In *Computers and Artificial Intelligence*, Vol. 14(3), pages 225–251, 1995.

Default Reasoning in a Terminological Logic

Fabrizio Sebastiani & Umberto Straccia
Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
Via S. Maria, 46 - 56126 Pisa (Italy)
E-mail : ⟨*lastname*⟩`@iei.pi.cnr.it`

Abstract

Terminological Logics (TLs) are knowledge representation formalisms of considerable applicative interest, as they are specifically oriented to the vast class of application domains that are describable by means of taxonomic organizations of complex objects. Although the field of TLs has lately been an active area of investigation, only few researchers have addressed the problem of extending these logics with the ability to perform *default reasoning*. Such extensions would prove of paramount applicative value, as many application domains may be formalized by means of monotonic TLs only at the price of oversimplification. In this paper we show how we can effectively integrate terminological reasoning and default reasoning, yielding a *terminological default logic*. The kind of default reasoning we embed in our TL is reminiscent of Reiter's Default Logic, but overcomes some of its drawbacks by subscribing to the "implicit" handling of exceptions typical of the Multiple Inheritance Networks with Exceptions proposed by Touretzky and others.

1 Introduction

Terminological Logics (TLs, variously known as *Frame Representation Languages* or *Concept Description Languages*) are knowledge representation formalisms of considerable applicative interest, as they are specifically oriented to the vast class of application domains that are describable by means of taxonomic organizations of complex objects.

Unlike better known logics (such as e.g. FOL), the primary syntactic expressions of TLs are *terms*, denoting monadic or dyadic relations on the domain of discourse. In general, the language of a TL consists of a number of term-forming operators by means of which one may build complex terms starting from a basic repertory of simple terms (namely, predicate symbols). By virtue of the semantics of such operators, a partial order is induced on the terms so defined, giving to a terminological KB the characteristic "hierarchical" (or taxonomic) structure of a directed acyclic graph.

The field of TLs has lately been an active area of research, with the attention of researchers especially focusing on the investigation of their logical and computational properties. Nevertheless, few researchers have addressed the problem of extending these logics with the ability to perform *default reasoning*, a kind of non-monotonic reasoning that is to be applied whenever the rules involved allow for *exceptions*.

Non-monotonic reasoning has been formally addressed in various ways, leading to the development of a variety of formalisms, most of which belong to the offspring of Doyle and McDermott's Nonmonotonic Logic [6], Reiter's Default Logic [18] and McCarthy's Circumscription [15]. Each of these formalisms may be seen as extending FOL with non-monotonic reasoning capabilities of some kind. Given that TLs may be viewed as (pragmatically and computationally interesting) subsets of FOL, one might be led to think that a simple integration of default and terminological reasoning could be obtained by simply considering one of the non-monotonic formalisms mentioned above and restricting it to deal with the chosen TL, rather than with FOL *tout court*. Unfortunately, these non-monotonic formalisms, besides having unattractive logical and computational properties, suffer from a problem that hinders their use in KR contexts requiring that KB construction be accomplished in an incremental fashion. We call this problem the *Exceptions Explicitation Problem* (EEP).

Incrementality of KB construction is a highly desirable property for KB management systems. Large KBs are the result of an evolutionary process, both because knowledge entry is a time-consuming process and because knowledge may simply become available at later stages of this process. When a large KB is built by this "stepwise refinement" process, it is highly desirable that the refinement consists in the plain, piecemeal *addition* of new knowledge chunks rather than in a time-consuming *revision* (with the possibly ensuing deletion) of pre-existing chunks. The effect of the EEP is exactly that of making purely additive knowledge acquisition impossible.

1.1 The EEP and Default Logic

Let us discuss, by means of a concrete example, what the EEP is in the context of Default Logic (DL)[1]. Let us consider the well-known "penguin" example. It may be represented by means of the DL rule

$$\frac{Bird(x) : Flies(x)}{(x)} \tag{1}$$

("if x is a bird, and it is consistent to assume that it flies, then infer that it flies", i.e. "birds typically fly"). Let us consider an exception to the rule, "penguins are birds that typically do not fly", and let us represent it by means of the following:

$$\forall x \; Penguin(x) \Rightarrow Bird(x) \tag{2}$$

$$\frac{Penguin(x) : \neg Flies(x)}{\neg Flies(x)} \tag{3}$$

There is a problem hidden in this set of expressions: in DL the addition of the assertion $Penguin(opus)$ to the set (1)÷(3) generates two different "sets of conclusions" ("extensions", in DL terminology): in one of them Opus flies while in the other Opus does not fly, depending on whether we "first" consider rule (1) or rule (3), respectively; this is contrary to intuitions, as our knowledge of the animal world makes us strongly prefer the conclusion according to which Opus does not fly.

Hence DL seems, at first sight, inadequate to account for this kind of reasoning, as it generates a situation of ambiguity in the representation of a state of affairs that is all but ambiguous to us[2]. At first sight, it might seem possible to overcome this problem without departing from DL; what one needs to do is *explicitly* consider exceptions in each default rule they involve. Accordingly, a set of expressions that induces a correct behaviour of DL for our example is the following:

$$\frac{Bird(x) : Flies(x) \wedge \neg Penguin(x)}{Flies(x)} \tag{4}$$

[1] To this respect, other formalisms such as Nonmonotonic Logic and Circumscription behave in a completely analogous way, and will therefore not be discussed.

