vdash_S q$.
(2) S attacks d_2 by specificity. Since the only default in D, that is more specific than d_2, is d_1, S attacks d_2 by specificity implies that $E \vdash_S bd(d_1)$. Thus $E \vdash_S q$.

It follows from the above two cases that $E \vdash_S q$. Therefore S contains d_4. Now, consider the two defaults d_5 and d_6. Since $d_2 \notin S$, $E \not\vdash_S bd(d_5)$. Therefore S does not attack d_6 by specificity. Further $E \not\vdash_S bd(d_5)$ implies that $E \not\vdash_S \neg r$. So, S does not attack d_6 by conflict either. Again, because S is an extension, we have that $d_6 \in S$. However, $E \vdash_{\{d_6\}} bd(d_3)$, which implies that S attacks d_4 by specificity, i.e., S attacks itself. This contradicts the assumption that S is an extension of T. Thus the assumption that $d_1 \in S$ leads to a contradiction. Therefore $d_1 \notin S$.

Similarly, we can prove that $d_3 \notin S$ and $d_5 \notin S$. Since S is an extension of T, S attacks d_1. This implies that S must attack d_1 by conflict because there is no default in D which is more specific than d_1. Thus $d_2 \in S$. Similar arguments lead to $d_4 \in S$ and $d_6 \in S$, i.e., $S = \{d_2, d_4, d_6\}$. However, S attacks d_2 by specificity because $d_1 < d_2$ and $E \cup B \vdash_S bd(d_1)$. This means that S attacks itself which contradicts the assumption that S is an extension of T. Thus the assumption that there exists an extension leads to a contradiction. Therefore, we can conclude that there exists no extension of T.

In the next section we will introduce the class of stratified default theories for which extensions always exist. We will also show that this class of theories is large enough to cover general inheritance reasoning.

4. Stratified default theories

The definition of stratified default theories is based on the notion of a *rank function* which is a mapping from the set of ground literals $lit(\mathcal{L}) \cup \{\top, \bot\}$ to the set of nonnegative integers.

Definition 4.1. A default theory $T = (E, B, D)$ over $\mathcal{L}$ is *stratified* if there exists a rank function of T, denoted by $rank$, satisfying the following conditions:
(1) $rank(\top) = rank(\bot) = 0$,
(2) for each ground atom l, $rank(l) = rank(\neg l)$,
(3) for all literals l and l' occurring in a clause in B, $rank(l) = rank(l')$, and
(4) for each default $l_1, \ldots, l_m \to l$ in D, $rank(l_i) < rank(l)$, $i \in \{1, \ldots, m\}$.

It is not difficult to see that all the default theories in Examples 2.1 and 3.1 are stratified. The following theorem shows that stratification guarantees the existence of extensions.

Theorem 4.1. *Every stratified default theory has at least one extension.*

Proof. In Appendix A.1. □

4.1. General properties of $\mid\!\sim$

There is a large body of work in the literature [4,20,28,31] on what properties characterize a defeasible consequence relation like $\mid\!\sim$. In general, it is agreed that such relation should extend the monotonic logical consequence relation. Further, since the intuition of a default rule d is that $bd(d)$ normally implies $hd(d)$, we expect that in the context $E = \{bd(d)\}$, $T \mid\!\sim hd(d)$ holds. Another important property of defeasible consequence relations is related to the adding of proved conclusions to a theory. Intuitively, this means that if $T \mid\!\sim a$ then we expect T and $T + a$ [8] to have the same set of conclusions. Formally, the discussed key properties are given below:

- *Deduction*: $T \mid\!\sim l$ if $E \cup B \vdash l$,
- *Conditioning*: If $E = \{bd(d)\}$ for $d \in D$, then $T \mid\!\sim hd(d)$,
- *Reduction*: If $T \mid\!\sim a$ and $T + a \mid\!\sim b$ then $T \mid\!\sim b$, and
- *Cumulativity*: [9] If $T \mid\!\sim a$ and $T \mid\!\sim b$ then $T + a \mid\!\sim b$,

where $T + a$ denotes the default theory $(E \cup \{a\}, B, D)$.

[8] $T + a$ denotes the default theory $(E \cup \{a\}, B, D)$.
[9] In [20], this property is called augmentation.

It is obvious that whatever entailed by $E \cup B$ is also entailed by T. Hence, we have the following theorem.

Theorem 4.2 (Deduction). *Let $T = (E, B, D)$ be an arbitrary default theory. Then, for every $l \in lit(\mathcal{L})$, $E \cup B \vdash l$ implies $T \mathrel{|\!\!\sim} l$.*

It is also easy to see that if $T \mathrel{|\!\!\sim} a$ then every extension of T is also an extension of $T + a$. Therefore from $T + a \mathrel{|\!\!\sim} b$, it is obvious that $T \mathrel{|\!\!\sim} b$. That means that $\mathrel{|\!\!\sim}$ satisfies the reduction property.

Theorem 4.3 (Reduction). *Let $T = (E, B, D)$ be an arbitrary default theory and $a, b \in lit(\mathcal{L})$ such that $T \mathrel{|\!\!\sim} a$ and $T + a \mathrel{|\!\!\sim} b$. Then, $T \mathrel{|\!\!\sim} b$.*

Though the entailment relation $\mathrel{|\!\!\sim}$ satisfies deduction and reduction, it does not satisfy cumulativity in general as the following example shows.

Example 4.1. Consider the default theory $T = (E, B, D)$ (Fig. 4) where

$$E = \{f\}, \quad B = \emptyset, \quad D = \{d_1 : f \to a,\ d_2 : a \to c,\ d_3 : c \to \neg a\}.$$

Because the only member of the more specific relation is $d_1 \prec_{\{d_1,d_2\}} d_3$, T has a unique extension $\{d_1, d_2\}$. Hence, $T \mathrel{|\!\!\sim} a$ and $T \mathrel{|\!\!\sim} c$.

Now consider $T + c$. $T + c$ has two extensions: $\{d_1, d_2\}$ and $\{d_2, d_3\}$. Thus, $T + c \mathrel{|\!\!\not\sim} a$. This implies that $\mathrel{|\!\!\sim}$ is not cumulative.

The next theorem proves that stratification is sufficient to guarantee cumulativity.

Theorem 4.4 (Cumulativity). *Let $T = (E, B, D)$ be a stratified default theory and a, b be literals such that $T \mathrel{|\!\!\sim} a$, and $T \mathrel{|\!\!\sim} b$. Then $T + a \mathrel{|\!\!\sim} b$.*

Proof. In Appendix A.2. $\square$

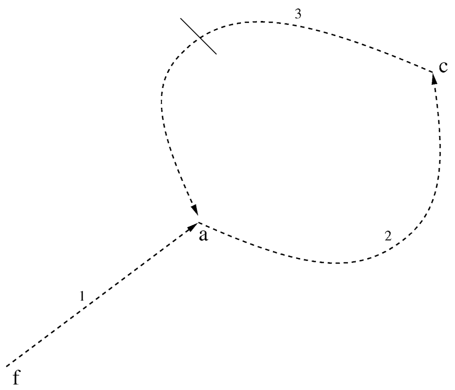

Fig. 4. A noncumulative default theory.

Because stratification does not rule out the coexistence of defaults like $a \to \neg c,\ a \to c$, conditioning does not hold for stratified theories as the following example shows.

Example 4.2. Let $T = (\{a\}, \emptyset, \{d_1 : a \to \neg c,\ d_2 : a \to c\})$. It is obvious that T is stratified. Because $d_1 < d_2$ and $d_2 < d_1$, both d_1, d_2 are attacked by specificity by the empty set of defaults. Thus the only extension of T is the empty set. Hence, $T \not\mathrel{|\!\sim} \neg c$, and $T \not\mathrel{|\!\sim} c$. That means that conditioning is not satisfied.

The coexistence of defaults like $a \to \neg c,\ \ a \to c$ means that a is normally c and normally $\neg c$ at the same time which is obviously not sensible. Hence it should not be a surprise that conditioning is not satisfied in such cases.

The conditioning property would hold for a default d if in the context of $bd(d)$, d is the most specific default. The following definition formalizes this intuition. Let $d \prec d'$ if $d \prec_K d'$ for some K. Let $\prec^*$ be the transitive closure of $\prec$.

Definition 4.2. A default theory $T = (E, B, D)$ is said to be *well-defined* if for every default d:

(1) $d \not\prec^* d$, and
(2) for every set $K \subseteq D$ such that $bd(d) \cup B \vdash_{K \cup \{d\}} \bot$ and $bd(d) \cup B \nvdash_K \bot$, there exist $d' \in K$ such that $d \prec_K d'$.

Theorem 4.5 (Conditioning). *Let $T = (E, B, D)$ be a well-defined default theory, d be a default in D, and $E = bd(d)$. Then $T \mathrel{|\!\sim} hd(d)$.*

Proof. In Appendix B. $\square$

It is interesting to note that well-definedness and stratification are two independent concepts. Default theories like the one in Example 4.1 are well-defined but not stratified while default theories like that in Example 4.2 are stratified but not well-defined. Further while the Example 4.2 shows that stratification does not imply conditioning, Example 4.1 shows that well-definedness does not imply cumulativity.

We will show shortly that acyclic and consistent inheritance networks are stratified and well-defined default theories.

4.2. Inheritance networks as stratified and well-defined default theories

In this subsection, we show that each inheritance network Γ can be viewed as a default theory T_Γ and the semantics of the latter (as defined by Definition 3.5) captures the credulous semantics of the former. Many different kinds of semantics of inheritance networks have been proposed in the literature [23,26,44,49,51]. Among them, the off-path credulous semantics is probably the most well-known and accepted semantics. In this subsection, we will prove that the off-path credulous semantics of an inheritance network Γ coincides with the stable semantics of T_Γ. We will not discuss the other types of semantics of inheritance networks here but we believe that they too could also be formalized within our framework. Technically, each semantics of inheritance networks relies on its own definition of the more specific relation between paths. As such, we only need to change the definition of the more specific relation accordingly, and the rest would follow. However, developing a framework that captures all well-known semantics of inheritance networks is in itself an interesting problem and deserves a separate study. For that reason, we leave it out as a future work and continue with a brief review of basic definitions of inheritance networks (see e.g. [23]).

An inheritance network Γ is a directed graph with two types of nodes and four types of links: *individual nodes*, *predicate nodes* and *strict positive*, *strict negative*, *defeasible positive*, and *defeasible negative* links. A node x is an individual node if there is no link which ends at x. Otherwise, it is a predicate node. A strict positive (respectively negative) *link* from x to y is denoted by $x \Rightarrow y$ or $y \Leftarrow x$ (respectively $x \not\Rightarrow y$ or $y \not\Leftarrow x$). A defeasible positive (respectively negative) *link* from x to y is denoted by $x \rightarrow y$ (respectively $x \not\rightarrow y$).

Using the above representation, the inheritance network in Example 2.1 can be represented by the set of links $\{s \not\rightarrow m,\ s \rightarrow y,\ y \Rightarrow a,\ a \rightarrow m\}$.

Notice the difference between strict link representation and defeasible link representation here. The reason lies in the fact that paths can be extended (to a longer path) from both ends of a strict link but only from the ending node of a defeasible link (see Definition 4.3). For instance, both $a \rightarrow b \Leftarrow c$ and $a \rightarrow d \Rightarrow c$ are considered as a path from a to c but $a \rightarrow b \leftarrow c$ is not.

Semantically, individual nodes and predicate nodes in Γ represent the constants and the unary predicates in T_Γ, respectively. Strict links denote material implication while defeasible links represent defaults. Hence an inheritance network Γ can be translated into a default theory T_Γ as follows:

Let I_Γ and $Pred_\Gamma$ be the set of individual and predicates nodes in Γ, respectively. The language $\mathcal{L}_\Gamma$ of T_Γ consists of

(1) the set of constants I_Γ and
(2) the set of unary predicate symbols $Pred_\Gamma$.

From the definition of $\mathcal{L}_\Gamma$, it is easy to see that each literal in $\mathcal{L}_\Gamma$ has the form $p(a)$ or $\neg p(a)$ where a is an individual node and p is a predicate node. $T_\Gamma = (E_\Gamma, B_\Gamma, D_\Gamma)$ is defined by [10]

(1) *Facts*: for every individual node a and a link $a \rightarrow p$ or $a \Rightarrow p$ (respectively $a \not\rightarrow p$ or $a \not\Rightarrow p$) in Γ, E_Γ contains $p(a)$ (respectively $\neg p(a)$),
(2) *Clauses*: for every strict link $p \Rightarrow q$ (respectively $p \not\Rightarrow q$) in Γ, $p \notin I_\Gamma$, B_Γ contains the clause $p(X) \Rightarrow q(X)$ (respectively $p(X) \Rightarrow \neg q(X)$), and
(3) *Defaults*: for every defeasible link $p \rightarrow q$ (respectively $p \not\rightarrow q$) in Γ, $p \notin I_\Gamma$, D_Γ contains the default $p(X) \rightarrow q(X)$ (respectively $p(X) \rightarrow \neg q(X)$).

It is easy to verify that the default theories in Examples 2.1 and 3.1 are obtained from the transformation of the corresponding inheritance networks with one (implicit) individual node linked to s in Example 2.1 and to p in Example 3.1.

Reasoning in inheritance network are represented by paths which are formally defined as special sequences of links and are classified into direct, compound, strict, defeasible, negative, or positive paths. A positive (respectively negative) path from x to y through

[10] Note that clauses or defaults with variables are considered a shorthand for the set of their ground instantiations.

a path σ is often denoted by $\pi(x,\sigma,y)$ (respectively $\bar{\pi}(x,\sigma,y)$). Paths are defined inductively as follows.

Definition 4.3 (*Paths* [23]).

(1) *Direct path*: A strict positive (respectively negative) link is a strict positive (respectively negative) path. Similarly, a defeasible positive (respectively negative) link is a defeasible positive (respectively negative) path.

(2) *Compound path*:

(a) if $\pi(x,\sigma,p)$ is a strict positive path, then $\pi(x,\sigma,p) \Rightarrow q$ is a strict positive path, $\pi(x,\sigma,p) \not\Rightarrow q$ is a strict negative path, $\pi(x,\sigma,p) \not\Leftarrow q$ is a strict negative path, $\pi(x,\sigma,p) \to q$ is a defeasible positive path, and $\pi(x,\sigma,p) \not\to q$ is a defeasible negative path;

(b) if $\bar{\pi}(x,\sigma,p)$ is a strict negative path, then $\bar{\pi}(x,\sigma,p) \Leftarrow q$ is a strict negative path;

(c) if $\pi(x,\sigma,p)$ is a defeasible positive path, then $\pi(x,\sigma,p) \Rightarrow q$ is a defeasible positive path, $\pi(x,\sigma,p) \not\Rightarrow q$ is a defeasible negative path, $\pi(x,\sigma,p) \not\Leftarrow q$ is a defeasible negative path, $\pi(x,\sigma,p) \to q$ is a defeasible positive path, and $\pi(x,\sigma,p) \not\to q$ is a defeasible negative path;

(d) if $\bar{\pi}(x,\sigma,p)$ is a defeasible negative path, then $\bar{\pi}(x,\sigma,p) \Leftarrow q$ is a defeasible negative path.

Paths represent proofs using defaults, modus ponens and contrapositive reasoning. A strict positive (respectively negative) path represents a derivation of an indefeasible conclusion. For example, a strict positive (respectively negative) path from an individual node x to a predicate node y is a proof for the conclusion "x has the property y" (respectively "x does not have the property y"). On the other hand, a defeasible positive (respectively negative) path represents a derivation of a defeasible conclusion.

Reasoning in inheritance networks is done by selecting a set of paths as a set of acceptable arguments. In the literature, the considered networks are often assumed to be acyclic and consistent [23,26,44,47–49,51]. We recall these two notions below.

The definition of acyclicity is based on the notion of generalized paths where a generalized path is either a link or a compound generalized path of one of the following the forms: $\tau \to x$, $\tau \not\to x$, $\tau \Rightarrow x$, $\tau \not\Rightarrow x$, $\tau \not\Leftarrow x$, $\tau \Leftarrow x$ where τ is a generalized path. A network Γ is *acyclic* if Γ contains neither a defeasible generalized path nor a strict positive path whose starting and end points coincide. By definition, the networks in all examples until now with the exception of Example 4.1, are acyclic.