[2] The generation of multiple extensions in DL may otherwise be due to an *inherent* ambiguity of the state of affairs being represented. A very famous example of this is the so-called "Nixon diamond".

$$\forall x\ Penguin(x) \Rightarrow Bird(x)$$

$$\frac{Penguin(x) : \neg Flies(x)}{\neg Flies(x)}$$

At this point, given $Penguin(opus)$, rule (4) does not allow one to conclude $Flies(opus)$; this leaves us with one extension only, an extension that contains only the conclusions we would actually subscribe to.

However, this solution looks like an *ad hoc* patch rather than a real solution. In fact, we would like the conclusion according to which Opus does not fly to already follow from the set (1)÷(3), as these already represent the fact that penguins are exceptional birds as far as the ability to fly is involved; consequently, the revision of rule (1) to yield (4) seems a redundant re-assertion of this fact. Besides, one should note that, until the KB did not contain any reference to penguins, rule (1) was more than adequate. It is the introduction of (2) and (3) that has obliged us to transform rule (1) into (4); in the more general case, the acquisition of new knowledge obliges us to *revise previously acquired knowledge*. It is then evident that the realization of default reasoning by means of DL is undermined by the impossibility to update KBs by simple piecemeal additions of new knowledge chunks. In semantic terms this means the impossibility, unlike in traditional logical formalisms, to characterize the denotational semantics of the modified KB in a compositional way, i.e. as a combination of the denotations of the original KB and of the newly acquired knowledge.

But the need to explicitly represent exceptions brings about other, even more compelling problems, having to do with the way in which, at the time of the introduction of a new item of knowledge into the KB, it is to be determined *which* items are to be modified and which are not. It turns out that this operation cannot be realized by simply performing a number of matchings between the formula to be introduced and the other formulae in the KB; on the contrary, it requires in general *a DL theorem proving operation*. This may clearly be seen by taking a look at our example: the relation between the precondition of rule (1) and the precondition of rule (3) has been determined by relying on the fact that $Bird(x)$ is *derivable* from $Penguin(x)$ through (2); the transformation of rule (1) into rule (4) is then realized in order to "inhibit" not rule (1) in itself, but the inferential chain that would lead us to conclude $Flies(x)$ from $Penguin(x)$. In general, given a DL knowledge base, it is not even determined *a priori* whether there exist one, several or no inferential chains that lead to the undesired conclusion (in our case, one, several or no inferential chains between $Penguin(x)$ and $Flies(x)$); in case several such chains exist, *each of them* has to be inhibited by means of a call to the DL theorem prover.

That the phase of KB construction requires repeated calls to the DL

theorem prover in order to maintain the consistency of the KB that is being built, is in itself rather implausible; but the whole endeavour becomes absurd once we consider that first-order DL is undecidable! This means that, unless the construction of the KB is realized in a completely *static* (non incremental) way, the problem of knowledge acquisition in DL (and, analogously, in the other non-monotonic formalisms mentioned previously) is an *unsolvable* problem.

1.2 The EEP and MINEs

While the general non-monotonic formalisms mentioned above are affected by the EEP, this is not true of the formalisms for *Multiple Inheritance Networks with Exceptions* (MINEs), a popular, albeit less general, class of non-monotonic KR languages oriented to the representation of taxonomic knowledge (see e.g. [1, 2, 5, 10, 11, 12, 21, 22]). Such languages are less expressive than the more general non-monotonic formalisms mentioned above, in the sense that they only allow for monadic predicate symbols, a limited use of negation and no disjunction at all. For our purposes, it is also essential to observe that their monotonic fragment is far less expressive than TLs as, having no term constructors in their syntactic apparatus, they only allow the definition of taxonomies of simple predicate symbols.

MINEs do not suffer from the EEP because they implement an *implicit* handling of exceptions by exploiting the partial ordering given by the taxonomy: as a first approximation we can say that, in case of "conflicts" (i.e. contradictory conclusions being supported), a "default rule" $a \rightarrow b$ ("a's are typically b's") is "preferred" to another default rule $c \not\rightarrow b$ ("c's are typically not b's") if the precondition of the first precedes the precondition of the second in the ordering. In other words, the implicit handling of exceptions obeys the so-called *specialization principle*, according to which conflicts are to be solved by preferring the properties belonging to a subclass over those belonging to a superclass.

In this paper we will show how we can effectively extend TLs in such a way that they allow a brand of default reasoning that obeys the specialization principle, thus creating a logic that combines the tools for describing taxonomic organizations of complex objects which are typical of TLs, the ability to describe default information which is typical of general nonmonotonic formalisms, and the incrementality in KB construction which is typical of MINEs. Such an endeavour constitutes perhaps the first completely formal realization of the notion of "frame" as originally proposed by Minsky [16], a notion that was intended to describe a highly structured aggregate of knowledge allowing the description of "prototypical" knowledge and resulting in KBs of taxonomic form. We will call our logic $\mathcal{TDL}^-$ (Terminological Default Logic—the "-" superscript distinguishes it from an earlier version). The

$\mathcal{TDL}^-$ logic is (intentionally!) a *minimal* logic, in the sense that it includes the minimal set of TL operators and of default rule types that are needed to highlight the problems resulting from the interaction between default knowledge and terminological knowledge. The extension to other TL operators and to other forms of default rules (such as e.g. those involving numeric restrictions) is conceptually easy, but would not tell us much with respect to the issue of the integration of default and terminological reasoning.