Before introducing the definition of consistent networks, we need a couple of new notation. For any arbitrary node x of Γ, let

$$P(x) = \{x\} \cup \{y \mid \text{there exists a strict positive path from } x \text{ to } y\},$$

and

$$N(x) = \{y \mid \text{there exists a strict negative path from } x \text{ to } y\}.$$

An acyclic network Γ is *inconsistent* if there is a node x such that

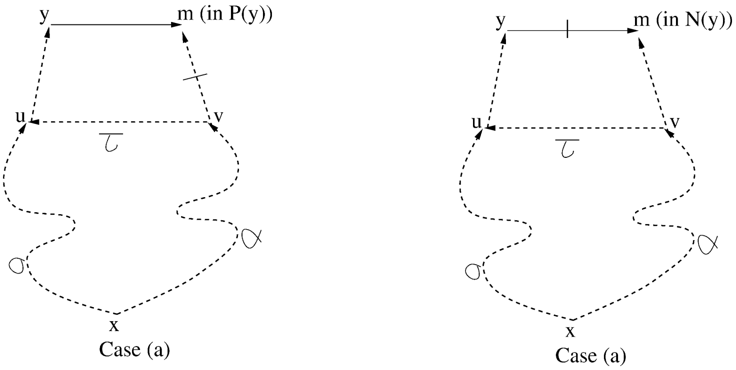

Fig. 5. $\pi(x,\sigma,u) \to y$ is preempted.

(1) $P(x) \cap N(x) \neq \emptyset$; or
(2) there are links $x \to u$ and $x \to v$ in Γ such that $v \in N(u)$; or
(3) there are links $x \to u$ and $x \not\to v$ in Γ such that $v \in P(u)$.

An acyclic network is *consistent* if it is not inconsistent.

It can be proven that if Γ is consistent and acyclic then T_Γ is well-defined and stratified.

Theorem 4.6. *For every consistent and acyclic network Γ, the default theory corresponding to Γ, T_Γ, is well-defined and stratified.*

Proof. In Appendix C. □

Each path σ can be divided into two subpaths $Str(\sigma)$ and $Def(\sigma)$ where $Str(\sigma)$ is the maximal strict end segment of σ and $Def(\sigma)$ is the defeasible initial segment of σ which is obtained by truncating $Str(\sigma)$ from σ. For instance, for $\sigma = x \Rightarrow y \to z \not\to v \Leftarrow t \Leftarrow u$ we have $Str(\sigma) = v \Leftarrow t \Leftarrow u$ and $Def(\sigma) = x \Rightarrow y \to z \not\to v$.

The semantics of an inheritance network Γ is based on the following notions.

Given a set of paths Φ, a path $\pi(x,\sigma,u) \to y$ (respectively $\pi(x,\sigma,u) \not\to y$) is *constructible* in Φ iff $\pi(x,\sigma,u) \in \Phi$ and $u \to y \in \Gamma$ (respectively $u \not\to y \in \Gamma$).

A positive path $\pi(x,\sigma,u)$ is *conflicted* in Φ iff (i) Φ contains a path of the form $\bar\pi(x,\tau,m)$ and $m \in P(u)$; or (ii) Φ contains a path of the form $\pi(x,\tau,m)$ and $m \in N(u)$.

A negative path $\bar\pi(x,\sigma,u)$ is *conflicted* in Φ iff Φ contains a path of the form $\pi(x,\tau,m)$ and $u \in P(m)$.

A defeasible positive path $\gamma = \pi(x,\sigma,u) \to y$ is *preempted* in Φ (Fig. 5) iff there exist nodes v and m such that

(i) either $v = x$ or there is a positive path of the form $\pi(x,\alpha,v,\tau,u) \in \Phi$, and
(ii) either (a) $v \not\to m \in \Gamma$, and $m \in P(y)$ or (b) $v \to m \in \Gamma$ and $m \in N(y)$.

Similarly, a defeasible negative path $\gamma = \pi(x,\sigma,u) \nrightarrow y$ is *preempted* in Φ (Fig. 6) iff there is a node v and a node m such that

(i) either $v = x$ or there is a positive path of the form $\pi(x,\alpha,v,\tau,u) \in \Phi$, and
(ii) $v \to m \in \Gamma$ and $y \in P(m)$.

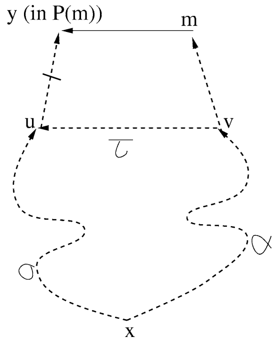

Fig. 6. $\pi(x, \sigma, u) \not\to y$ is preempted.

Definition 4.4 [23]. σ is *defeasibly inheritable* in Φ, written as $\Phi \mid\!\sim \sigma$, if one of the following condition holds:

(1) $\sigma \neq Def(\sigma)$ and $\sigma \neq Str(\sigma)$. Then, $\Phi \mid\!\sim \sigma$ iff $\Phi \mid\!\sim Def(\sigma)$ and $\Phi \mid\!\sim Str(\sigma)$.
(2) $\sigma = Str(\sigma)$. Then, $\Phi \mid\!\sim \sigma$ iff σ is a path constructed from links in Γ.
(3) $\sigma = Def(\sigma)$. Then, $\Phi \mid\!\sim \sigma$ iff either σ is a direct link or
  (a) σ is constructible in Φ, and
  (b) σ is not conflicted in Φ, and
  (c) σ is not preempted in Φ.

In the following definition, we recall the off-path credulous semantics.

Definition 4.5. Let Γ be an inheritance network.

(1) A set Φ of paths is a *credulous extension* of Γ if $\Phi = \{\sigma \mid \Phi \mid\!\sim \sigma\}$.
(2) Let a be an individual node, and p be a predicate node. We define $\Gamma \mid\!\sim_c p(a)$ (respectively $\Gamma \mid\!\sim_c \neg p(a)$) if each credulous extension of Γ contains a positive path of the form $\pi(a, \sigma, p)$ (respectively a negative path of the form $\bar{\pi}(a, \sigma, p)$).

The following theorem shows that for inheritance networks, the path-based semantics and our argumentation-theoretic semantics coincide.

Theorem 4.7. *Let Γ be an acyclic and consistent inheritance network, a be an individual node, and p be a predicate node. Then*

(1) *$\Gamma \mid\!\sim_c p(a)$ iff $T_\Gamma \mid\!\sim p(a)$, and*
(2) *$\Gamma \mid\!\sim_c \neg p(a)$ iff $T_\Gamma \mid\!\sim \neg p(a)$.*

Proof. In Appendix C. □

5. Computing $\mid\!\sim$ by translating into Reiter's default logic

In this section, we show how the newly defined entailment relation can be computed. Instead of developing new algorithms for that purpose, we will take advantage of many well-known algorithms of Reiter's default logic such as computing an extension or all extensions of a Reiter's default theory, etc. We achieve that by translating each default theory T into an equivalent Reiter's default theory R_T. In other words, the translation preserves the semantics of default theories. Moreover, the translation is modular and polynomial, i.e., R_T can be modularly constructed and has a size polynomial in the size of T. Before presenting the translation let us recall some basic notion of Reiter's default logic [30,37,42].

A *R-default* is a rule of the form

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

where α, β_i $(i = 1, \ldots, n)$, and γ are first-order formulas which are referred as the prerequisite, the justification, and the consequent of the rule, respectively.

A *R-default theory* is a pair (W, D) where W is a first-order theory and D is a set of R-defaults. The semantics of R-default theories is defined by the notion of an extension, defined as follows.

Definition 5.1. Let (W, D) be a R-default theory, S be a set of formulas. $\Gamma(S)$ is the smallest set of formulas such that

(1) $W \subseteq \Gamma(S)$;
(2) $\Gamma(S)$ is deductively closed; and
(3) if $\frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma} \in D$, $\alpha \in \Gamma(S)$, and $\neg\beta_i \notin S$, $i = 1, \ldots, n$, then $\gamma \in \Gamma(S)$.

S is called an *extension* of (W, D) if $\Gamma(S) = S$. [11]

Note that while an extension of a default theory in our framework is defined as a set of defaults, an extension of a R-default theory is a set of formulas.

We will now discuss the main characteristics of the translation. Assume that $R_T = (W_T, D_T)$ is obtained from $T = (E, B, D)$ after the translation. Since the reasoning in T relies on the evidence set E, the set of rules B, the set of defaults D, and the more specific relation $\prec$ to draw conclusions, a translation from T to R_T, that preserves the semantics of T, must address the following representational and computational issues:

- What is the role of E and B in R_T?
- How to represent a default of the form $bd(d) \rightarrow hd(d)$ in R_T?, and
- How to translate the more specific relation $\prec$ into elements of R_T?

Obviously, the first issue is easy to resolve. Since $E \cup B$ represents the first-order part of T, it is natural to make it a part of W_T, the first-order part of R_T. That is, $E \cup B \subseteq W_T$ should hold.

A default $d = bd(d) \rightarrow hd(d)$ represents the normative statement "normally, if $bd(d)$ holds then $hd(d)$ holds". Such a statement can be represented as a R-default, say r_d, with $bd(d)$ and $hd(d)$ as its prerequisite and consequent, respectively, and a justification that indicates that r_d is applicable if and only if the default d is applicable. This can be easily achieved by introducing a new propositional symbol [12] ab_d, whose truth value is identical

[11] For more on Reiter's default logic and the many algorithms for Reiter's default theories, see [30,42].

[12] Recall that T is propositional.

to the abnormality of d. I.e., if ab_d holds then the default is abnormal, and hence, r_d cannot be applied. Thus, $\neg ab_d$ can be used as the justification for r_d. So, a default d in T can be translated into a R-default:

$$\frac{bd(d) : \neg ab_d}{hd(d)} \tag{1}$$

of R_T. Furthermore, since a default is abnormal when the complement of its conclusion has been drawn, the R-default

$$\frac{\neg hd(d) : \top}{ab_d} \tag{2}$$

must be paired with (1). In other words, we represent a default d with two R-defaults (1) and (2).

An important part of the reasoning in T is the use of the more specific relation, $\prec$. Therefore, to preserve the semantics of T, the translation must preserve its more specific relation. The main obstacle for this task lies in the fact that $\prec$ is computed independently from the context E but the applicability of its elements depends on E. More precisely, for every pair of two defaults d and d', the fact that $d \prec_K d'$ for some set of defaults K is independent from E but whether d overrides d' depends on E and the applicability of defaults in K. In an early version of this paper [17], we proposed a translation of $\prec$ which relies on the fact that the applicability of a default d can be characterized by ad_d. Thus, $d \prec_K d'$ in T can be translated into the default

$$\frac{bd(d) : \bigwedge_{c \in K} \neg ab_c}{ab_{d'}} \tag{3}$$

in R_T. Intuitively, (3) means that if $bd(d)$ can be concluded and every default in K is not abnormal (or applicable) then the default d' cannot be applied. We proved that this translation indeed preserves the semantics of T in [17]. However, this translation suffers from a severe drawback in that it has a high complexity. This is because of the fact that $|\prec|$, the number of elements in the more specific relation, could be exponential on the size of T in the worst case. This can be seen in the next example, suggested by a reviewer and can also be found in [8,45].

Example 5.1. Let $T = (\{b\}, \emptyset, D)$, D consists of

$$b \to c,$$
$$d \to \neg c,$$
$$b \to b_{i1} \quad \text{for } i \in \{1,2\},$$
$$b_{ij} \to b_{i'(j+1)} \quad \text{for } i, i' \in \{1,2\} \text{ and } j \in \{1, \ldots, n-1\},$$
$$b_{in} \to d \quad \text{for } i \in \{1,2\}.$$

It is easy to verify that the cardinality of the set $\{K \mid b \to c \prec_K d \to \neg c\}$ is 2^n. Thus, while there are only $4n$ defaults in T, $\prec$ has 2^n elements.

The above example represents a real challenge to the translation. It also shows that separating the process of computing the more specific relation from the translation will probably not lead to a polynomial translation. To accomplish that goal, a better way to encode $\prec$ is needed. To this end we develop a new technique to encode the more specific relation. Instead of using the abnormal atoms, we use intermediate variables, which play a role similar to that of the variables recording the connectivity between nodes of an inheritance network in You et al. [53].

We introduce, for each default d in D, and for each atom $a \in \mathcal{L}$ such that a does not occur in the body of d, a new atom a_d. Let L_d denote the propositional language $\{a_d \mid a$ does not occur in $bd(d)\}$. Note that for two different defaults d, c in D, $L_d \cap L_c = \emptyset$.

For illustration, consider Example 2.1. Then

$$L_{d_2} = \{y_{d_2}, a_{d_2}, m_{d_2}\},$$
$$L_{d_1} = \{y_{d_1}, m_{d_1}, s_{d_1}\}, \quad \text{and}$$
$$L_{d_3} = \{y_{d_3}, a_{d_3}, m_{d_3}\}.$$

For each default d in D, define a new default theory $T_d = (\emptyset, B_d, D_d)$ as follows:

For each rule r in B, let r_d be the rule obtained from r by replacing every occurrence of an atom a in r, that does not occur in $bd(d)$, with a_d. Let $B_d = \{r_d \mid r \in B\}$. Similarly, for each default c in D, let c_d be the default obtained from c by replacing every occurrence of an atom a in c, that does not occur in $bd(d)$, with a_d. Let $D_d = \{c_d \mid c \in D\}$. For default d_1 in Example 2.1, we have that

$$B_{d_1} = \{y_{d_1} \Rightarrow a\}, \quad D_{d_1} = \left\{(d1)_{d_1} : a \to m_{d_1},\ (d2)_{d_1} : s_{d_1} \to \neg m_{d_1},\ (d3)_{d_1} : s_{d_1} \to y_{d_1}\right\}.$$

The connection between T and T_d is illustrated in the following lemma.

Lemma 5.1. *Let $K \subseteq D$. Then for each default $c \in D$, K attacks c by specificity if and only if there exists a default $d \in D$ such that following conditions are satisfied*:

(1) *$E \cup B \cup B_d \vdash_{K \cup K_d} bd(d) \wedge bd(c_d)$, where $K_d = \{e_d \mid e \in K\} \subseteq D_d$, and*

(2) *$B \cup \{hd(d), hd(c)\}$ is inconsistent.*

Proof. Follows directly from Lemma D.3 and D.4, Appendix D. □

The above lemma suggests that T can be translated into R_T as follows:

- Defaults in T and T_d are translated according to (1)–(2).
- To guarantee that whenever a default c is dismissed in T then its variant in T_d is also dismissed, the R-default
$$\frac{ab_c : \top}{ab_{c_d}} \tag{4}$$
can be used.
- For defaults d, c in D such that $B \cup \{hd(d), hd(c)\}$ is inconsistent, the following R-default
$$\frac{bd(d), bd(c_d) : \top}{ab_c} \tag{5}$$
can be used to dismiss default c. As we will later prove formally, an extension S of R_T is determined by the set of atoms ab_c, for $c \in D \cup \bigcup D_d$. Further, it also holds

that for all $c, d \in D$: $ab_c \in S$ iff $ab_{c_d} \in S$. Thus an extension S of R_T corresponds to an extension K_S of T in the following sense: $d \in K_S$ iff $ab_d \notin S$. Given an extension S, a default of the form (5) will be applied iff $E \cup B \cup B_d \vdash_{K_S \cup K_d} bd(d) \wedge bd(c_d)$ where $K_d = \{e_d \mid e \in K_S\} \subseteq D_d$.

The above translation would yield a R-default theory R_T whose size is polynomial in the size of T. But the time complexity of the translation remains problematic since it requires to check for the inconsistency of $B \cup \{hd(d), hd(c)\}$ that is an instance of the unsatisfiability problem in propositional logic that is known to be coNP-complete [18]. This problem could be avoided by introducing for each atom a in $\mathcal{L}$ a new atom a' not occurring in $\mathcal{L}$ or in any of the language L_d, and introducing for each pair of defaults $d, c \in D$, a R-default [13]

$$\frac{bd(d), bd(c_d), (B' \wedge (hd(d))' \Rightarrow \neg(hd(c))') : \top}{ab_c}, \tag{6}$$

where B' is obtained from B by replacing every occurrence of the atoms $a \in \mathcal{L}$ in each rule r in B by the corresponding atoms $a' \in \mathcal{L}'$.

Let

$$B^* = B \cup B' \cup \bigcup_{d \in D} B_d \quad \text{and} \quad D^* = D \cup \bigcup_{d \in D} D_d.$$

To summarize, T is translated into R_T as follows:

We first associate with each default d in D^* a new atom ab_d. The R-default theory R_T, that corresponds to T, is defined by

$$R_T = (E \cup B^*, D_T), \tag{7}$$

where D_T consists of defaults of the following forms

- for each default $d \in D^*$,

$$\frac{bd(d) : \neg ab_d}{hd(d)} \quad \text{and} \quad \frac{\neg hd(d) : \top}{ab_d}$$

  belong to D_T,
- for each default $d \in D$ and default $d_c \in D_c$,

$$\frac{ab_d : \top}{ab_{d_c}}$$

  belongs to D_T, and
- for each pair of defaults d and c in D, $d \neq c$,

$$\frac{bd(d), bd(c_d), (B' \wedge (hd(d))' \Rightarrow \neg(hd(c))') : \top}{ab_c}$$

  belongs to D_T.

---

[13] Notice that $hd(d)$ is a literal. Therefore $(hd(d))'$ is the literal obtained from $hd(d)$ by replacing the atom, say a, that occurs in $hd(d)$, with a'. B' stands for the conjunction of all clauses in B'.

We denote the set of defaults in the above items by $\mathit{regular}(T)$, $\mathit{equi}(T)$, and $\mathit{specific}(T)$, respectively. The next examples illustrate the translation from T to R_T.

Example 5.2. Consider the theory T in Example 3.1. For simplicity of presentation, we will write i instead of d_i in the definition of R_T. Further, d_{ij} stands for $(d_i)_{d_j}$. For convenience, we will also omit the justification $\top$ in listing the defaults of R_T.

$$L_1 = \{p_1, f_1\}, \qquad B_1 = \{p_1 \Rightarrow b\}, \qquad D_1 = \{d_{11} : b \to f_1,\ d_{21} : p_1 \to \neg f_1\},$$
$$L_2 = \{b_2, f_2\}, \qquad B_2 = \{p \Rightarrow b_2\}, \qquad D_2 = \{d_{12} : b_2 \to f_2,\ d_{22} : p \to \neg f_2\},$$

and

$$\mathcal{L}' = \{p', b', f'\}, \qquad B' = \{p' \Rightarrow b'\}.$$

Thus, $R_T = (\{p, p \Rightarrow b, p_1 \Rightarrow b, p \Rightarrow b_2, p' \Rightarrow b'\}, D_T)$ where $D_T = \mathit{regular}(T) \cup \mathit{equi}(T) \cup \mathit{specific}(T)$ and

$$\begin{aligned}\mathit{regular}(T) = {} & \left\{\frac{b : \neg ab_1}{f}, \frac{\neg f :}{ab_1}, \frac{p : \neg ab_2}{\neg f}, \frac{f :}{ab_2}\right\} \cup \\ & \left\{\frac{b : \neg ab_{11}}{f_1}, \frac{\neg f_1 :}{ab_{11}}, \frac{p_1 : \neg ab_{21}}{\neg f_1}, \frac{f_1 :}{ab_{21}}\right\} \cup \\ & \left\{\frac{b_2 : \neg ab_{12}}{f_2}, \frac{\neg f_2 :}{ab_{12}}, \frac{p : \neg ab_{22}}{\neg f_2}, \frac{f_2 :}{ab_{22}}\right\},\end{aligned}$$

$$\mathit{equi}(T) = \left\{\frac{ab_1 :}{ab_{11}}, \frac{ab_1 :}{ab_{12}}, \frac{ab_2 :}{ab_{21}}, \frac{ab_2 :}{ab_{22}}\right\},$$

and

$$\mathit{specific}(T) = \left\{\frac{p, b_2, (B' \wedge f' \Rightarrow f') :}{ab_1}, \frac{b, p_1, (B' \wedge \neg f' \Rightarrow \neg f') :}{ab_2}\right\}.$$

It is easy to see that p_1 cannot belong to any extension of R_T. On the other hand, b_2 must belong to every extension of R_T because p and $p \Rightarrow b_2$ belong to W_T. Furthermore, $B' \wedge f' \Rightarrow f'$ is a valid sentence. Thus, from the first default in $\mathit{specific}(D)$, we conclude that ab_1 belongs to every extension of R_T. This implies that R_T has only one extension $\mathit{Th}(\{p, b, \neg f, b_2, \neg f_2, ab_1, ab_{11}, ab_{12}\} \cup E \cup B^*)$, where $\mathit{Th}(X)$ denotes the logical closure of X in the language of R_T, which corresponds to the unique extension $\{d_2\}$ of T.

We show how the context E affects the applicability of defaults in R_T in the next example.

Example 5.3. Let us consider again the theories T, T' in Example 2.1. We have that,

$$L_1 = \{y_1, m_1, s_1\}, \qquad B_1 = \{y_1 \Rightarrow a\},$$
$$D_1 = \{d_{11} : a \to m_1,\ d_{21} : s_1 \to \neg m_1,\ d_{31} : s_1 \to y_1\},$$
$$L_2 = \{y_2, m_2, a_2\}, \qquad B_2 = \{y_2 \Rightarrow a_2\},$$
$$D_2 = \{d_{12} : a_2 \to m_2,\ d_{22} : s \to \neg m_2,\ d_{32} : s \to y_2\},$$
$$L_3 = \{y_3, m_3, a_3\}, \qquad B_3 = \{y_3 \Rightarrow a_3\},$$

$D_3 = \{d_{13}\colon a_3 \to m_3,\ d_{23}\colon s \to \neg m_3,\ d_{33}\colon s \to y_3\}$, and
$\mathcal{L}' = \{a', y', s', m'\}$, $B' = \{y' \Rightarrow a'\}$.

Thus, $R_T = (E \cup B^*, D_T)$ and $R_{T'} = (E' \cup B^*, D_{T'})$ where $B^* = \{y \Rightarrow a,\ y_1 \Rightarrow a,\ y_2 \Rightarrow a_2,\ y_3 \Rightarrow a_3, y' \Rightarrow a'\}$, $D_T = D_{T'} = \mathit{regular}(T) \cup \mathit{equi}(T) \cup \mathit{specific}(T)$ and

$$\begin{aligned}
\mathit{regular}(T) = {} & \left\{\frac{a:\neg ab_1}{m}, \frac{\neg m:}{ab_1}, \frac{s:\neg ab_2}{\neg m}, \frac{m:}{ab_2}, \frac{s:\neg ab_3}{y}, \frac{\neg y:}{ab_3}\right\} \cup \\
& \left\{\frac{a:\neg ab_{11}}{m_1}, \frac{\neg m_1:}{ab_{11}}, \frac{s_1:\neg ab_{21}}{\neg m_1}, \frac{m_1:}{ab_{21}}, \frac{s_1:\neg ab_{31}}{y_1}, \frac{\neg y_1:}{ab_{31}}\right\} \cup \\
& \left\{\frac{a_2:\neg ab_{12}}{m_2}, \frac{\neg m_2:}{ab_{12}}, \frac{s:\neg ab_{22}}{\neg m_2}, \frac{m_2:}{ab_{22}}, \frac{s:\neg ab_{32}}{y_2}, \frac{\neg y_2:}{ab_{32}}\right\} \cup \\
& \left\{\frac{a_3:\neg ab_{13}}{m_3}, \frac{\neg m_3:}{ab_{13}}, \frac{s:\neg ab_{23}}{\neg m_3}, \frac{m_3:}{ab_{23}}, \frac{s:\neg ab_{33}}{y_3}, \frac{\neg y_3:}{ab_{33}}\right\},
\end{aligned}$$

$$\mathit{equi}(T) = \left\{\frac{ab_1:}{ab_{11}}, \frac{ab_1:}{ab_{12}}, \frac{ab_1:}{ab_{13}}, \frac{ab_2:}{ab_{21}}, \frac{ab_2:}{ab_{22}}, \frac{ab_2:}{ab_{23}}, \frac{ab_3:}{ab_{31}}, \frac{ab_3:}{ab_{32}}, \frac{ab_3:}{ab_{33}}\right\},$$

and

$$\mathit{specific}(T) = \left\{\begin{array}{l}
\dfrac{a, s_1, (B' \wedge m' \Rightarrow m'):}{ab_2} \\
\dfrac{a, s_1, (B' \wedge m' \Rightarrow \neg y'):}{ab_3} \\
\dfrac{s, a_2, (B' \wedge \neg m' \Rightarrow \neg m'):}{ab_1} \\
\dfrac{s, (B' \wedge \neg m' \Rightarrow \neg y'):}{ab_3} \\
\dfrac{s, a_3, (B' \wedge y' \Rightarrow \neg m'):}{ab_1} \\
\dfrac{s, (B' \wedge y' \Rightarrow m'):}{ab_2}
\end{array}\right\}.$$

Consider the two cases:

(1) *Case* 1: $E = \{s, y, a\}$. We can easily check that s_1 and $\neg y_2$ cannot belong to any extension of R_T. This implies that every extension of R_T contains y_2 and a, which again implies that ab_1 belongs to every extension of R_T. Hence, R_T has only one extension

$$\begin{aligned}
& \mathit{Th}\big(\{s, y, a, \neg m, y_2, y_3, a_2, a_3, \neg m_2, \neg m_3, ab_1, ab_{11}, ab_{12}, ab_{13}, \} \\
& \quad \cup (E \cup B^*)\big),
\end{aligned}$$

which corresponds to the unique extension $\{d_2, d_3\}$ of T.

(2) *Case* 2: $E' = \{s, \neg y, a\}$. We have that ab_3 belongs to every extension of $R_{T'}$ because $\neg y$ holds. Thus, every extension of R_T will contain $\{ab_{31}, ab_{32}, ab_{33}\}$. Thus, none of the defaults in $\mathit{specific}(T)$ can be applied. This implies that $R_{T'}$ has two extensions:

$$Th\big(\{a, \neg y, s, \neg m, \neg m_2, \neg m_3, ab_1, ab_{11}, ab_{12}, ab_{13}, ab_3, ab_{31}, ab_{32}, ab_{33}\} \cup E' \cup B^*\big)$$

and

$$Th\big(\{a, \neg y, s, m, ab_2, ab_{21}, ab_{22}, ab_{23}, ab_3, ab_{31}, ab_{32}, ab_{33}\} \cup E' \cup B^*\big),$$

which correspond to the two extensions $\{d_2\}$ and $\{d_1\}$ of T', respectively.

We now prove the equivalence between T and R_T. More precisely, we will prove that for each ground literal l, $T \mathrel{|\!\sim} l$ iff l is contained in every extension of R_T.

Theorem 5.1. *Let T be a default theory and l be a ground literal. Then, $T \mathrel{|\!\sim} l$ iff l is contained in every extension of R_T.*

Proof. In Appendix D. □

It is easy to see that the translation from T to R_T is incremental in the following sense:

Theorem 5.2. *Let $T = (E, B, D)$ and $T' = (P, Q, R)$ such that $E \subseteq P$, $B \subseteq Q$ and $D \subseteq R$. Assume that $R_T = (W_T, D_T)$ and $R_{T'} = (W_{T'}, D_{T'})$. Then, $W_T \subseteq W_{T'}$ and $D_T \subseteq D_{T'}$.*

Proof. Since $B \subseteq P$ and $D \subseteq R$, we have that $B' \subseteq Q'$, $B_d \subseteq P_d$ and $D_d \subseteq R_d$ for every $d \in D$. Since $W_T = E \cup B \cup B' \cup \bigcup_{d \in D} B_d$ and $W_{T'} = P \cup Q \cup Q' \cup \bigcup_{d \in R} Q_d$, we have that $W_T \subseteq W_{T'}$. Furthermore, it is easy to see that $\mathit{regular}(T) \subseteq \mathit{regular}(T')$ and $\mathit{equi}(T) \subseteq \mathit{equi}(T')$. Furthermore, $\mathit{specific}(T) \subseteq \mathit{specific}(T')$ because of $B \subseteq Q$ and $D \subseteq R$. This completes the proof of the theorem. □

Theorem 5.2 has an important implication on the translation from default theories to Reiter's default theories. It shows that adding new facts, rules, or defaults into a default theory only introduces new propositions or defaults to its corresponding Reiter's default theory. Thus, no revision is necessary, i.e., $R_{T'}$ can be obtained from R_T by adding some new facts or rules. For example, when we add one default d to T, besides the introduction of the language L_d, we add to R_T the set of propositions B_d, the set of defaults of forms (1) and (2) representing D_d, the set of defaults of form (4) and (6) for the default d. We now show that the complexity of the translation from T to R_T is polynomial in the size of T which is characterized by $|\mathcal{L}|$, $|B|$, and $|D|$, the size of the language, the number of rules, and the number of defaults in T, respectively.

Theorem 5.3. *For a finite default theory T, the translation from T to R_T is polynomial in the size of T.*

Proof. Assume that $R_T = (W_T, D_T)$. Obviously, the complexity of the translation from T to R_T depends on the size of R_T, which again depends on three numbers: the number of atoms in the language of R_T, the number of propositions in W_T, and the number of defaults in D_T. We will show that the size of $R_T = (W_T, D_T)$ is $\mathrm{O}((|\mathcal{L}| + |B| + |D|) \times |D| \times |\mathcal{L}|)$ where $|\mathcal{L}|$, $|B|$, and $|D|$ represent the size of the language, the number of rules, and the number of defaults in T, respectively. It is easy to see that the number of atoms of the form a_d is at most $|D| \times |\mathcal{L}|$. In addition, there are $|D| \times (|D| + 1)$ abnormal atoms and at most $|\mathcal{L}| \times |B|$ atoms used in constructing B'. Thus, the size of the language of R_T is at most $|D| \times |\mathcal{L}| \times (2 + |D| + |B|)$. Since $W_T = E \cup B \cup B' \cup \bigcup_{d \in D} B_d$ and $|B| = |B'| = |B_d|$ for every $d \in D$, we have that $|W_T| = |E| + |B'| + |B| \times (|D| + 1)$ which is less than $2 \times |\mathcal{L}| + |B| \times (|D| + 2)$. Since each default in D^* generates two defaults in *regular*(T), $|\mathit{regular}(T)| = 2 \times |D| \times (|D| + 1)$. For each abnormal atom ab_d, $d \in D$, there exist $|D|$ defaults in *equi*(T). Hence, $|\mathit{equi}(T)| = |D|^2$. Furthermore, there are $|D| \times |D - 1|$ defaults in *specific*(T) since there are $|D| \times |D - 1|$ pairs of defaults in D. Thus, we have that $|D_T| \leqslant 4 \times |D| \times (|D| + 1)$. Therefore, the size of R_T is at most $|D| \times |\mathcal{L}| \times (2 + |D| + |B|) + 2 \times |\mathcal{L}| + |B| \times (|D| + 2) + 4|D| \times |D + 1| \leqslant 9|D| \times |\mathcal{L}| \times (|\mathcal{L}| + |D| + |B|)$. This implies that the size of R_T is $\mathrm{O}((|\mathcal{L}| + |B| + |D|) \times |D| \times |\mathcal{L}|)$, i.e., the translation is polynomial in the size of T. □

We conclude the section with a brief discussion on the complexity of computing $\mid\!\sim$, i.e., the complexity of the *entailment problem*:

- Given a default theory T and a literal l.
- Determine whether $T \mid\!\sim l$.

Because the entailment problem of $(E, B, \emptyset)$ is coNP-complete, and because it is polynomially decidable whether a default theory (E, B, D) has an empty set of defaults D, the entailment problem is coNP-hard.

6. Related work

Approaches to reasoning with specificity differ from each others in two aspects. One is how specificity information is obtained and the other is how this information can be used to eliminate unintended models and to resolve conflicts. Specificity information can be extracted from the theory (or *implicit specific knowledge*) or obtained from users (or *explicit specific knowledge*). It is often the case that only one source of specificity information is used. However, in all of these approaches, the specificity principle is the only principle used for conflict resolution and discarding unintended models. We note that some authors (see, e.g., Vreeswijk [52]) have argued that there are situations in which the specificity principle might not necessarily be the only principle that can be used. Even though this maybe true, we will concentrate on comparing our work with others that advocate the specificity principle. We have discussed the shortcoming of previous approaches in their treatment of specificity in Section 2. We now compare our approach with some earlier work on reasoning with specificity in more detail. In particular, we distinguish the current work with our early work [15] and some of the close related work such as condition entailment of Geffner and Pearl [20] and the approach of Simari and Loui

[46]. We choose to do so since both approaches use implicit specificity information and are argument-based. We then compare our specificity relation in this paper with Z-ordering, a well-known specificity ordering introduced by Pearl [38]. Finally, we compare our translation of default theories into Reiter's default theories with Delgrande and Schaub's translation of default theories into Reiter's default theories.

6.1. Our early work

Our current work is a continuation and improvement of our own work in [15]. Throughout the paper, we have mentioned the differences and similarities between the two approaches. We now discuss the major differences and similarities between them in more details.