This paper is organized as follows. In Section 2 we formally introduce the syntax and the semantics of the monotonic fragment of $\mathcal{TDL}^-$. In Section 3 we deal with the non-monotonic part of $\mathcal{TDL}^-$, describing the notion of "extension" of a set of $\mathcal{TDL}^-$ formulae (i.e. the set of conclusions that may be derived from these formulae) and some of its properties, including some relations with Default Logic. In Section 4 we discuss an algorithm that computes an extension of a set of $\mathcal{TDL}^-$ formulae (when it exists), and discuss the issue of the computational complexity of $\mathcal{TDL}^-$. In Section 5 we argue that $\mathcal{TDL}^-$ effectively contains MINEs, by presenting a translation of MINEs into $\mathcal{TDL}^-$ which preserves semantics. Section 6 concludes.

2 The monotonic fragment of $\mathcal{TDL}^-$

The $\mathcal{TDL}^-$ logic, like many other TLs, allows the specification of three fundamental types of terms: *frames*, *slots* and *individual constants*. Frames (also known as *concepts*) are terms denoting sets of individuals, and are, so to speak, the first-class citizens of $\mathcal{TDL}^-$. Slots (also known as *roles*) are terms denoting binary relations between individuals; their function is to allow the specification of structural constituents of frames. Individual constants denote individuals of the domain of discourse.

From a syntactic point of view, frames (resp. slots) are either unary (resp. binary) predicate symbols or complex terms built by the application of frame-forming (resp. slot-forming) operators to other frames and/or slots[3]. For example, the set of predicate symbols might contain the unary predicate symbols `Triangle`, `Polyhedron` and `Regular_polyhedron`, denoting the sets of triangles, polyhedra and regular polyhedra, respectively, and the binary predicate symbol `Side`, denoting all those pairs of individuals $\langle x, y \rangle$ such that the individual denoted by y is one of the sides of the individual denoted by x. This would allow, in the logic we will introduce next, the definition of more complex frames such as:

- the frame ($\forall$`Side.Triangle`), denoting the set of those individuals whose sides are all triangles (i.e. the set of tetrahedra);

[3]The $\mathcal{TDL}^-$ logic does not actually include slot-forming operators; however, we mention their existence in order to indicate that our framework applies in a straightforward way to TLs that do include such operators.

- the frame ($\neg$`Triangle`), denoting the set of those individuals that are not triangles;
- the frame (`Regular_polyhedron`$\sqcap$($\forall$`Side.Triangle`)), denoting the set of those regular polyhedra whose sides are all triangles (i.e. the set of regular tetrahedra);

and to subsequently define other frames by using those defined before.

In order to introduce the syntax of $\mathcal{TDL}^-$ we will need three disjoint alphabets: an alphabet I of individual constants (with metavariables i, i_1, i_2, ...), an alphabet MP of monadic predicate symbols (with metavariables M, M_1, M_2, ...) and an alphabet DP of dyadic predicate symbols (with metavariables D, D_1, D_2, ...). The syntax of $\mathcal{TDL}^-$ frames is specified by the following definition.

Definition 1 *A* frame *in $\mathcal{TDL}^-$ is defined by the following BNF clauses:*

$$\begin{array}{rcl} F_1, F_2 & \rightarrow & (F_1 \sqcap F_2) \mid M \mid (\neg M) \mid (\forall S.F_1) \mid \bot \mid \top \\ S & \rightarrow & D \end{array}$$

We will use metavariables $F, F_1, F_2, \ldots$ ranging on frames and metavariables $S, S_1, S_2, \ldots$ ranging on slots.

Let us now switch to the formal semantics of $\mathcal{TDL}^-$ frames. The meaning of the linguistic constructs introduced above may be given in terms of the notion of extension function.

Definition 2 *An* interpretation $\mathcal{I}$ *over a nonempty set of individuals $\mathcal{D}$ is a function that maps elements of MP into subsets of $\mathcal{D}$, elements of DP into subsets of $\mathcal{D}\times\mathcal{D}$, and elements of I into elements of $\mathcal{D}$ such that $\mathcal{I}(i_1) \neq \mathcal{I}(i_2)$ whenever $i_1 \neq i_2$. We will say that $\mathcal{I}$ is an* extension function *iff $\mathcal{I}(\bot) = \emptyset$, $\mathcal{I}(\top) = \mathcal{D}$, $\mathcal{I}(\neg M) = \mathcal{D} \setminus \mathcal{I}(M)$, $\mathcal{I}(F_1 \sqcap F_2) = \mathcal{I}(F_1) \cap \mathcal{I}(F_2)$, $\mathcal{I}(\forall S.F) = \{x \in \mathcal{D} \mid \forall y : \langle x, y\rangle \in \mathcal{I}(S) \Rightarrow y \in \mathcal{I}(F)\}$.* ∎

Our language also allows the expression of assertions, stating that individual constants are instances of frames and pairs of individual constants are instances of slots.

Definition 3 *An* assertion *is an expression of the form $F[i]$ or $S[i_1, i_2]$, where i_1, i_2 are elements of I, F is a frame and S is a slot. An extension function $\mathcal{I}$ over a nonempty domain $\mathcal{D}$* satisfies an assertion *$F[i]$ iff $\mathcal{I}(i) \in \mathcal{I}(F)$, and* satisfies an assertion *$S[i_1, i_2]$ iff $\langle \mathcal{I}(i_1), \mathcal{I}(i_2)\rangle \in \mathcal{I}(S)$.* ∎

We will use metavariables $\alpha, \alpha_1, \alpha_2, \ldots$ ranging on assertions and metavariables $A, A_1, A_2, \ldots$ ranging on sets of assertions. For example, the following expressions are assertions.