- The current approach is more general than its predecessor in the sense that default theories in [15] are special cases of default theories considered in this paper. There, we consider only acyclic and consistent default theories without rules (or ground clauses). In this paper, we lift all these restrictions and consider more general default theories, which can have rules or even cycles in their atom dependency graphs. The technical framework developed in [15] cannot be applied for the general cases.
- The "more specific" relation in this paper is much simpler than its counterpart in [15]. It is a faithfully generalization of Touretzky's specificity principle in inheritance reasoning to more general default theories—it is defined in a single definition (Definition 3.3). On the other hand, its counterpart in [15] is given by a series of five definitions (Definitions 4.6–4.10 of [15]), in which *minimal conflict set* (MCS), *conflicted defaults*, *more specific default*, *most specific default*, and *specific relevant MCS* are defined. These definitions provide an adequate framework for the class of default theories considered in our early paper but they cannot be easily extended to more general default theories. In hindsight, we would say that the definitions in [15] are unnecessary complicated.

  We note that the more specific relation in this paper is not the same as the more specific relation in [15]: in the Nixon diamond example (Fig. 3), d_4 is more specific than d_2 (with respect to the MCS $\{d_1, d_2, d_3, d_4\}$) in [15] while the more specific relation in this paper will yield an empty set.
- Both approaches are argumentation-based, i.e., both employ the principles or argumentation in defining the semantics of default theories. In [15], each default theory is translated into an argumentation framework and the semantics of a default theory is defined by the preferred semantics of its corresponding argumentation framework. On the other hand, the current approach does not employ an explicit notion of arguments. Its attacks relation is defined between a set of defaults and a default. This semantics is a type of stable semantics of argumentation. As such, in this paper we face the problem of existence of extensions that does not occur in [15].
- In this paper, we give a polynomial translation from default theories into Reiter's default theories. The translation from default theories into logic programs presented in [15] is exponential in the worst case.
- Having stated the major differences between the two approaches, we now present some informal results about the connection between the two papers. Readers, who are

not interested in technical details, might want to skip this paragraph. For the rest of this subsection, by a default theory we mean a consistent and acyclic default theory, which satisfies the Definitions 4.1, 4.3, and 4.4 of [15].

Let $(E, \emptyset, D)$ be default theory. Furthermore, let d and d' be defaults in D and K be a minimal set of defaults such that $d \prec_K d'$, i.e., there exists no $K' \subset K$ such that $d \prec_{K'} d'$. Then, we can prove that

(1) $C = \{d, d'\} \cup K$ is a MCS with respect to d,
(2) d and d' are two conflicted defaults of C,
(3) d is the most specific default of C, and
(4) C is a specific relevant MCS with respect to d.

On the other hand, let C be a specific relevant MCS (with respect to d), d be a most specific default in C, and d' be the other conflicted default in C. Then, $d \prec_K d'$.

It can be shown that the newly defined entailment relation subsumes the old ones by showing that for every default theory $T = (E, \emptyset, D)$, if S is an extension of T (with respect to the new approach) then $Arg(S) = \{A \mid A \subseteq S,\ A$ is an argument (with respect to the old approach)$\}$ is a stable extension of AF_T, the corresponding argumentation framework of T.

6.2. Conditional entailment

In this subsection, we compare our approach with conditional entailment, a prominent approach to reasoning with specificity introduced by Geffner and Pearl [20]. In their paper, after discussing the pros (dealing with irrelevant evidence and the general properties of the entailment relation) and cons of extensional and conditional approaches to reasoning with specificity, Geffner and Pearl wrote:

> ... "The question arises whether a unifying framework can be developed which combines the virtues of both the extensional and conditional interpretations." ...

Conditional entailment does indeed express the best features of the ε-entailment of the conditional approaches and the p-entailment of the extensional approaches: it can deal with irrelevant evidence and it satisfies many desirable properties of nonmonotonic consequence relations such as deduction, reduction, conditioning, cumulativity, and disjunction.

In conditional entailment, each default schema $p(x) \to q(x)$ is encoded by a sentence $p(x) \wedge \delta_i(x) \Rightarrow q(x)$ and a default schema $p(x) \to \delta_i(x)$ where δ_i denotes a new and unique assumption predicate which summarizes the normality conditions required for concluding $q(x)$ from $p(x)$. Hence default theories in their formalization are called *assumption-based default theories*.

An irreflexive and transitive priority order $\prec$ over the set of assumptions of a default theory is *admissible* if for every default δ and a set of assumptions Δ that is logically inconsistent with δ in the context $\{p\}$, i.e., $\{p\} \cup B \cup \Delta \vdash \neg\delta$, there exists one assumption $\delta' \in \Delta$ such that $\delta' \prec \delta$, i.e., δ' has lower priority than δ. Preferred models are then defined with respect to admissible priority orders similar to what has been described in Section 1. Finally, a conclusion q is conditional entailed by a theory T if q holds in very preferred model of T.

We now list the differences and similarities between our formalism and Geffner and Pearl's conditional entailment.

- We demonstrate the difference by using a default theory in Example 2.1. It is easy to see that the ordering between assumptions requires that $\{\delta_1, \delta_3\} \prec \delta_2$ and $\{\delta_1, \delta_2\} \prec \delta_3$. This implies that any admissible priority ordering must satisfy $\delta_1 \prec \delta_2$ and $\delta_1 \prec \delta_3$. Furthermore, the context $E = \{a, s, \neg y\}$ gives rise to two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of minimal models M_1 and M_2 with the gaps $\Delta[M_1] = \{\delta_1, \delta_3\}$ and $\Delta[M_2] = \{\delta_2, \delta_3\}$, respectively. The priority ordering implies that $\mathcal{C}_1$ is the preferred class of models, i.e., $\neg m$ is supported in conditional entailment in the context $E = \{a, s, \neg y\}$. This shows that conditional entailment treats priority between defaults *unconditional*. It follows from Example 2.1 that conditional entailment cannot capture inheritance reasoning. Both of these points distinguish our approach from conditional entailment.
- δ_i—in their encoding of defaults—plays the role of $\neg ab_i$ in our translation from default theories into Reiter's default theories. Both approaches rely on an implicit priority ordering between defaults to resolve conflicts. The proof theory of conditional entailment is defined around the notion of arguments. Each argument is a set of assumptions which is consistent. In our formalization, we do not have an explicit notion of arguments as it is not necessary for our purpose.
- One important feature of conditional entailment, that distinguishes conditional entailment from other approaches, is that it can deal with irrelevant evidence. We next demonstrate, using an example given in [20], that our approach can also deal with irrelevant evidence correctly.

Example 6.1 (*Dealing with irrelevant evidence* [20]). Consider the default theories $T = (E, \emptyset, D)$ and $T' = (E', \emptyset, D)$ with

$$D = \{d_1 : b \to f,\ d_2 : p \to \neg f,\ d_3 : p \to b,\ d_4 : r \to b\},$$

$E = \{r\}$ and $E' = \{r, p\}$. The defaults are depicted in Fig. 7.
It is easy to see that $d_2 \prec_{\{d_3\}} d_1$. The priority order of conditional entailment requires that $\delta_1 \prec \delta_2$ and $\delta_1 \prec \delta_3$.

– $E = \{r\}$. Since default d_3 is not applicable, T has only one extension $\{d_1, d_3, d_4\}$. This corresponds to the class of preferred models supporting b and f of conditional entailment.

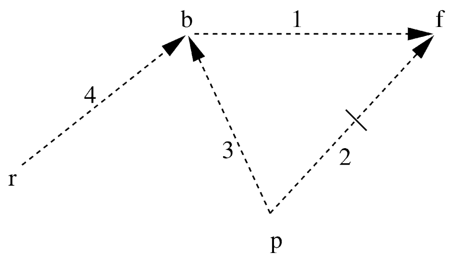

Fig. 7. Red birds do fly.

– $E' = \{r, p\}$. Obviously, any extension of T' must contain d_3 and d_4 since there exists no attack against them. Furthermore, $\{d_1, d_2, d_3, d_4\}$ attacks itself by conflict (also by specificity). Thus, there are two possible extensions of $T' : H_1 = \{d_4, d_3, d_2\}$ and $H_2 = \{d_4, d_3, d_1\}$. Since there exists no default, which is more specific than d for $d \in H_1$, and H_1 attacks d_1 by conflict, we conclude that H_1 is an extension of T'. On the other hand, H_2 attacks itself by specificity since $d_1 \in H_2$ and $d_1 \prec_{\{d_3\}} d_2$ and $E' \vdash_{H_2} p = bd(d_3)$. Hence, the only extension of T' is H_1 which yields $\neg f$ and b. These are also the conclusions sanctioned by conditional entailment.

- Another important difference between conditional entailment and the entailment relation $\mid\!\sim$ defined in Section 3 lies in the fact that conditional entailment satisfies conditioning and cumulativity and $\mid\!\sim$ does not. Even though we agree that conditioning and cumulativity are important properties of nonmonotonic consequence relations, we are not sure if they should always be enforced. Given a theory $T = (\{p\}, \emptyset, \{p \to q, p \to \neg q\})$, neither q nor $\neg q$ is concluded in our approach but both will be concluded in conditional entailment.[14] This, together with the cumulativity property, implies that we should conclude $\neg q$ given the default theory $T + q = (\{p, q\}, \emptyset, \{p \to q, p \to \neg q\})$. This seems to contradict the common understanding about defaults that says that a default can be applied to derive new conclusions if no contrary information is available. In this case, the default $p \to \neg q$ can be used to derive new conclusion $(\neg q)$ only if no information contrary to $\neg q$ is available. As such, instead of enforcing the two properties, we characterize situations when they hold.
- Finally, we note that even though formulas are not allowed in the head of a default in our formalization, a default of the form $p \to q$ where p and q are propositional formulas can be easily encoded in our formalization by
  (i) introducing two new atoms p' and q',
  (ii) replacing $p \to q$ with $p' \to q'$, and
  (iii) adding the clauses $p \Leftrightarrow p'$ and $q \Leftrightarrow q'$ to B.
  Thus, the class of default theories considered in conditional entailment and in our approach is the same.

6.3. *Simari and Loui's approach*

The goal of Simari and Loui [46] is to develop a general framework that unifies different argument-based approaches to defeasible reasoning. They want to achieve this goal by defining a framework that combines the best ideas of two well-known approaches to defeasible reasoning: Poole's [40] (a comparative measure of the relevance of information) and Pollock's [39] (the interaction between arguments).

The language for knowledge representation in Simari and Loui's approach is a first-order language $\mathcal{L}$ plus a metalinguistic relation between non-grounded well-formed formulas, denoted by $\succ\!\!-$, which represents *defeasible rules*. For example, $\alpha \succ\!\!- \beta$ means that

[14] It is easy to check that there is no admissible priority order for defaults in T and hence there exists no preferred model of T. We took the view that in this case conditional entailment entails every possible conclusions of the theory.

"reasons to believe in α provide reasons to believe in β. Thus the defeasible rules are the defaults in our representation. A theory is represented by a pair $(\mathcal{K}, \Delta)$ where $\mathcal{K}$, called *context*, is a set of sentences in $\mathcal{L}$ and Δ is a finite set of defeasible rules. $\mathcal{K}$ is further divided into two sets: the set of grounded sentences $\mathcal{K}_C$ and the set of non-grounded sentences $\mathcal{K}_N$. For brevity, we omit here the definitions of a defeasible derivation and defeasible consequence $\mid\!\sim$ of a set of ground instances of sentences in $\mathcal{K} \cup \Delta$ as they are fairly close to our Definition 3.2. An argument A for a conclusion h, written $\langle A, h\rangle$, [15] is a subset of the set of ground instances of defeasible rules $\Delta^{\downarrow}$ that satisfies the following conditions:

(1) $\mathcal{K} \cup A \mid\!\sim h$,
(2) $\mathcal{K} \cup A \not\mid\!\sim \bot$, and
(3) there exists no argument $A' \subset A$ such that $\mathcal{K} \cup A' \mid\!\sim h$.

An argument $\langle A_1, h_1\rangle$ is a subargument of $\langle A, h\rangle$ if $A_1 \subseteq A$. Two comparative measures between arguments are defined.

- $\langle A_1, h_1\rangle$ is *strictly more specific* than $\langle A_2, h_2\rangle$, denoted by $\langle A_1, h_1\rangle \prec_{spec} \langle A_2, h_2\rangle$, if
  - for each ground sentence e in $\mathcal{L}$ such that $\mathcal{K}_N \cup \{e\} \cup A_1 \mid\!\sim h_1$ and $\mathcal{K}_N \cup \{e\} \not\mid\!\sim h_1$, then $\mathcal{K}_N \cup \{e\} \cup A_2 \mid\!\sim h_2$, and
  - there exists a grounded sentence e in $\mathcal{L}$ such that (i) $\mathcal{K}_N \cup \{e\} \cup A_2 \mid\!\sim h_2$, (ii) $\mathcal{K}_N \cup \{e\} \cup A_1 \not\mid\!\sim h_1$, and (iii) $\mathcal{K}_N \cup \{e\} \not\mid\!\sim h_2$.
- $\langle A_1, h_1\rangle$ and $\langle A_2, h_2\rangle$ are *equi-specific*, denoted by $\langle A_1, h_1\rangle \equiv_{spec} \langle A_2, h_2\rangle$, if
  - for each ground sentence e in $\mathcal{L}$, $\mathcal{K}_N \cup \{e\} \cup A_1 \mid\!\sim h_1$ if and only if $\mathcal{K}_N \cup \{e\} \cup A_2 \mid\!\sim h_2$.

The two specificity relations are used to define the *counterargument* and *defeat* relations between arguments, which are used to draw the (defeasible) conclusions of the theory.

We now list some similarities and differences between Simari and Loui's approach and ours.

- The strictly more specific relation is defined between arbitrary arguments. The first condition of the definition of $\prec_{spec}$ is similar but stronger than condition (ii) in our definition of the more specific relation (Definition 3.3). Furthermore, our more specific relation is defined only between conflicting defaults.
- The entailment relation defined in Simari and Loui's paper does not satisfy the cumulativity property, even for stratified default theories. To see why, consider a modification of the famous penguin-bird example in which the implication $p(x) \supset b(x)$ is replaced by a defeasible rule, i.e., we have a theory with the context $\mathcal{K} = \{p(a)\}$ and the set of defeasible rules $\Delta = \{p(x) \succ\!\!- b(x),\ b(x) \succ\!\!- f(x),\ p(x) \succ\!\!- \neg f(x)\}$. It is easy to see that $\langle\{p(a) \succ\!\!- \neg f(a)\}, \neg f(a)\rangle$ is strictly more specific than $\langle\{p(a) \succ\!\!- b(a),\ b(a) \succ\!\!- f(a)\}, f(a)\rangle$. Therefore, the theory entails $\neg f(a)$. Furthermore, no argument is in conflict with $\langle\{p(a) \succ\!\!- b(a)\},\ b(a)\rangle$. So, the theory entails $b(a)$ too. However, the theory $(\mathcal{K} \cup \{b(a)\}, \Delta)$ does not entail $\neg f(a)$, nor does it entail $f(a)$ (see Example 6.5 [46]).
- Simari and Loui's definition of an argument in [46] requires that an argument is minimal with respect to the set inclusion. One consequence of this requirement is that adding a new fact to a theory might eliminate some existing arguments, thus altering

[15] In [46], $\langle A, h\rangle$ is called a *argument structure*. We follow [41] and call it an argument for convenience.

the specificity relations and the ordering between arguments. In the above example, adding $b(a)$ to $(\mathcal{K}, \Delta)$ removes the argument $\langle\{p(a) \succ\!\!- b(a), b(a) \succ\!\!- f(a)\}, f(a)\rangle$ from $\Delta^{\downarrow}$ and introduces a new one, $\langle\{b(a) \succ\!\!- f(a)\}, f(a)\rangle$.

6.4. System Z and its use in Delgrande and Schaub's approach

In System Z [38], Pearl uses consistency check to determine the order of a default. The lower the order of a default is, the higher is its priority. He only considered theories whose background knowledge is empty, i.e., default theories without rules. In this respect, System Z is closely related to our previous work [15] than this one. A default is of the form $\alpha \to \beta$, where α and β are propositional formulas. For convenience, a default r is often used interchangeable with $\alpha_r \to \beta_r$, when no confusion is possible.

Let R be a set of defaults. A default $\alpha \to \beta$ is tolerated by R if $\{\alpha \wedge \beta\} \cup \{\alpha_r \supset \beta_r \mid r \in R\}$ is satisfiable.

A set of defaults R is *Z-consistent* if for every nonempty $R' \subseteq R$, some $r' \in R'$ is tolerated by R'. A set of defaults R is partitioned into an ordered list of mutually exclusive sets of rules $R_0, R_1, \ldots, R_n$, called *Z-ordering* on R, in the following way:

(1) Find all defaults tolerated by R, and call this subset R_0.
(2) Next, find all defaults tolerated by $R \setminus R_0$, and call this subset R_1.
(3) Continue in this fashion until all defaults in R have been accounted for.

It is easy to see that

$$R_i = \{r \mid r \text{ is tolerated by } R \setminus (R_0 \cup \cdots \cup R_{i-1})\}$$

for $1 \leqslant i \leqslant n$. R is said to have a non-trivial Z-ordering if $n > 0$. Otherwise, it has a trivial Z-ordering. For $i < j$, defaults in R_i are considered less specific than defaults in R_j. This order is used to define the rank of an interpretation of R, the rank of a formula, and the 1-entailment. Since the weakness of System Z has been discussed in [8], we will not compare the entailment relation $\mid\!\sim$ with Pearl's 1-entailment. Instead we will compare our approach with the approach of Delgrande and Schaub which exploits the Z-ordering but overcomes its weakness. Delgrande and Schaub [8] showed that sometimes Z-order introduces unwanted priority and cannot deal with irrelevant knowledge. However, the Z-ordering shares some of the properties of our specificity relation such as

- it is defined independently from the context (E),
- it is unique, and
- it is monotonic with respect to the addition of new defaults.

In [8], Delgrande and Schaub showed how the Z-ordering can be used to deal with specificity. They improved it by *not using it directly* but for the purpose of finding minimal conflict sets (MCS). They also extended it to work with rules. In their notation, a default theory (E, B, D) is called an *entire world description* of which (D, B) is called a *generic world description*. Rules in B are given in the form $\alpha \supset \beta$.

For a default theory $T = (E, B, D)$, the Z-ordering of T is the ordering of the set of defaults $R = D \cup \{\alpha \to \beta \mid \alpha \supset \beta \in B\}$. That is, in determining the Z-ordering of defaults of T, Delgrande and Schaub considered rules as defaults.

Let $R = (D, B)$ be a world description. $C \subseteq R$ is a *minimal conflict set* in R iff C has a nontrivial Z-ordering (C_0, C_1) and any $C' \subset C$ has a trivial Z-ordering.

Delgrande and Schaub proved a number of important properties of MCS's. To resolve the conflict they identify the least specific defaults in a MCS and falsify some of them by defining the *conflicting core* of a MCS.

Let $R = (D, B)$ be a world description and $C \subseteq R$ a MCS with the Z-ordering (C_0, C_1). A conflicting core of C is a pair of least sets $(\min(C), \max(C))$ where

(1) $\min(C) \subseteq C_0 \cap D$,
(2) $\max(C) \subseteq C_1 \cap D$, and
(3) $\{\alpha_r \wedge \beta_r \mid r \in \max(C) \cup \min(C)\} \models \bot$

provided that $\min(C)$ and $\max(C)$ are nonempty.

They use this to convert a default theory into a Reiter's default theory whose semantics specifies the semantics of the original theory. The translation is similar to our translation (Section 5) but has also some differences due to the differences in the specificity relation and in our treatment of defaults. For example,

- Both translations use only information about defaults and specificity information to create defaults of the destination theory.
- They do not introduce the literal $ab(d)$ for each default d as we do.
- For each default $\alpha \to \beta$, their translation produces only one default (in Reiter's sense) in the destination theory whose prerequisite encodes the applicability condition of higher priority defaults; thus making the default applicable only when none of the higher priority defaults is applicable (later, we demonstrate this in an example). This makes the translation *not modular*: when adding a default that introduces some new MCS, some defaults must be revised. On the other hand, our translation is modular: none of the previous defaults needs to be revised. Also, our translation converts each default $\alpha \to \beta$ into two defaults and each element of the specificity relation into one default.
- White try to enforce the order between defaults they consider defaults as rules. For example, if r has higher priority than r', then the prerequisite of the Reiter's default corresponding to r' contains a conjunction $\alpha_r \supset \beta_r$.
- We show that our approach captures inheritance reasoning. Delgrande and Schaub did not compare their approach with inheritance reasoning. They wrote [8, p. 306],

  > ... "Lastly there are *direct* or *path-based* approaches to nonmonotonic inheritance, as expressed using inheritance networks [23]. It is difficult to compare such approaches with our own for two reasons."...

  We show now by example that their approach does not capture inheritance reasoning. We continue with the default theory in Example 2.1. In their notation, we have that $R = (D, B)$ is a world description with $B = \{y \Rightarrow a\}$, and $D = \{d_1 : a \to m,\ d_2 : s \to \neg m,\ d_3 : s \to y\}$. The Z-ordering of R is $(\{a \to m,\ y \to a\}, \{s \to y,\ s \to \neg m\})$. Furthermore, R is a minimal conflict set. Its only conflicting core is $(\{a \to m\}, \{s \to \neg m\})$. In Example 5.3, we present the Reiter's default theories corresponding to the theory T and T' of the Example 2.1 already. In Delgrande and Schaub's translation, a default $\alpha_r \to \beta_r$ is translated into

  $$\frac{\alpha_r : \beta_r \wedge \bigwedge_{r' \in R_r} \alpha_{r'} \supset \beta_{r'}}{\beta_r},$$

where $R_r = \{r' \in \max(C^i) \mid r \in \min(C^i)\}$ and $(C^i)_{i \in I}$ is the family of all MCS of D. Thus, it yields the default theory $DT = (E \cup B, D')$ where

$$D' = \left\{ \frac{s : y}{y}, \ \frac{s : \neg m}{\neg m}, \ \frac{a : m \wedge \neg s}{m} \right\},$$

where the last default is obtained from the default $\frac{a:m \wedge s \supset \neg m}{m}$. As such, for $E = \{s\}$ or $E = \{s, \neg y, a\}$, Z-default theories will conclude $\neg m$. This also shows that priority between defaults is used *unconditional* in Delgrande and Schaub framework.

7. Conclusion and future work

In this paper we present a new approach to reasoning with specificity which subsumes inheritance reasoning. We show that priorities between defaults can be computed a priory but cannot be used unconditional. We generalize Touretzky's principle to specificity to define a "more specific" relation among defaults and use the stable semantics of argumentation to define the semantics of default theories. We present sufficient conditions for the existence of extensions. We identify a class of stratified and well-defined default theories, in which the newly defined entailment relation satisfies the basic properties of nonmonotonic consequence relations such as deduction, reduction, conditioning, and cumulativity. To show how well-known algorithms for computing extensions and consequences of Reiter's default theories can be used to compute extensions and consequences of default theories as defined here, we translate each default theory into a semantically equivalent Reiter's default theory. We prove that the translation is modular and polynomial in the size of the original default theory.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments that help us to improve the paper in many ways. A part of this manuscript appeared in [16]. We would also like to thank Yves Moinard for his valuable comments and discussion on the topic of this paper and for his support in sending us his paper [35].

Appendix A. Proofs of Theorems 4.1–4.4

A.1. Stratification guarantees existence of extensions

Let *rank* be a ranking function of the literals. We can extend *rank* on the set of clauses and defaults in T by defining:

- $rank(c) = \max\{rank(l) \mid l \text{ appears in } c\}$, for every clause c; and
- $rank(d) = rank(hd(d))$, for every default d.

For every set of literals, clauses, or defaults X, define $X|_i = \{x \in X \mid rank(x) = i\}$, and $X\|_i = \{x \in X \mid rank(x) \leqslant i\}$. Further define $T_i = (E\|_i, B\|_i, D\|_i)$.

Lemma A.1. *Let T be a stratified default theory. For all i, all $K \subseteq D$, and $l \in lit(\mathcal{L})$ with $rank(l) = i$:*

$$E \cup B \vdash_K l \quad \textit{iff} \quad E\|_i \cup B\|_i \vdash_{K\|_i} l.$$

Proof. The if-direction is trivial. We only need to prove the only-if-direction. There are two cases:

Case 1: $K = \emptyset$, i.e., $E \cup B \vdash l$. In this case the lemma follows immediately from the fact that a set of positive ground literals M is a model of $E \cup B$ iff for each i, $M\|_i$ is a model of $E\|_i \cup B\|_i$.

Case 2: There exists a sequence of defaults $d_1, \ldots, d_m$ $(m \geqslant 1)$ in K such that

(1) $E \cup B \vdash bd(d_1)$,

(2) $E \cup B \cup \{hd(d_1), \ldots, hd(d_j)\} \vdash bd(d_{j+1})$ for $j \in \{1, \ldots, m-1\}$, and

(3) $E \cup B \cup \{hd(d_1), \ldots, hd(d_m)\} \vdash l$.

Without loss of generality, we can assume that $d_1, \ldots, d_m$ is one of the shortest defeasible derivations of l, where the length of a defeasible derivation is defined as the number of defaults appearing in it. We want to show that there is no default in this derivation whose rank is greater than i. Assume the contrary, i.e., there exists some defaults in $\{d_1, \ldots, d_m\}$ whose rank is greater than i. Let $k = \max\{j \mid 1 \leqslant j \leqslant m,\ rank(d_j) > i\}$. Therefore, for each $j > k$, $rank(d_j) \leqslant i$. Hence from Case 1, it is easy to see that for each $j > k$, $E \cup B \cup \{hd(d_t) \mid t < j \text{ and } t \neq k\} \vdash bd(d_j)$. Thus $d_1, \ldots, d_{k-1}, d_{k+1}, \ldots, d_m$ is a defeasible derivation of l. This contradicts our assumption that $d_1, \ldots, d_m$ is a shortest defeasible derivation of l.

Thus we have proved that $E\|_i \cup B\|_i \vdash_{K\|_i} l$. □

The following lemma follows immediately from Lemma A.1.

Lemma A.2. *Let T be a stratified default theory, $S \subseteq D$, and $d \in D|_i$. Then,*

(1) *S attacks d by conflict in T iff $S\|_i$ attacks d by conflict in T_i, and*

(2) *S attacks d by specificity in T iff $S\|_i$ attacks d by specificity in T_i.*

Lemma A.2 implies the following lemma.

Lemma A.3. *Let T be a stratified default theory. $S \subseteq D$ is an extension of T iff for each $i \geqslant 1$, $S\|_i$ is an extension of T_i.*

In the following lemma, we give an algorithm to construct an extension of T_i from an extension of T_{i-1}.

Lemma A.4. *Let T be a stratified default theory. Let $K \subseteq D\|_{i-1}$.*

- *Let C denote the set of all defaults in $D|_i$ which are not attacked by specificity by K.*
- *Let $C_0, C_1 \subseteq C$ such that*

$$C_1 = \big\{c \in C \mid E \cup B \vdash_K bd(c)\big\} \quad \textit{and} \quad C_0 = C \setminus C_1.$$

- *Let H be a maximal* (*with respect to set-inclusion*) *set of defaults such that $H \subseteq C_1$, and $K \cup H$ is consistent in T* (*or equivalently in T_i*).
- *Let $G = \{c \in C_0 \mid E \cup B \not\vdash_{H \cup K} \neg hd(c)\}$, i.e., G consists of those defaults in C_0 which are not attacked by $K \cup H$ by conflict.*

Then $K \cup H \cup G$ is an extension of T_i iff K is an extension of T_{i-1}.

Proof. Let $S = K \cup H \cup G$.

• *Only-If-direction*. It is obvious that $S\|_{i-1} = K$. Because S is an extension of T_i, it is clear from Lemma A.3, that K is an extension of T_{i-1}.

• *If-direction*. From Lemma A.1, it follows that

$$\text{for each } d \in D|_i,\ E \cup B \vdash_K bd(d) \text{ iff } E \cup B \vdash_{K \cup H} bd(d) \text{ iff } E \cup B \vdash_S bd(d). \qquad (*)$$

We prove first that S attacks every default $d \in D\|_i \setminus S$. If $rank(d) < i$ then it is clear that K attacks d. Hence Lemma A.2 implies that S attacks d. Let now $rank(d) = i$. Then there are two cases:

• $d \notin C$. Then d is attacked by K by specificity. Hence d is attacked by S by specificity.

• $d \in C$. Therefore either $d \in C_1 \setminus H$ or $d \in C_0 \setminus G$. Let $d \in C_1 \setminus H$. Then $K \cup H \cup \{d\}$ is inconsistent in T. Since $E \cup B \vdash_K bd(d)$, $E \cup B \vdash_{K \cup H} \neg hd(d)$. Hence S attacks d by conflict.

Let $d \in C_0 \setminus G$. Then from the definition of G, it follows that S attacks d by conflict.

Now we want to prove that S does not attack itself. Assume the contrary. Lemma A.2 implies that S attacks some $c \in D|_i$. Suppose that S attacks c by specificity, i.e., there exists c' such that $c' \prec_S c$, $E \cup B \vdash_S bd(c')$. Further, from Lemma A.2, $E\|_{i-1} \cup B\|_{i-1} \vdash_{S\|_{i-1}} bd(c')$. Lemma A.2 also implies that $bd(c') \cup B\|_{i-1} \vdash_{S\|_{i-1}} bd(c)$. Hence $c' \prec_K c$. Therefore K attacks c by specificity. This contradicts the construction of H and G. Hence S must attack c by conflict. That means $E \cup B \vdash_S \neg hd(c)$. From the construction of S, we can see that S is consistent. Therefore, $c \in G$. But then from the assertion $(*)$, we have $E \cup B \vdash_{K \cup H} \neg hd(c)$. Again, this contradicts the construction of G. Thus S cannot attack itself. □

We now prove Theorem 4.1.

Theorem 4.1. *Every stratified default theory $T = (E, B, D)$ has at least one extension*

Proof. Since $rank(d) > 0$ for every $d \in D$, we have that $T_0 = (E_0, B_0, \emptyset)$. Furthermore, $E \cup B$ is consistent implies that $E_0 \cup B_0$ is consistent. Thus, $S_0 = \emptyset$ is an extension of T_0. This, together with Lemma A.4, proves that T has an extension. □

A.2. Stratification guarantees cumulativity

The following lemma is a key step in our proof of the cumulativity property.

Lemma A.5. *Let $T = (E, B, D)$ be a stratified default theory, l be a literal l such that $T \mid\!\sim l$, and $S \subseteq D$ be an extension of $T + l$. Then S is also an extension of T.*

Proof. Let $rank(l) = i$. We want to show that for each j, $S\|_j$ is an extension of T_j. There are three cases:

(1) $j < i$. Obviously because $T_j = (T + l)_j$.

(2) $j = i$. Let $K = S\|_{i-1}$ and C, C_0, C_1, H, G be defined as in Lemma A.4 with respect to $T + l$. Further let C', C'_0, C'_1 be defined as in Lemma A.4 with respect to T. It is not difficult to see that $C = C'$, $C_1 = C'_1$, and $C_0 = C'_0$. It is also obvious that $K \cup H$ is consistent in T. Let H' be a maximal (with respect to set-inclusion) set of defaults such that

- $H \subseteq H' \subseteq C_1$, and
- $K \cup H'$ is consistent in T.

Let $G' = \{c \in C_0 \mid E \cup B \not\vdash_{K \cup H'} \neg hd(c)\}$.

From Lemma A.4, it follows that $R = K \cup H' \cup G'$ is an extension of T_i. From the assumption $T \mathrel{|\!\sim} l$, it follows that R is also an extension of $(T + l)_i$. From the definition of G', it is easy to see that for each literal h, $E \cup B \vdash_R h$ iff $E \cup B \vdash_{K \cup H'} h$. It is clear that $T_i \mathrel{|\!\sim} l$. Hence $E \cup B \vdash_R l$. Therefore $E \cup B \vdash_{K \cup H'} l$. Hence $K \cup H'$ is consistent in $T + l$. From the definition of H, it follows that $H = H'$. That means that $S\|_i = R$. So $S\|_i$ is also an extension of T.

(3) $j > i$. From Case 2, it is clear that for each $j \geqslant i$, $E\|_j \cup B\|_j \vdash_{S\|_j} l$. Therefore for each literal h, $E\|_j \cup \{l\} \cup B\|_j \vdash_{S\|_j} h$ iff $E\|_j \cup B\|_j \vdash_{S\|_j} h$. Hence it is obvious that $S\|_j$ is an extension of T_j. $\square$

We are now ready to prove Theorem 4.4.