$(\texttt{Polyhedron} \sqcap (\neg \texttt{Regular\_polyhedron}))\ [\texttt{poly034726}]$
$\texttt{Triangle}\ [\texttt{trian6253265}]$
$\texttt{Side}\ [\texttt{trian6253265}, \texttt{poly034726}]$

We next introduce a feature of the language that allows us to associate names to complex frames, with the net effect that we will be able to define new frames using these names instead of the corresponding complex frames.

Definition 4 *A* naming *is an expression of the form $M \doteq F$ or of the form $M \lessdot F$, where F is a frame and M an element of MP; in either case, we will say that M is the name of F. An extension function $\mathcal{I}$ over a nonempty domain $\mathcal{D}$* satisfies a naming $M \doteq F$ *iff $\mathcal{I}(M) = \mathcal{I}(F)$, and* satisfies a naming $M \lessdot F$ *iff $\mathcal{I}(M) \subseteq \mathcal{I}(F)$.* ∎

We will use metavariables $\eta, \eta_1, \eta_2, \ldots$ ranging on namings and metavariables $N, N_1, N_2, \ldots$ ranging on sets of namings. For example, the following expressions are namings.

$$\begin{array}{rcl} \texttt{Irregular\_polyhedron} & \doteq & (\texttt{Polyhedron} \sqcap (\neg \texttt{Regular\_polyhedron})) \\ \texttt{Regular\_tetrahedron} & \doteq & (\texttt{Regular\_polyhedron} \sqcap (\forall \texttt{Side.Triangle})) \\ \texttt{Regular\_polygon} & \lessdot & \texttt{Polygon} \end{array}$$

Namings of type $M \doteq F$ define necessary and sufficient conditions for an individual to be in the denotation of M, while the conditions defined by namings of type $M \lessdot F$ are necessary but not sufficient.

Hence, in a TL knowledge is represented by either namings or assertions. The notion of a *Terminological Knowledge Base* (or *T-set*, for short) can then be specified as follows.

Definition 5 *A* T-set *is a pair $\Omega = \langle A, N \rangle$, where A is a set of assertions and N is a set of namings.* ∎

We will use metavariables $\Omega, \Omega_1, \Omega_2, \ldots$ and $\Psi, \Psi_1, \Psi_2, \ldots$ ranging on T-sets[4]. The notion of *satisfiability* of a T-set Ω, and that of a *model* of a T-set Ω, may be defined as follows.

Definition 6 *Let $\Omega = \langle A, N \rangle$ be a T-set, and $\mathcal{I}$ an extension function over a nonempty domain $\mathcal{D}$. $\mathcal{I}$ is a* model *of Ω iff $\mathcal{I}$ satisfies all the assertions in A and all the namings in N. A T-set Ω is* satisfiable *iff it has a model (otherwise, we say Ω is* unsatisfiable*).* ∎

[4]For brevity of notation, in the rest of the paper we will sometimes consider a T-set Ω as a set of assertions and namings, rather than as an ordered pair $\langle A, N \rangle$ of sets; we will thus write $\Omega \cup \alpha$ as short for $\langle A \cup \{\alpha\}, N \rangle$, and $\Omega \cup \eta$ a short for $\langle A, N \cup \{\eta\} \rangle$.

We are now ready to introduce the most important notions of our TL, those of *subsumption* and *support*, two notions that are conceptually analogous to the notion of "logical consequence" in standard logics. Intuitively, given two monadic predicate symbols M_1 and M_2 and a T-set Ω, we say that M_1 *subsumes* M_2 in Ω when Ω allows us to draw the conclusion that M_1 is "more general" than M_2. We say that a T-set Ω *supports* an assertion α when Ω allows us to draw the conclusion that α.

Definition 7 *Let Ω be a satisfiable T-set, and let M_1, M_2 be two elements of MP. We say that M_1 is* subsumed *by M_2 in Ω (written $M_1 \preceq_\Omega M_2$) iff for every model $\mathcal{I}$ of Ω it is true that $\mathcal{I}(M_1) \subseteq \mathcal{I}(M_2)$. We say that Ω* supports *an assertion α (written $\Omega \models \alpha$) iff α is satisfied by all models of Ω.* ∎

Finally, in order to better understand the nonmonotonic extension of our logic, we need to introduce the notion of the *transitive closure* of a T-set Ω.

Definition 8 *Let $\Omega = \langle A, N\rangle$ be a satisfiable T-set. The* transitive closure *of Ω, written $TC(\Omega)$, is the set $\Omega' = \langle A \cup \{\alpha \mid \Omega \models \alpha\}, N\rangle$.* ∎

It is easy to show that TC is monotonic (i.e. if $\Omega \subseteq \Omega'$, then $TC(\Omega) \subseteq TC(\Omega')$), and that TC is in fact a closure (i.e. $TC(\Omega) = TC(TC(\Omega))$).

3 Default reasoning in $\mathcal{TDL}^-$

Up to now we have described the monotonic fragment of $\mathcal{TDL}^-$. Let us now discuss the addition of non-monotonic features.

Definition 9 *A* default *is an expression of the form $M \mapsto S.F$, where M is an element of MP, S is a slot and F is a frame.* ∎

We will use metavariables $\delta, \delta_1, \delta_2, \ldots$ ranging on defaults and metavariables $\Delta, \Delta_1, \Delta_2, \ldots$ ranging on sets of defaults.