Theorem 4.4. *Let $T = (E, B, D)$ be a stratified default theory and a, b be literals such that $T \mathrel{|\!\sim} a$, and $T \mathrel{|\!\sim} b$. Then $T + a \mathrel{|\!\sim} b$.*

Proof. Assume the contrary, $T + a \not\mathrel{|\!\sim} b$. This means that there exists an extension S of $T + a$ such that $b \notin S$. Lemma A.5 shows that S is an extension of T, which contradicts the fact that $T \mathrel{|\!\sim} b$. So, our assumption is incorrect, i.e., for every extension S of $T + a$, $b \in S$. This means that $T + a \mathrel{|\!\sim} b$. $\square$

Appendix B. Conditioning of well-defined default theories

Let $csq(K) = \{l \mid E \cup B \vdash_K l\}$.

Theorem 4.5. *Let $T = (E, B, D)$ be a well-defined default theory, d be a default in D, and $E = bd(d)$. Then, $T \mathrel{|\!\sim} hd(d)$.*

Proof. Let S be an extension of T. We need to prove that $hd(d) \in csq(S)$. Assume the contrary that $hd(d) \notin csq(S)$. Since $bd(d) \subseteq csq(S)$, it follows that $d \notin S$. Hence S attacks d. There are two cases:

(1) S attacks d by conflict, i.e., $\neg hd(d) \in csq(S)$. Hence $bd(d) \cup B \vdash_{S \cup \{d\}} \bot$. Because S is consistent, $bd(d) \cup B \vdash_{S \cup \{d\}} \bot$, and T is well-defined, we can conclude that there exists $d_0 \in S$ such that $d \prec_S d_0$. That means S attacks d_0 by specificity. Hence S attacks

itself (because $d_0 \in S$). This contradicts the assumption that S is an extension. So, this case cannot occur.

(2) S attacks d by specificity, i.e., there exists a default d' such that $d' \prec_S d$, $bd(d') \subseteq csq(S)$. Hence $B \cup \{hd(d), hd(d')\} \vdash \bot$. Because $bd(d') \subseteq csq(S)$, $bd(d) \cup B \vdash_S bd(d')$. It is clear that $bd(d) \cup B \nvdash_S \bot$. Therefore, $d \prec_S d'$. Hence, $d \prec^* d$. This contradicts the fact that D is well-defined. That means this case cannot occur either.

Since both cases are impossible, $hd(d) \in csq(S)$. This holds for every extension of T. Hence, $T \mathrel{|\!\sim} hd(d)$. $\square$

Appendix C. Properties of default theories of inheritance networks

In this section by Γ we denote an arbitrary but fixed, acyclic, and consistent network. Let T_Γ be the default theory corresponding to Γ. For a path σ in Γ, let $d(\sigma)$ and $r(\sigma)$ denote the set of defaults and rules corresponding to defeasible and strict links belonging to σ, which do not begin from an individual node, respectively. In other words,

$$\begin{aligned} d(\sigma) &= \big\{p(X) \to q(X) \mid p \notin I_\Gamma,\ p \to q \text{ belongs to } \sigma\big\} \cup \\ &\quad \big\{p(X) \to \neg q(X) \mid p \notin I_\Gamma,\ p \not\to q \text{ belongs to } \sigma\big\}. \\ r(\sigma) &= \big\{p(X) \Rightarrow q(X) \mid p \notin I_\Gamma,\ p \Rightarrow q \text{ belongs to } \sigma\big\} \cup \\ &\quad \big\{p(X) \Rightarrow \neg q(X) \mid p \notin I_\Gamma,\ p \not\Rightarrow q \text{ belongs to } \sigma\big\}. \end{aligned}$$

By $d(\sigma)/a$ and $r(\sigma)/a$ we denote the set of ground defaults and ground rules obtained from $d(\sigma)$ and $r(\sigma)$ by instantiating the variable X with a.

Lemma C.1. *For acyclic and consistent network Γ, T_Γ is stratified.*

Proof. To prove the lemma, we define a *rank* function over ground literals of $T_\Gamma \cup \{\top, \bot\}$, that satisfies the conditions of Definition 4.1, as follows.

(i) $rank(\top) = rank(\bot) = 0$; and

(ii) for each individual node a and predicate node p, $rank(p(a)) = rank(\neg p(a)) = \max\{|d(\sigma)/a| \mid \sigma$ is a generalized path from a to $p\}$ where $|d(\sigma)/a|$ denotes the cardinality of the set $d(\sigma)/a$.

Since Γ is acyclic, for every $a \in I_\Gamma$ and $p \in Pred_\Gamma$, $rank(p(a))$ is defined for every p and a. In other words, *rank* is defined for every literal of T_Γ. Furthermore, by its definition, *rank* satisfies the first two conditions of Definition 4.1. Thus, to complete the proof, we consider the following two cases.

- $p(a) \Rightarrow q(a)$ is a rule in B_Γ. Since for each generalized path σ from a to p there exists a generalized path $\sigma' = \sigma \Rightarrow q$ from a to q with $|d(\sigma)/a| = |d(\sigma')/a|$, by definition of *rank*, we conclude that $rank(p(a)) \leqslant rank(q(a))$. By definition of generalized paths, we can also prove that $rank(q(a)) \leqslant rank(p(a))$. Hence, $rank(p(a)) = rank(q(a))$. Similarly, we can prove that if $p(a) \Rightarrow \neg q(a) \in B_\Gamma$, then $rank(p(a)) = rank(\neg q(a))$.

- $p(a) \to q(a)$. Since for each generalized path σ from a to p there exists a generalized path $\sigma' = \sigma \to q$ from a to q with $|d(\sigma)/a| + 1 = |d(\sigma')/a|$, by definition of *rank*, we conclude that $rank(p(a)) < rank(q(a))$. Similarly, we can prove that if $p(a) \not\to q(a) \in D_\Gamma$, then $rank(p(a)) < rank(\neg q(a))$.

The above two cases conclude the lemma. □

The next two lemmas show the correspondence between the consequence relation $\vdash$ in T_Γ and paths in a consistent and acyclic Γ.

Lemma C.2. *Let a be a constant in the language of T_Γ and $\sigma = \pi(x, \delta, u)$ be a positive path in Γ. Then, $x(a) \cup r(\sigma)/a \vdash_{d(\sigma)/a} u(a)$ and $x(a) \cup B_\Gamma \not\vdash_{d(\sigma)/a} \bot$.*

Likewise, let a be a constant in the language of T_Γ and $\sigma = \overline{\pi}(x, \delta, u)$ be a negative path of Γ. Then, $x(a) \cup r(\sigma)/a \vdash_{d(\sigma)/a} \neg u(a)$ and $x(a) \cup B_\Gamma \not\vdash_{d(\sigma)/a} \bot$.

Lemma C.3. *Let $x(a)$ and $u(a)$ be ground literals of T_Γ and K be a minimal set of defaults such that $x(a) \cup B_\Gamma \vdash_K u(a)$ (respectively $x(a) \cup B_\Gamma \vdash_K \neg u(a)$). Then, there exists a positive path $\sigma = \pi(x, \delta, u)$ of Γ (respectively a negative path $\sigma = \overline{\pi}(x, \delta, u)$ of Γ) such that $d(\sigma)/a = K$.*

The next lemma represents a relationship between the more specific relation in T_Γ and paths in Γ. We note that if d is a default in D_Γ then the predicate symbol occurring in the body of d is the label of a node in Γ. We will call it the start node of d.

Lemma C.4. *For two ground defaults d and d' of T_Γ if $d \prec_K d'$ then there exists a positive path from p to q in Γ where p and q are the begin nodes of d and d' respectively.*

We omitted here the proofs of these three lemmas as they are fairly simple and straightforward.

Theorem 4.6. *For every consistent and acyclic network Γ, the default theory corresponding to Γ, T_Γ, is well-defined and stratified.*

Proof. By Lemma C.1, we have that T_Γ is stratified. To prove that T_Γ is well-defined, we need to prove for each default d in D_Γ, the following conditions are satisfied.

(i) $d \not\prec^* d$; and

(ii) for each set of defaults K in T_Γ such that $bd(d) \cup B_\Gamma \vdash_{K\cup\{d\}} \bot$ and $bd(d) \cup B_\Gamma \not\vdash_K \bot$ there exists a default $d_0 \in K$ such that $d \prec_K d_0$.

Let us begin with (i). Assume the contrary, there exists a sequence of default $d \prec d_1 \prec \cdots \prec d_n \prec d$. Let $p, p_1, \ldots, p_n$ be the begin nodes of $d, d_1, \ldots, d_n$ respectively. Then, by Lemma C.4, there exists a path from p to p over $p_1, \ldots, p_n$. This violates the acyclicity of Γ. Thus, our assumption is incorrect, i.e., $d \not\prec^* d$, or (i) holds. (1)

We now prove (ii). Let $H \subseteq K$ be a minimal set satisfying that $bd(d) \cup B_\Gamma \vdash_{H\cup\{d\}} \bot$ and $bd(d) \cup B_\Gamma \not\vdash_H \bot$. We will prove that there exists a default $d' \in K$ such that $d \prec_H d'$, and hence, $d \prec_K d'$. Assume that d is a default of the form $u(c) \to v(c)$ [16]. If $H = \emptyset$, then $u \in N(v)$ or $v \in N(u)$. In both cases, we have that Γ is inconsistent. Thus, $H \neq \emptyset$. Then, from our assumption about d and H, we have that $u(c) \cup B_\Gamma \vdash_H \neg v(c)$. This, together with Lemmas C.3 and C.2, implies that there exists a path $\sigma = \bar{\pi}(u, \delta, v)$ such that $d(\sigma)/c = H$, and for every $d' \in H$, $u(c) \cup B_\Gamma \vdash_H bd(d')$ and $u(c) \cup B_\Gamma \not\vdash_H \perp$. It is easy to check that $d \prec_H d'$ where d' is the default in H such that $rank(hd(d')) = \max\{rank(hd(d'')) \mid d'' \in H\}$. This implies that (ii) holds for d. The proof for (ii) for default of the form $u(c) \to \neg v(c)$ is similar. Hence, we conclude that for every $d \in D_\Gamma$, (ii) holds. (2)

It follows from (1) and (2) that T_Γ is well-defined. □

To prove Theorem 4.7, we need the following notation. For a set of paths Δ in Γ, let

$$\begin{aligned} csq\_path(\Delta) &= \big\{p(a) \mid a \in I_\Gamma,\ \exists \pi(a, \ldots, p) \in \Delta\big\} \cup \\ &\quad \big\{\neg p(a) \mid a \in I_\Gamma,\ \exists \bar{\pi}(a, \ldots, p) \in \Delta\big\}, \\ S_\Delta &= \big\{d \in D_\Gamma \mid \text{there exists a ground path in } \Delta \text{ containing } d\big\} \cup \\ &\quad \big\{d \in D_\Gamma \mid bd(d) \not\subseteq csq\_path(\Delta) \text{ and } \neg hd(d) \notin csq\_path(\Delta)\big\}, \end{aligned}$$

and for a set of defaults S,

$$path(S) = \big\{\pi(a, \sigma, p) \mid d(\sigma)/a \subseteq S\big\} \cup \big\{\bar{\pi}(a, \sigma, p) \mid d(\sigma)/a \subseteq S\big\}.$$

It is easy to see that the following lemma holds.

Lemma C.5. *For every credulous extension Δ of Γ, $csq\_path(\Delta)$ is consistent and $csq\_path(\Delta) = csq(S_\Delta)$.* [17]

This leads us to the following lemma.

Lemma C.6. *Let Δ be a credulous extension of Γ. Then, S_Δ is an extension of T_Γ.*

Proof. First, we show that S_Δ does not attack itself. Assume the contrary, S_Δ attacks itself. We consider two cases:

- S_Δ attacks a default $d \in S_\Delta$ by conflict. Assume that $d = u(c) \to v(c)$. That means that $\neg v(c) \in csq(S_\Delta) = csq\_path(\Delta)$. By definition of S_Δ, $u(c) \in csq\_path(\Delta)$. Thus, from $d \in S_\Delta$ and $u(c) \in csq\_path(\Delta)$, we conclude that there exists a path σ containing d in Δ. Due to the constructivity of Δ, we have that $v(c) \in csq\_path(\Delta)$. This contradicts the fact that $csq\_path(\Delta)$ is consistent. So, this case cannot occur.
- S_Δ attacks a default $d \in S_\Delta$ by specificity. Assume that $d = u(c) \to v(c)$. This means that there exists a default d' in D_Γ and a set of defaults $K \subseteq S_\Delta$ such that $d' \prec_K d$, $p(c) \in csq(S_\Delta)$ where p is the start node of the link d'. Again, there are two cases:
  (a) d' has the form $p(c) \to \neg q(c)$. From the definition of the more specific relation, we can easily verify that $u(c) \in csq(S_\Delta)$. Therefore, by construction of S_Δ, we conclude that there exists a path $\sigma_1 = \pi(c, \eta_1, p, \eta_2, u, v) \in \Delta$. Since $d' \prec_K d$, we have that $B_\Gamma \cup \{\neg q(c), v(c)\}$ is inconsistent. This implies that $v \in P(q)$ or $q \in P(v)$. Furthermore, $\sigma_2 = \pi(c, \eta_1, p) \in \Delta$. Thus, σ_1 is preempted in Δ (because of σ_2 and

[16] Recall that we assume that defaults are grounded.

[17] Recall that for a set of defaults X of a theory (E, B, D), $csq(X) = \{l \mid E \cup B \vdash_X l\}$.

$p \not\to q$). This contradicts the fact that Δ is a credulous extension of Γ. So, this case cannot occur. (1)

(b) d' has the form $p(c) \to q(c)$. Similar to the above case, we can show that Δ contains a path which is preempted in Δ, and hence, we conclude that this case cannot occur too. (2)

The second case is proved by (1) and (2).

The above two cases show that S_Δ does not attack itself. (3)

To complete the proof, we need to show that S_Δ attacks every $d \notin S_\Delta$. From the construction of S_Δ, there are two cases:

- $bd(d) \not\subseteq csq\_path(\Delta)$. Then, by definition of S_Δ, $\neg hd(d) \in csq\_path(\Delta) = csq(S_\Delta)$. Hence, S_Δ attacks d by conflict. (4)
- $bd(d) \subseteq csq\_path(\Delta)$. Assume that $d = p(c) \to q(c)$ for some individual node c. This implies that there exists a path $\sigma_1 = \pi(c, \eta, p)$ in Δ. Thus, the path $\sigma_2 = \pi(c, \eta, p) \to q$ is constructible in Δ. Since $d \notin S_\Delta$, $\sigma_2 \notin \Delta$. If σ_2 is conflict in Δ, then $\neg hd(d) \in csq(S_\Delta)$, and hence, σ_2 is attacked by conflict by S_Δ. Otherwise, σ_2 is preempted in Δ, then we can easily check that d is attacked by S_Δ by specificity. (5)

From (4)–(5) we conclude that S_Δ attacks every $d \notin S_\Delta$. Together with (3), we have proved that S_Δ is an extension of T_Γ. □

We now prove the reverse of Lemma C.6.

Lemma C.7. *Let S be a consistent extension of T_Γ. Then, $path(S) = \{\sigma \mid \sigma$ is a ground path and $path(S) \mathrel{|\!\!\sim} \sigma\}$.*

Proof. Consider a path $\sigma = \pi(a, \delta, p)$ in $path(S)$. We prove that $path(S) \mathrel{|\!\!\sim} \sigma$ by induction over $|d(\sigma)/a|$.

Base case: $|d(\sigma)/a| = 0$. That is, σ is a direct link or a strict path. Thus, $d(\sigma)/a \subseteq S$. By construction of $path(S)$, we have that $\sigma \in path(S)$. The base case is proved.

Inductive step: Assume that we have proved $path(S) \mathrel{|\!\!\sim} \sigma$ for $\sigma \in path(S)$ with $|d(\sigma)/a| \leqslant n$. We need to prove it for $|d(\sigma)/a| = n + 1$. First, let us prove the case where $\sigma = Def(\sigma)$. Assume that $\sigma = \pi(a, \delta, u) \to p$. Let $\tau = \pi(a, \delta, u)$. By construction of $path(S)$ and inductive hypothesis, we have that $\tau \in path(S)$ and $path(S) \mathrel{|\!\!\sim} \tau$. Thus, σ is constructible in $path(S)$. (1)

Since S is an extension of T_Γ and $p(a) \in csq(S)$ (because $E_\Gamma \cup B_\Gamma \vdash_{d(\sigma)/a} p(a)$), we have that $\neg p(a) \notin csq(S)$. Thus, there exists no path in $path(S)$ supporting $\neg p(a)$. In other words, σ is not conflicted in $path(S)$. (2)

To prove that $path(S) \mathrel{|\!\!\sim} \sigma$, we need to show that σ is not preempted in $path(S)$. Assume the contrary, then there are the following cases:

- There exists a link $a \not\to t$ in Γ and $t \in P(p)$. In this case, we have that $\neg t(a) \in E_\Gamma$ and $E_\Gamma \cup B_\Gamma \vdash_\emptyset \neg p(a)$. Hence, $\neg p(a) \in csq(S)$. This contradicts the fact that $p(a) \in csq(S)$ and S is a consistent extension of T_Γ. Thus, this case cannot occur.
- There exists a link $a \to t$ in Γ and $t \in N(p)$. In this case, we have that $t(a) \in E_\Gamma$ and $E_\Gamma \cup B_\Gamma \vdash_\emptyset \neg p(a)$. Hence, $\neg p(a) \in csq(S)$. This contradicts the fact that $p(a) \in csq(S)$ and S is a consistent extension of T_Γ. Thus, this case cannot occur.

- There exists a path $\pi(a,\alpha,v,\beta,u)$ in *path*(S) and $v \not\to t \in \Gamma$, and $t \in P(p)$. Let $\gamma = \pi(v,\beta,u)$. It is easy to see that we have $v(a) \to \neg t(a) \prec_{d(\gamma)/a} u(a) \to p(a)$ and $d(\gamma)/a \subseteq S$. This implies that d is attacked by S by specificity and $d \in S$, i.e., S attacks itself. This contradicts the fact that S is an extension of T_Γ. Therefore, this case cannot occur too.
- There exists a path $\pi(a,\alpha,v,\beta,u)$ in *path*(S) and $v \to t \in \Gamma$ and $t \in N(p)$. Similar to the third the case, we conclude that this case cannot occur.

The above four cases show that our assumption that σ is preempted in *path*(S) is incorrect. In other words, σ is not preempted in *path*(S). (3)

Similarly, we can prove (1)–(3) for defeasible negative paths in *path*(S) whose last link is a defeasible link. (4)

Now, consider the case the last link of σ is a strict link, i.e., $|d(\sigma)| = n+1$ and $\sigma \neq \mathit{Def}(\sigma)$. From (3) and (4), we can show that *path*$(S) \mid\!\sim \mathit{Def}(\sigma)$ is $\sigma \in$ *path*(S). Furthermore, by definition of *path*(S), we have that *path*$(S) \mid\!\sim \mathit{Str}(\sigma)$. Hence, we have that *path*$(S) \mid\!\sim \sigma$ for this case too. (5)

The inductive step follows from (1)–(5) for paths in *path*(S). So, we have proved that if $\sigma \in$ *path*(S) then *path*$(S) \mid\!\sim \sigma$. (6)

Similarly, we can show that if $\sigma \notin$ *path*(S) then *path*$(S) \not\mid\!\sim \sigma$. Together with (6), we conclude the lemma. $\square$

The next lemma is the final step toward the proof of Theorem 4.6.

Lemma C.8. *Let S be an extension of T_Γ. Then, there exists a credulous extension Δ of Γ such that csq(S) = csq_path(Δ).*

Proof. From Lemma C.7 and the definition of *path*(S), we have that *csq*(S) = *csq_path*(*path*(S)). Thus, to prove the lemma we prove that there exists a credulous extension Δ of Γ such that *path*$(S) \subseteq \Delta$ and every ground path $\sigma \in \Delta$ belongs to *path*(S). $(*)$

Let Γ' be the network obtained from Γ by removing from Γ all individual nodes and links going out from these nodes. Let Φ be a credulous extension of Γ'. It is easy to see that $\Delta =$ *path*$(S) \cup \Phi$ is a credulous extension of Γ that satisfying $(*)$. The lemma is proved. $\square$

We now prove the theorem about the relationship between the credulous semantics of Γ and that of T_Γ.

Theorem 4.7. *For every acyclic and consistent inheritance network Γ, an individual node a, and a predicate node p,*

(1) $\Gamma \mid\!\sim_c p(a)$ *iff* $T_\Gamma \mid\!\sim p(a)$; *and*
(2) $\Gamma \mid\!\sim_c \neg p(a)$ *iff* $T_\Gamma \mid\!\sim \neg p(a)$.

Proof. Assume that $\Gamma \mid\!\sim_c p(a)$. We prove that $T_\Gamma \mid\!\sim p(a)$. Assume the contrary, $T_\Gamma \not\mid\!\sim p(a)$. This means that there exists an extension S of T_Γ such that $p(a) \notin$ *csq*(S). By

Lemma C.8, there exists a credulous extension of Γ which does not support $p(a)$. In other words, $\Gamma \not\mid\!\sim_c p(a)$. This contradicts the assumption that $\Gamma \mid\!\sim_c p(a)$. Thus, $T_\Gamma \mid\!\sim p(a)$. (1)

Similarly, using Lemma C.6, we can show that if $T_\Gamma \mid\!\sim p(a)$ then $\Gamma \mid\!\sim p(a)$. (2)

The first conclusion of the theorem follows from (1) and (2).

The second conclusion of the theorem can be proven similarly. □

Appendix D. Translation into Reiter's default logic

Let $T = (E, B, D)$ be a default theory. Recall that for $K \subseteq D$, $csq(K) = \{l \mid E \cup B \vdash_K l\}$. Further, for a set of first-order sentences X, $Th(X)$ denotes the least logical closure of X. For a literal l in $\mathcal{L}$, let $atom(l)$ denote the atom occurring in l and l_d denote the literal obtained from l by replacing $a = atom(l)$ with a_d if $a \notin bd(d)$. For simplicity of the presentation, we define $\mathcal{L}_d = L_d \cup \{a \mid a$ is an atom occurring in $bd(d)\}$. For a literal l in $\mathcal{L}_d$, let $origin(l, d)$ denote the literal h in $\mathcal{L}$ such that $h_d = l$. For a set of literals X in $\mathcal{L}$, let $X_d = \{l_d \mid l \in X\}$. It is easy to see that the construction of B_d and D_d satisfies the following lemma.

Lemma D.1. *For a set of literals X and a literal l of $\mathcal{L}$, $X \cup B \vdash l$ iff $X_d \cup B_d \vdash l_d$.*

Lemma D.2. *For a default $d \in D$, a set of defaults $K \subseteq D$, a literal l in $\mathcal{L}$, and a set of literals X, if $X \cup B \vdash_K l$ then $X_d \cup B_d \vdash_{K_d} l_d$ where $K_d = \{c'_d \mid c' \in K\}$.*

Proof. Without the loss of generality, we can assume that K is a minimal set of defaults (with respect to the set inclusion operator). We prove the lemma by induction over $|K|$, the cardinality of K. The inductive case, $|K| = 0$, is trivial because of Lemma D.1. Assume that we have proved the lemma for $|K| = m$. We need to prove the lemma for $|K| = m + 1$. Assume that $K = \{c^1, \ldots, c^m, c^{m+1}\}$ and $c^1, \ldots, c^m, c^{m+1}$ is a defeasible derivation of l. By Definition 3.2, we have that $X \cup B \vdash_{K \setminus \{c^{m+1}\}} bd(c^{m+1})$ and $X \cup B \cup Y \vdash l$ where $Y = \{hd(c^1), \ldots, hd(c^{m+1})\}$. From the inductive hypothesis, we can conclude that $X_d \cup B_d \vdash_{K_d \setminus \{c_d^{m+1}\}} bd(c_d^{m+1})$. Furthermore, by Lemma D.1, we have that $X_d \cup B_d \cup Y_d \vdash l_d$. It follows from Definition 3.2 that $X_d \cup B_d \vdash_{K_d} l_d$. The inductive step is proved and hence the lemma is proved. □

Lemma D.3. *For every literal $l \in \mathcal{L}_d$ and a set of defaults $K \subseteq D^*$, $E \cup (B^* \setminus B') \vdash_K l$ iff there exists a set $X \subseteq bd(d)$ such that $E \cup B \vdash_{K \cap D} X$ and $X \cup B_d \vdash_{K \cap D_d} l$.*

Proof. It follows from the construction of B_d and D_d that

$E \cup (B^* \setminus B') \vdash_K l$

iff $E \cup B \cup B_d \vdash_{K \cap (D \cup D_d)} l$

iff $E \cup B \vdash_{K \cap D} Y$ and $Y \cup B_d \vdash_{K \cap D_d} l$

iff $E \cup B \vdash_{K \cap D} X$ and $X \cup B_d \vdash_{K \cap D_d} l$ for $X = bd(d) \cap Y$.

The lemma is proved. □

Lemma D.4. *Let X be a set of literals of $\mathcal{L} \cup \mathcal{L}_d$ and l be a literal in $\mathcal{L}_d$ such that $X \cup B_d \vdash_K l$ for some set of defaults $K \subseteq D_d$. Then $X^d \cup B \vdash_{K^d} origin(l, d)$ where $X^d = \{origin(l, d) \mid l \in X\}$ and $K^d = \{c \mid c \in D$ and $c_d \in K\}$.*

Proof. Without the loss of generality, we can assume that K is a minimal set of defaults (with respect to the set inclusion operator). We prove that $X^d \cup B \vdash_{K^d} origin(l, d)$ by induction over $|K|$, the cardinality of K. The inductive case, $|K| = 0$ follows from Lemma D.1. Assume that we have proved the lemma for $|K| = m$. We need to prove the lemma for $|K| = m + 1$. Assume that $K = \{c_d^1, \ldots, c_d^m, c_d^{m+1}\}$ and $c_d^1, \ldots, c_d^m, c_d^{m+1}$ is a defeasible derivation of l. By Definition 3.2, we have that $X \cup B_d \vdash_{K \setminus \{c_d^{m+1}\}} bd(c_d^{m+1})$ and $X \cup B_d \cup Y_d \vdash l$ where $Y = \{hd(c^1), \ldots, hd(c^{m+1})\}$. From the inductive hypothesis, we can conclude that $X^d \cup B \vdash_{K^d \setminus \{c^{m+1}\}} bd(c_{m+1})$ and $X^d \cup B \cup Y \vdash origin(l, d)$ (Lemma D.1). This implies that $X^d \cup B \vdash_{K^d} origin(l, d)$. The inductive step is proved and hence the lemma is proved. □

To continue, we need some additional notations. Let S be a set of defaults in T, $S \subseteq D$. Define $Ab(S) = Ab_1(S) \cup Ab_2(S)$ where

$$Ab_1(S) = \{ab_d \mid d \notin S\},$$
$$Ab_2(S) = \{ab_{d_c} \mid ab_d \in Ab_1(S)\},$$
$$Reduct(D_T, S) = \left\{\alpha \to \gamma \;\middle|\; \frac{\alpha : \beta}{\gamma} \in D_T \text{ and } \neg\beta \notin \big(Ab_1(S) \cup Ab_2(S)\big)\right\},$$

and

$$Consequence(D_T, S) = \big\{l \mid E \cup (B^* \setminus B') \vdash_{Reduct(D_T, S)} l\big\}.$$

Furthermore, let $R'_T = (W_T \setminus B', D'_T)$ where

$$D'_T = regular(T) \cup equi(D) \cup \left\{\frac{bd(d), bd(c_d) : \top}{ab_c} \;\middle|\; d, c \in D, \text{ and } B \cup \{hd(d), hd(c)\} \text{ is inconsistent}\right\}.$$

First, it is easy to see that R_T and R'_T are equivalent in the following sense.

Lemma D.5. *For every default theory T, S is an extension of R'_T iff $Th(S' \cup B')$ is an extension of $R_T = (W_T, D_T)$.*

Proof. It is easy to see that any default $y \in D_T \setminus D'_T$ has the form

$$\frac{bd(d), bd(c_d), (B' \cup (hd(d))' \Rightarrow \neg(hd(c))') : \top}{ab_c}$$

and $B \cup \{hd(d), hd(c)\}$ is consistent. This implies that the prerequisite of y can never be satisfied, and hence, y cannot be applied in any consistent set of formulas of R_T. This, together with the fact that no default in R_T has its consequent in the language of B', concludes the lemma. □

It follows from Lemma D.5 that to prove Theorem 5.1 it is sufficient to prove that T is equivalent to R'_T. This is what we will do in the rest of this appendix. The following lemma follows from the definition of $Reduct(D_T, S)$.

Lemma D.6. *For every set of defaults S, a default $d \in S$, and a set of defaults $K \subseteq S$,*

$$K_d = \{c'_d \mid c' \in K\} \subseteq Reduct(D_T, S).$$

Lemma D.7. *For an extension S of T, $Consequence(D_T, S) \cap \mathcal{L} = csq(S)$.*

Lemma D.8. *For an extension S of T and a default $d \in D^*$, $ab_d \in Consequence(D_T, S)$ iff $ab_d \in Ab(S)$.*

Proof. First, we show that if $ab_d \in Ab(S)$ then $ab_d \in Consequence(D_T, S)$. Consider two cases:

- $d \in D$. Then, $ab_d \in Ab_1(S)$. Thus, $d \notin S$. This implies that either (i) S attacks d by conflict or (ii) S attacks d by specificity. (i) and Lemma D.8 imply that $\neg hd(d) \in Consequence(D_T, S)$, and hence, $ab_d \in Consequence(D_T, S)$ because $\neg hd(d) \to ab_d \in Reduct(D_T, S)$. (ii) implies that there exists some $c \in S$ and $K \subseteq S$ such that $c \prec_K d$ and $E \cup B \vdash_K bd(c)$. Again, from Lemma D.8, we have that $bd(c) \subseteq Consequence(D_T, S)$. This also implies that $K \subseteq Reduct(D_T, S)$, and hence, $K_c = \{d'_c \mid d' \in K\} \subseteq Reduct(D_T, S)$ (Lemma D.7). Because $bd(c) \cup B \vdash_K bd(d)$, we have that $bd(c) \cup B_c \vdash_{K_c} bd(d_c)$ (Lemma D.2). By definition of $Consequence(D_T, S)$, $ab_d \in Consequence(D_T, S)$.
- d is a default in D_c, say p_c. $ab_{p_c} \in Ab(S)$ means that $ab_{p_c} \in Ab_2(S)$. By definition of $Ab_2(S)$, we have that $ab_p \in Ab_1(S)$. From the above case, we have that $ab_p \in Consequence(D_T, S)$, and hence, $ab_{p_c} \in Consequence(D_T, S)$.

We now prove that if $ab_d \in Consequence(D_T, S)$ then $ab_d \in Ab(S)$. Consider two cases:

- $d \in D$. Then, $ab_d \in Consequence(D_T, S)$ if either (i) $\neg hd(d) \in Consequence(D_T, S)$ or (ii) there exists some c such that $bd_c \cup bd_{d_c} \subseteq Consequence(D_T, S)$. (i), together with Lemma D.8, implies that $\neg hd(d) \in csq(S)$, and hence, d is attacked by conflict by S. (ii), together with Lemmas D.3 and D.4, implies that there exists a set of defaults $K \subseteq S$ such that $bd(c) \cup B \vdash_K bd(d)$, and $E \cup B \vdash_S bd(c)$. Because $B \cup \{hd(d), hd(c)\}$ is inconsistent, this implies that d is attacked by specificity by S. In both case, we have that $d \notin S$ which implies that $ab_d \in Ab_1(S)$.
- d is a default in D_c, say p_c. $ab_{p_c} \in Consequence(D_T, S)$ if either (i) $\neg hd(p_c) \in Consequence(D_T, S)$ or (ii) $ab_p \in Consequence(D_T, S)$. From (i) and Lemma D.4 imply that $\neg hd(p) \in Consequence(D_T, S)$, and hence, $ab_p \in Consequence(D_T, S)$. In both cases, we have that $ab_p \in Consequence(D_T, S)$. It follows from the above case that $ab_{p_c} \in Ab_2(S)$.

The conclusion of the lemma follows from the above two cases. $\square$

To prove the equivalence between R'_T and T we will need the following notation. Let Y be a set of first-order formula in the language of R'_T. Define, $B^+ = B^* \setminus B'$,

$$appl(Y) = \left\{\alpha \to \gamma \;\middle|\; \frac{\alpha : \beta}{\gamma} \in D_T, \neg\beta \notin Y\right\}$$

and

$$concl(Y) = \left\{l \mid E \cup B^+ \vdash_{appl(Y)} l\right\}.$$

Let $\Gamma(Y)$ be the smallest set of first-order sentences satisfying the following properties:
(c1) $E \cup B^+ \subseteq \Gamma(Y)$;
(c2) $\Gamma(Y)$ is deductively closed; and
(c3) if $\frac{\alpha:\beta}{\gamma} \in D_T$, $\alpha \in \Gamma(Y)$, and $\neg\beta \notin Y$ then $\gamma \in \Gamma(Y)$.