Informally, $M \mapsto S.F$ means: "if i is an M such that i' is an S-filler of i and the assumption that i' is an F does not lead to a contradiction, assume it". For example, the default `IU` $\mapsto$ `FM.I` means: "if i is an Italian university, i' is one of its faculty members, and the assumption that i' is Italian does not lead to a contradiction (i.e. we do not know that he is not an Italian), assume it".

The particular syntax we have chosen for defaults is due to the following reasons:

1. an analysis of the literature concerning the interaction between frames and default knowledge, ranging from the more informal and "impressionistic" proposals (such as those of e.g. [16, 19]) to more formally motivated ones [4, 17], reveals that default rules with consequents in a "slot-filler" form have always been identified as the most natural way in which to convey default frame-like knowledge[5];

2. this type of default rules is sufficient to highlight the problems resulting from the interaction between default knowledge and terminological knowledge; on the other hand, the extension to other forms of defaults (such as e.g. those involving numeric restrictions) is conceptually easy, but does not teach us much with respect to the issue of the integration between default and terminological reasoning.

We may now define what we mean by a $\mathcal{TDL}^-$ theory.

Definition 10 *A $\mathcal{TDL}^-$* theory *is a pair $\mathcal{T} = \langle \Psi, \Delta \rangle$, where Ψ is a satisfiable finite T-set, and Δ is a finite set of defaults.* ∎

We are now able to define what the *extensions* of a $\mathcal{TDL}^-$ theory are. Informally, by the term "extension" we mean the set of assertions and namings that we can reasonably believe to be true as a consequence of the theory. For instance, if we knew that the University of Bellosguardo is an Italian university, that Professor Dolcevita is one of its faculty members, and that the faculty members of Italian universities are typically Italian, we would like to conclude (formally: to be included in the corresponding extension) that Dolcevita is an Italian.

Our definition of "extension" is similar to the one given by Reiter for Default Logic, i.e. an extension is a fixpoint of a consequence relation. However, unlike in Default Logic, "wired" in our definition is the specialization principle: in the presence of conflicting defaults, the one with the more specific premise will be preferred. For instance, suppose that, besides the fact that the faculty members of Italian universities are typically Italian, we also knew that the faculty members of South Tyrolean universities are typically not Italian; knowing that South Tyrolean universities are Italian universities[6], that the University of Pflunders is a South Tyrolean university, and that Professor Katzenjammer is one of its faculty members, we will be able to derive, as desired, that Katzenjammer is not Italian. It is crucial to note that *such a conclusion could not be drawn if we simply confined ourselves to*

[5]Note that the availability of the "naming" mechanism is such that it is "virtually" possible to express default rules of type $F_1 \mapsto S.F_2$; this is accomplished by first defining m to be equivalent to F_1 by means of a naming $M = F_1$, and then using the default $M \mapsto S.F_2$.

[6]South Tyrol is, in fact, a German-speaking region of Italy.

employing the terminological subset of Default Logic (or, for that matter, of any general non-monotonic formalism): the specialization principle embodied in Definition 11 plays a critical role in the inferential behaviour displayed by our formalism.

Definition 11 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory. Let Θ be an operator such that, for any satisfiable T-set Ω, $\Theta(\Omega, \mathcal{T})$ is the smallest satisfiable T-set satisfying the following closure conditions:*

1. $\Psi \subseteq \Theta(\Omega, \mathcal{T})$;
2. $\Theta(\Omega, \mathcal{T}) = TC(\Theta(\Omega, \mathcal{T}))$;
3. *for all defaults $M_1 \mapsto S.F_1 \in \Delta$, for all the assertions $M_1[i_1] \in \Theta(\Omega, \mathcal{T})$ such that $S[i_1, i_2] \in \Theta(\Omega, \mathcal{T})$, it happens that $F_1[i_2] \in \Theta(\Omega, \mathcal{T})$ unless there exists a unary predicate symbol M_2 such that*
   (a) $M_2[i_1] \in \Omega$, $M_2 \preceq_\Omega M_1$;
   (b) $M_2 \mapsto S.F_2 \in \Delta$;
   (c) $\Omega \cup \{(F_1 \sqcap F_2)[i_2]\}$ *is unsatisfiable.*

A satisfiable T-set $\mathcal{E}$ is an extension *of the $\mathcal{TDL}^-$ theory $\mathcal{T}$ iff $\mathcal{E} = \Theta(\mathcal{E}, \mathcal{T})$, i.e. iff $\mathcal{E}$ is a fixpoint of the operator Θ.* ∎

Conditions (1.) and (2.) are obviously to be satisfied if we want "extensions" to be "sets of conclusions" according to the sense generally accepted in KR. Condition (3.) embodies the specialization principle: if a default $M_1 \mapsto S.F_1$ is "applicable" and (3a.)÷(3c.) there is no evidence contradicting the conclusion of the default (i.e. i_1 does not belong to any subclass of M_1 which is the premise of a default whose conclusion would be inconsistent with F_1), then the default may be safely applied and the conclusion drawn.

We now consider an example to show how Definition 11 works, and, in particular, how $\mathcal{TDL}^-$ employs an implicit handling of exceptions.