Lemma D.9. *For every set of sentences Y, $concl(Y) \subseteq \Gamma(Y)$.*

Proof. Let l be an arbitrary literal in $concl(Y)$. By definition, there exists a set of defaults $K \subseteq appl(Y)$ such that $E \cup B^+ \vdash_K l$. Without the loss of generality, we can assume that K is minimal (with respect to the set inclusion operator). We prove that $l \in \Gamma(Y)$ by induction over $|K|$, the cardinality of K. The inductive case, $|K| = 0$, is trivial since $E \cup B^+ \subseteq \Gamma(Y)$ and $\Gamma(Y)$ is deductively closed. Assume that we have proved the lemma for $|K| = n$. Consider the case $|K| = n + 1$. Without the loss of generality, we can assume that $K = \{d_1, \ldots, d_n, d_{n+1}\}$ and $d_1, \ldots, d_{n+1}$ is a defeasible derivation of l. By Definition 3.2 and the inductive step, we can conclude that $bd(d_i) \subseteq concl(Y)$ and $bd(d_i) \subseteq \Gamma(Y)$ for every $i \in \{1, \ldots, n+1\}$. This implies that $hd(d_i) \in concl(Y)$ and $hd(d_i) \in \Gamma(Y)$ for every $i \in \{1, \ldots, n+1\}$. Again, because $\Gamma(Y)$ is deductively closed, we conclude that $l \in \Gamma(Y)$ because $l \in Th(E \cup B^+ \cup \{hd(d_1), \ldots, hd(d_{n+1})\})$. The inductive step, and hence, the lemma is proved. □

Lemma D.10. *S is an extension of R'_T iff $S = Th(B^+ \cup concl(S))$.*

Proof. (⇒) Let S be an extension of R'_T. Recall that S is an extension of R'_T iff $S = \Gamma(S)$. It is easy to see that $S' = Th(B^+ \cup concl(S))$ satisfies the following properties:
(i1) $E \cup B^+ \subseteq S'$ (because $E \cup B^+ \subseteq Th(concl(S))$),
(i2) S' is deductively closed (because of its definition), and
(i3) if $\frac{\alpha:\beta}{\gamma} \in D_T$, $\alpha \in S'$, and $\neg\beta \notin S$ then $\gamma \in S'$.
Thus, because of the minimality of $\Gamma(S)$, we have that $\Gamma(S) \subseteq Th(B^+ \cup concl(S))$. On the other hand, because S is an extension of R'_T, $concl(S) \subseteq \Gamma(S)$. This, together with (c3), shows that $B^+ \cup concl(S) \subseteq \Gamma(S)$. Because of (c2), we can conclude that $Th(B^+ \cup concl(S)) \subseteq \Gamma(S)$. Hence, $S = \Gamma(S) = Th(B^+ \cup concl(S))$.

(⇐) Let S be a set of first-order sentences in R'_T such that $S = Th(B^+ \cup concl(S))$. It is easy to see that $E \cup B^+ \subseteq S$ and S is deductively closed. Furthermore, $concl(S) \subseteq S$, and hence, $\Gamma(S) \subseteq S$ because of the minimality of $\Gamma(S)$. By Lemma D.9, we have that $concl(S) \subseteq \Gamma(S)$. Because $B^+ \subseteq \Gamma(S)$ and $\Gamma(S)$ is deductively closed, we have that $S = Th(B^+ \cup concl(S)) \subseteq \Gamma(S)$. This implies that $S = \Gamma(S)$, i.e., S is an extension of R'_T. □

Lemma D.11. *Let S be an extension of T. Then, $A = Th(B^+ \cup Consequence(D_T, S))$ is an extension of R'_T.*

Proof. It is easy to see that $ab_d \in A$ iff $ab_d \in Consequence(D_T, S)$. Thus, by Lemma D.8, $ab_d \in A$ iff $ab_d \in Ab(S)$. Hence, $appl(A) = Reduct(D_T, S)$. Therefore,

$$concl(A) = Consequence(D_T, S).$$

Thus, $A = Th(B^+ \cup concl(A))$. By Lemma D.10, we have that A is an extension of R'_T. □

Lemma D.12. *Let A be a consistent extension of R'_T. Then, $S = \{d \mid d \in D$ and $ab_d \notin A\}$ is an extension of S.*

Proof. It is easy to see that $Ab(S) = \{ab_d \mid d \in D^*$ and $ab_d \in A\}$. By construction of S, it is easy to see that $S \subseteq Reduct(D_T, S)$.

First, we prove that S does not attack itself. Assume the contrary, i.e., S attacks some $d \in S$. We consider two cases. S attacks d by conflict, that means that $E \cup B \vdash_S \neg hd(d)$. This implies that $\neg hd(d) \in A$, i.e., $ab_d \in A$. This contradicts the assumption that $d \in S$. Hence, this case cannot occur. S attacks d by specificity. This means that there exists a default $c \in S$ such that $c \prec_S d$ and $E \cup B \vdash_S bd(c)$. So, $bd(c) \subseteq A$ and $bd(c_d) \subseteq A$, which implies that $ab_d \in A$. Again, this contradicts the fact that $d \in S$, i.e., this case cannot happen too. Thus, our assumption is incorrect, i.e., we have proved that S does not attack itself.

For each $d \in D \setminus S$, either (i) $\neg hd(d) \in A$ or (ii) there exists some $c \in D$ such that $bd(c) \subseteq A$ and $bd(c_d) \subseteq A$. (i) implies that S attacks d by conflict; (ii) implies that $bd(c) \vdash_K bd(d)$ for some $K \subseteq S$ and $E \cup B \vdash_S bd(c)$, i.e., S attacks d by specificity. Both cases prove that S attacks d, i.e., S attacks every default that does not belong to S. Together with the fact that S does not attack itself, we conclude that S is an extension of T. The lemma is proved. □

We now prove Theorem 5.1.

Theorem 5.1. *Let T be a default theory and l be a ground $\mathcal{L}$-literal. Then, $T \mid\sim l$ iff l is contained in every extension of R_T.*

Proof. Because of Lemma D.5, it is sufficient to prove that $T \mid\sim l$ iff l is contained in every extension of R'_T.

Let $T \mid\sim l$, and A be an extension of R'_T. We want to prove that $l \in A$. Let $S = \{d \mid d \in D$ and $ab_d \notin A\}$. From Lemma D.11, it follows that S is an extension of T. Hence $l \in csq(S)$. Because $csq(S) \subseteq concl(W)$, it follows that $l \in concl(A)$. Thus, $l \in A$.

Let l be a $\mathcal{L}$-literal contained in every extension of R'_T. We want to prove that $T \mid\sim l$. Let S be an extension of T. Let $A = Th(B^+ \cup Consequence(D_T, S))$. We have that $l \in A$. From the definition of $csq(S)$, it follows that $l \in csq(S)$. □

References

[1] J.J. Alferes, L.M. Pereira, Reasoning with Logic Programming, Lecture Notes in Artificial Intelligence, Vol. 1111, Springer, Berlin, 1996.

[2] F. Baader, B. Hollunder, How to prefer more specific defaults in terminological default logic, in: Proc. IJCAI-93, Chambéry, France, Morgan Kaufmann, San Mateo, CA, 1993, pp. 669–674.

[3] A. Bondarenko, P.M. Dung, R.A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artificial Intelligence 97 (1997) 63–101.

[4] G. Brewka, Cumulative default logic: In defense of nonmonotonic inference rules, Artificial Intelligence 50 (1992) 183–205.

[5] G. Brewka, Adding priorities and specificity to default logic, in: Proc. European Workshop on Logics in AI (JELIA'94), Lecture Notes in Artificial Intelligence, Vol. 838, Springer, Berlin, 1994, pp. 247–260.

[6] G. Brewka, Reasoning about priorities in default logic, in: Proc. AAAI-94, Seattle, WA, AAAI Press, 1994, pp. 940–945.

[7] G. Brewka, T. Eiter, Preferred answer sets for extended logic programs, Artificial Intelligence 109 (1999) 297–356.

[8] J.P. Delgrande, T.H. Schaub, Compiling specificity into approaches to nonmonotonic reasoning, Artificial Intelligence 90 (1997) 301–348.

[9] J.P. Delgrande, T.H. Schaub, A general approach to specificity in default reasoning, in: Proc. 4th International Conference on Knowledge Representation and Reasoning (KR-94), Bonn, Germany, Morgan Kaufmann, San Mateo, CA, 1994, pp. 47–158.

[10] Y. Dimopoulos, B. Nebel, F. Toni, Preferred arguments are harder to compute than stable extensions, in: Proc. IJCAI-99, Stockholm, Sweden, Morgan Kaufmann, San Mateo, CA, 1999, pp. 36–41.

[11] Y. Dimopoulos, B. Nebel, F. Toni, Finding admissible and preferred arguments can be very hard, in: T. Cohn, F. Giunchiglia, B. Selman (Eds.), Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), Breckenridge, CO, Morgan Kaufmann, San Mateo, CA, 2000, pp. 53–61.

[12] P.M. Dung, Negation as hypothesis: An abductive foundation for logic programming, in: K. Furukawa (Ed.), Proceedings of the 8th International Conference on Logic Programming (ICLP'91), MIT Press, Cambridge, MA, 1991, pp. 3–17.

[13] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–357.

[14] P.M. Dung, T.C. Son, Nonmonotonic inheritance, argumentation, and logic programming, in: Proc. 3rd International Conference on Logic Programming and NonMonotonic Reasoning Conference (LPNMR'95), Lecture Notes in Artificial Intelligence, Vol. 928, Springer, Berlin, 1995, pp. 316–329.

[15] P.M. Dung, T.C. Son, An argumentation-theoretic approach to default reasoning with specificity, in: Proc. 5th International Conference on Knowledge Representation Reasoning (KR-96), Cambridge, MA, Morgan Kaufmann, San Mateo, CA, 1996, pp. 407–418.

[16] P.M. Dung, T.C. Son, Default reasoning with specificity, in: Proc. First International Conference on Computational Logic (CL'00), Lecture Notes in Computes Science, Vol. 1861, Springer, Berlin, 2000, pp. 792–805.

[17] P.M. Dung, T.C. Son, An argument-based approach to reasoning with specificity, Technical Report, Computer Science Department, New Mexico State University, Las Cruces, NM, March 2001.

[18] M.R. Garey, D.S. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[19] P. Geerts, D. Vermeir, A nonmonotonic reasoning formalism using implicit specificity information, in: Proc. 2th International Conference on Logic Programming NonMonotonic Reasoning Conference (LPNMR'93), MIT Press, Cambridge, MA, 1993, pp. 380–396.

[20] H. Geffner, J. Pearl, Conditional entailment: Bridging two approaches to default reasoning, Artificial Intelligence 53 (1992) 209–244.

[21] M. Gelfond, T.C. Son, Prioritized default theory, in: Selected Papers from the Workshop on Logic Programming Knowledge Representation 1997, Lecture Notes in Artificial Intelligence, Vol. 1471, Springer, Berlin, 1998, pp. 164–223.

[22] M. Goldszmidt, J. Pearl, Qualitative probabilities for default reasoning, belief revision, and causal modeling, Artificial Intelligence 84 (1–2) (1996) 57–112.
[23] J.F. Horty, Some direct theories of non-monotonic inheritance, in: D. Gabbay, C. Hogger, J. Robinson (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford University Press, Oxford, 1994, pp. 111–187.
[24] A.C. Kakas, R.A. Kowalski, F. Toni, The role of abduction in logic programming, in: D. Gabbay, C. Hogger, J. Robinson (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming 5, Oxford University Press, Oxford, 1998, pp. 235–324.
[25] A.C. Kakas, F. Toni, Computing argumentation in logic programming, J. Logic Comput. 9 (1999) 515–562.
[26] J.F. Horty, R.H. Thomason, D.S. Touretzky, A skeptical theory of inheritance in nonmonotonic semantic networks, Artificial Intelligence 42 (1987) 311–348.
[27] K. Konolige, Hierarchic autoepistemic theories for nonmonotonic reasoning, in: Proc. AAAI-88, St. Paul, MN, AAAI Press, 1988, pp. 439–443.
[28] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, Artificial Intelligence 44 (1990) 167–207.
[29] V. Lifschitz, Computing circumscription, in: Proc. IJCAI-85, Los Angeles, CA, Morgan Kaufmann, San Mateo, CA, 1985, pp. 121–127.
[30] W. Marek, M. Truszczyński, Nonmonotonic Logic: Context Dependent Reasoning, Springer, Berlin, 1993.
[31] D. Matkinson, K. Schlechta, Floating conclusion and zombie paths: Two deep difficulties in the 'directly skeptical' approach to defeasible inheritance nets, Artificial Intelligence 48 (1989) 99–209.
[32] J. McCarthy, Circumscription—A form of nonmonotonic reasoning, Artificial Intelligence 13 (1–2) (1980) 27–39, 171–172.
[33] J. McCarthy, Applications of circumscription to formalizing common sense knowledge, Artificial Intelligence 26 (3) (1986) 89–116.
[34] D. McDermott, J. Doyle, Nonmonotonic logic I, Artificial Intelligence 13 (1–2) (1980) 41–72.
[35] Y. Moinard, Preference by specificity in default logic, in: ESPRIT Project DRUMS RP1 1st Workshop, Marseille, France, February 1990.
[36] R. Moore, Semantical considerations on nonmonotonic logic, Artificial Intelligence 25 (1) (1985) 75–94.
[37] I. Niemelä, Towards efficient default reasoning, in: Proc. IJCAI-95, Montreal, Quebec, Morgan Kaufmann, San Mateo, CA, 1995, pp. 312–318.
[38] J. Pearl, System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning, in: Proc. 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'90), 1990, pp. 121–135.
[39] J.L. Pollock, Defeasible reasoning, Cognitive Sci. 17 (1987) 481–518.
[40] D. Poole, On the comparison of theories: Preferring the most specific explanation, in: Proc. IJCAI-85, Los Angeles, CA, Morgan Kaufmann, San Mateo, CA, 1985, pp. 144–147.
[41] H. Prakken, G.A.W. Vreeswijk, Logics for defeasible argumentation, in: D. Gabbay (Ed.), Handbook of Philosophical Logic, Oxford University Press, Oxford, 1991.
[42] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1–2) (1980) 81–132.
[43] R. Reiter, G. Criscuolo, On interacting defaults, in: Proc. IJCAI-81, Vancouver, BC, Morgan Kaufmann, San Mateo, CA, 1981, pp. 270–276.
[44] E. Sandewall, Nonmonotonic inference rules for multiple inheritance with exceptions, Proc. IEEE 74 (10) (1986) 1345–1353.
[45] B. Selman, H.J. Levesque, The complexity of path-based defeasible inheritance, Artificial Intelligence 62 (2) (1993) 303–340.
[46] G.R. Simari, R.P. Loui, A mathematical treatment of defeasible reasoning and its implementation, Artificial Intelligence 52 (1992) 125–257.
[47] G. Simonet, On Sandewall's paper: Nonmonotonic inference rules for multiple inheritance with exceptions, Artificial Intelligence 86 (2) (1996) 359–374.
[48] G. Simonet, R. Ducournau, On Stein's paper: Resolving ambiguity in nonmonotonic inheritance hierarchies, Artificial Intelligence 71 (1994) 183–193.
[49] L.A. Stein, Resolving ambiguity in nonmonotonic inheritance hierarchies, Artificial Intelligence 55 (1993) 259–310.

[50] M. Thielscher, A nonmonotonic disputation-based semantics and proof procedure for logic programs, in: M. Maher (Ed.), Proc. 1996 Joint International Conference and Symposium on Logic Programming, Bonn, Germany, MIT Press, Cambridge, MA, 2000, pp. 483–497.
[51] D. Touretzky, The Mathematics of Inheritance Systems, Morgan Kaufmann, San Mateo, CA, 1986.
[52] G.A.W. Vreeswijk, The feasibility of defeat in defeasible reasoning, in: Proc. 2nd International Conference on Knowledge Representation Reasoning (KR-91), Cambridge, MA, Morgan Kaufmann, San Mateo, CA, 1991, pp. 478–483.
[53] J.H. You, X. Wang, L.Y. Yuan, Compiling defeasible inheritance networks to general logic programs, Artificial Intelligence 113 (1) (1999) 247–268.