Example 1 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be the $\mathcal{TDL}^-$ theory that formalizes our "Professors" example, with $\Psi = \langle \{\mathtt{IU[b]}, \mathtt{FM[b,d]}, \mathtt{STU[p]}, \mathtt{FM[p,k]}\}, \{\mathtt{STU} \lessdot \mathtt{IU}\} \rangle$ and $\Delta = \{\mathtt{IU} \mapsto \mathtt{FM.I}, \mathtt{STU} \mapsto \mathtt{FM.}(\neg \mathtt{I})\}$. Let $\mathcal{E} = TC(\Psi \cup \{(\neg \mathtt{I})[\mathtt{k}], \mathtt{I[d]}\})$. It is not hard to show that $\Theta(\mathcal{E}, \mathcal{T}) = \mathcal{E}$ and that $\mathcal{E}$ satisfies the conditions of Definition 11; therefore $\mathcal{E}$ is an extension of $\mathcal{T}$.*

It is important to observe that the same example may be formalized (for example) in Default Logic only at the price of a cumbersome operation of "exceptions explicitation", i.e. by imposing the following defaults and set of axioms.

$$\frac{IU(x) \wedge FM(x,y) : I(y) \wedge \neg STU(x)}{I(y)} \tag{5}$$

$$\frac{STU(x) \wedge FM(x,y) : \neg I(y)}{\neg I(y)} \tag{6}$$

$$\forall x \; STU(x) \Rightarrow IU(x) \tag{7}$$

$$IU(b) \wedge FM(b,d) \wedge STU(p) \wedge FM(p,k) \tag{8}$$

As rule (5) shows, in Default Logic we must make explicit the fact that a South Tyrolean university is an exceptional Italian university with respect to the citizenship of its faculty members; in $\mathcal{TDL}^-$ this is not necessary, and this, as hinted in Section 1, allows KB update to be completely additive. ∎

We can use Example 1 to show that $\mathcal{TDL}^-$ is in fact non-monotonic.

Proposition 1 *$\mathcal{TDL}^-$ is non-monotonic, i.e. there exists a $\mathcal{TDL}^-$ theory $\mathcal{T} = \langle \Psi, \Delta \rangle$, an extension $\mathcal{E}$ of $\mathcal{T}$, a T-set Ψ' and a set of defaults Δ', such that for no extension $\mathcal{E}'$ of $\mathcal{T}' = \langle \Psi \cup \Psi', \Delta \cup \Delta' \rangle$ it is true that $\mathcal{E} \subseteq \mathcal{E}'$.*

Proof. *Let $\mathcal{T}$ be the $\mathcal{TDL}^-$ theory of Example 1 and $\mathcal{E}$ its (unique) extension. Let $\Psi' = \{\mathtt{STU}[\mathtt{b}]\}$ and $\Delta' = \emptyset$. It is easy to see that the only extension of $\mathcal{T}' = \langle \Psi \cup \Psi', \Delta \cup \Delta' \rangle$ is $\mathcal{E}' = TC(\Psi \cup \Psi' \cup \{(\neg \mathtt{I})[\mathtt{k}], (\neg \mathtt{I})[\mathtt{d}]\})$. Therefore $\mathcal{E} \not\subseteq \mathcal{E}'$.* ∎

We go on to discuss some properties of the notion of extension as formalized in Definition 11. The following proposition parallels the one given by Reiter for Default Logic, stating that extensions are "maximal" sets.

Proposition 2 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory, and let $\mathcal{E}_1$ and $\mathcal{E}_2$ be extensions of $\mathcal{T}$. If $\mathcal{E}_1 \subseteq \mathcal{E}_2$, then $\mathcal{E}_1 = \mathcal{E}_2$.* ∎

In some cases, there exists an easy one-to-one relation between our notion of extension and that of Default Logic.

Definition 12 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory. Let* the set of conflicts *$C_{\mathcal{T}}$ of $\mathcal{T}$ be the set*

$$C_{\mathcal{T}} \;=\; \bigcup_S \{\langle M_1, M_2 \rangle \mid \begin{array}{l} M_1 \mapsto S.F_1 \in \Delta, M_2 \mapsto S.F_2 \in \Delta, \\ (F_1 \sqcap F_2) \textit{ is unsatisfiable,} \\ \textit{for some } i_1, i_2 \; S[i_1, i_2] \in \Psi, \\ M_1 \preceq_\Psi M_2\} \end{array}$$

We will say that $\mathcal{T}$ is conflict-free *iff $C_{\mathcal{T}} = \emptyset$.* ∎

Conflict-free $\mathcal{TDL}^-$ theories are interesting because they can be translated into normal default theories of Default Logic:

Definition 13 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a conflict-free $\mathcal{TDL}^-$ theory. Its translation into Default Logic is the normal default theory $\mathcal{T}_{DL} = \langle \Psi_{DL}, \Delta_{DL} \rangle$, where Ψ_{DL} is the obvious translation of namings and assertions into first order logic and Δ_{DL} is obtained by translating defaults $M \mapsto S.F \in \Delta$ into normal defaults*

$$\frac{M(x) \wedge S(x,y) \ : \ F'(x)}{F'(x)} \tag{9}$$

where F' is the obvious translation of the frame F into first order logic. ∎

The following proposition shows the relation between conflict-free $\mathcal{TDL}^-$ theories and Default Logic theories.

Proposition 3 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a conflict-free $\mathcal{TDL}^-$ theory and $\mathcal{T}_{DL}$ its translation into Default Logic. Then $\mathcal{E}$ is an extension of $\mathcal{T}$ iff $Tcl(\mathcal{E}_{DL})$ is an extension of $\mathcal{T}_{DL}$, where Tcl is the transitive closure function over sets of first order formulas and $\mathcal{E}_{DL}$ is the translation of $\mathcal{E}$ into first order logic.* ∎

Since every normal Default Logic theory has an extension, from Proposition 3 it follows immediately that:

Proposition 4 *Every conflict-free $\mathcal{TDL}^-$ theory has an extension.* ∎

Note that it is not easy to give a translation for generic $\mathcal{TDL}^-$ theories into Default Logic, since we would need to explicit the exceptions, as rule (5) shows. Furthermore, it is not easy to discover the exceptions. Consider in fact the T-set $\langle \{M_1[i],\ M_2[i],\ S_1[i,i],\ S_2[i,i],\ M_2 \lessdot M_1\},\ \{M_1 \mapsto S_1.(\forall S_2.(\neg M_3)),\ M_2 \mapsto S_1.M_3\} \rangle$. The default $M_2 \mapsto S_1.M_3$ is exceptional to the default $M_1 \mapsto S_1.(\forall S_2.(\neg M_3))$ in a non-obvious way. Therefore, it seems that in order to give a translation into Default Logic we need to compute an entire extension, and only after this we may be able to discover the exceptions.

Similarly to what happens in most non-monotonic formalisms, some $\mathcal{TDL}^-$ theories have more than one extension; in particular, the number of extensions can be exponential with respect to the size of the $\mathcal{TDL}^-$ theory, as the following proposition shows.

Proposition 5 *There exists a conflict-free $\mathcal{TDL}^-$ theory $\mathcal{T}$ such that the number of extensions of $\mathcal{T}$ is exponential with respect to the size of $\mathcal{T}$.*

Proof. *This proposition is proven by considering the $\mathcal{TDL}^-$ theory $\mathcal{T} = \langle \Psi, \Delta \rangle$ (which we will call* multiple Nixon Diamond*).*

$$\Psi = \langle \{M'_k[i_j] \mid 1 \leq j \leq n, 1 \leq k \leq 2\} \cup \{S_k[i_j, i_j] \mid 1 \leq j, k \leq n\}, \emptyset \rangle$$

$$\Delta = \{M'_1 \mapsto S_j.M_j \mid 1 \leq j \leq n\} \cup \{M'_2 \mapsto S_j.(\neg M_j) \mid 1 \leq j \leq n\}$$

$\mathcal{T}$ contains n^2 embedded Nixon Diamonds. Since for every Nixon Diamond we have two possibilities (i.e. whether the corresponding S's of the i's are M's or not), $\mathcal{T}$ has 2^{n^2} extensions. Therefore, considering that $|\mathcal{T}| = |\Psi| + |\Delta|$ is $O(n)$, the number of extensions of $\mathcal{T}$ is exponential with respect to the size of $\mathcal{T}$. ∎

Although the number of extensions can be very large in the worst case, as the multiple Nixon Diamond example shows, in actual knowledge representation applications this number need not be computationally discouraging. Furthermore, we might observe that this exponential number of extensions is not a characteristic of $\mathcal{TDL}^-$ itself, but *is rather a common characteristic of almost all non-monotonic formalisms.* The multiple Nixon Diamond can be easily formulated in these formalisms, and gives rise to the same phenomenon; for example, the set of formulae and rules of Default Logic:

$$A_j(a_i) \qquad \text{for } 1 \leq i \leq n \text{ and } 1 \leq j \leq 2$$

$$S_j(a_i, a_i) \qquad \text{for } 1 \leq i, j \leq n$$

$$\frac{A_1(x) : P_i(x)}{P_i(x)} \qquad \text{for } 1 \leq i \leq n$$

$$\frac{A_2(x) : \neg P_i(x)}{\neg P_i(x)} \qquad \text{for } 1 \leq i \leq n$$

gives rise to the same number of extensions. Unfortunately, it is also true that:

Proposition 6 *There exists a $\mathcal{TDL}^-$ theory with no extensions.* ∎

This is proven by showing that the sample theory $\mathcal{T} = \langle \Psi, \Delta \rangle$, with $\Psi = \langle \{M_1[i], S[i,i]\}, \{M_1 \doteq (F_1 \sqcap F_2), M_2 \doteq (F_1 \sqcap (F_2 \sqcap F_3))\} \rangle$ and $\Delta = \{M_1 \mapsto S.F_3, M_2 \mapsto S.\bot\}$, has no extensions. Our studies show that it would not be easy to find sublanguages of $\mathcal{TDL}^-$ such that the existence of at least one extension is always guaranteed: in fact, removing from $\mathcal{TDL}^-$ the causes that are responsible for the non-existence of an extension for theory $\mathcal{T}$ would dramatically curtail the expressive power of the language itself.

4 Computing an extension of a $\mathcal{TDL}^-$ theory

In this section we will discuss the properties of the EXT (nondeterministic) algorithm that computes (when it exists) an extension of a $\mathcal{TDL}^-$ theory.

4.1 Correctness and completeness issues

The EXT algorithm is heavily dependent on the decision of the monotonic fragment of $\mathcal{TDL}^-$, i.e. of the $\preceq_\Omega$ and $\models$ relations. It is well-known (see [9]) that in most TLs (and the monotonic fragment of $\mathcal{TDL}^-$ is no exception), deciding $\preceq_\Omega$ can be reduced to the decision of $\models$, and that the decision of $\models$ can in turn be reduced to deciding if Ω is unsatisfiable.

There exists a well-known technique, based on constraint propagation (see [9]), for deciding unsatisfiability in TLs. By using this technique, it may be shown that it is decidable whether a finite and "acyclic" T-set (i.e. a T-set that contains no namings in which the *definiendum* is defined, either directly or indirectly, in terms of itself) is unsatisfiable. By considering acyclic $\mathcal{TDL}^-$ theories only, we can profitably exploit this result.

Definition 14 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory. Then $\mathcal{T}$ is* acyclic *iff $\Phi = \{\eta \mid \eta$ is a naming in $\Psi\} \cup \{M{<}F \mid M \mapsto S.F \in \Delta\}$ is acyclic.* ∎

Definition 15 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory and let i_1, i_2 be individual constants. An* instantiated default *of $\mathcal{T}$ is a triple $\gamma = \langle i_1, i_2, \delta \rangle$ with $\delta = M \mapsto S.F$ and $\delta \in \Delta$. We also define the function Consequent to be such that Consequent$(\gamma) = F[i_2]$.* ∎

We will use metavariables $\gamma, \gamma_1, \gamma_2, \ldots$ ranging on instantiated defaults and metavariables $\Gamma, \Gamma_1, \Gamma_2, \ldots$ ranging on sets of instantiated defaults.

Definition 16 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory, Ω a set of namings and assertions, and $\gamma = \langle i_1, i_2, M_1 \mapsto S.F_1 \rangle$ an instantiated default of $\mathcal{T}$. Then γ is an* applicable default *of $\mathcal{T}$ in Ω iff*

1. *$\Omega \models M_1[i_1]$;*
2. *$S[i_1, i_2] \in \Psi$;*
3. *for all M_2 such that $M_2 \preceq_\Omega M_1$, $M_1 \not\preceq_\Omega$ and $\langle i_1, i_2, M_2 \mapsto S.F_2 \rangle$ is an instantiated default of $\mathcal{T}$, with $\Omega \models M_2[i_1]$, it is true that $\Omega \cup \{(F_1 \sqcap F_2)[i_2]\}$ is satisfiable.* ∎

We will write $A_\mathcal{T}(\Omega)$ as short for the set of applicable defaults of $\mathcal{T}$ in Ω.

Given an acyclic $\mathcal{TDL}^-$ theory $\mathcal{T} = \langle \Psi, \Delta \rangle$ and a satisfiable, finite and acyclic set of namings and assertions Ω, Δ and Ψ are finite sets; hence, the following proposition holds.

Proposition 7 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory and Ω a finite and acyclic set of namings and assertions. Then the set of instantiated defaults of $\mathcal{T}$ and $A_\mathcal{T}(\Omega)$ are recursive sets.* ∎

Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory;
begin
  $\Omega_0 \leftarrow \Psi; n \leftarrow 0;$
  repeat
    $n \leftarrow n+1; \omega_0 \leftarrow \Psi; \Gamma_0 \leftarrow \emptyset; i \leftarrow 0;$
    repeat
      $A^i_{\mathcal{T}} \leftarrow (A_{\mathcal{T}}(\omega_i) \cap A_{\mathcal{T}}(\Omega_{n-1})) \setminus \Gamma_i;$
      if not $empty(A^i_{\mathcal{T}})$
        then choose γ **from** $A^i_{\mathcal{T}}$;
          $\Gamma_{i+1} \leftarrow \Gamma_i \cup \{\gamma\};$
          $\omega_{i+1} \leftarrow \omega_i \cup \{Consequent(\gamma)\};$
      endif
      $i \leftarrow i+1;$
    until $empty(A^{i-1}_{\mathcal{T}});$
    $\Omega_n \leftarrow \omega_{i-1};$
  until $\Omega_n = \Omega_{n-1};$
end

Figure 1: The EXT algorithm computes an extension of a $\mathcal{TDL}^-$ theory.

Let us now discuss the EXT algorithm for computing extensions. The extension is built by a series of subsequent approximations. Each approximation Ω_n is built from the first component Ψ of an acyclic $\mathcal{TDL}^-$ theory $\mathcal{T}$ by using applicable defaults, one at time. At each step, the instantiated default to be applied is chosen from those which have not yet been considered and which were applicable in the previous approximation and still are in the current state of the current approximation. When no more instantiated defaults are applicable, the algorithm continues with the next approximation. If two successive approximations are the same set, the algorithm is said to *converge*.

The choice of which instantiated default to apply at each step of the inner loop introduces in general a degree of nondeterminism. Generality requires this nondeterminism, since $\mathcal{TDL}^-$ theories need not have unique extensions (see Proposition 5). The algorithm is detailed in Figure 1.

To see how this algorithm works, let us consider the following example.

Example 2 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory, with*

$$\Psi = \{M_1[i], F_1[i], S[i,i], M_2 \doteq (F_1 \sqcap F_2)\}$$
$$\Delta = \{M_1 \mapsto S.F_2, M_2 \mapsto S.F_3\}$$

Let $\gamma_1 = \langle i, i, M_1 \mapsto S.F_2 \rangle$ and $\gamma_2 = \langle i, i, M_2 \mapsto S.F_3 \rangle$ be instantiated defaults of $\mathcal{T}$. The application of the EXT algorithm to $\mathcal{T}$ would yield the following sequence of steps:

1. $\Omega_0 = \omega_0 = \Psi, \Gamma_0 = \emptyset, A_{\mathcal{T}}(\omega_0) = A_{\mathcal{T}}(\Omega_0) = A^0_{\mathcal{T}} = \{\gamma_1\}$;

2. $\omega_1 = \omega_0 \cup \{Consequent(\gamma_1)\}, \Gamma_1 = \{\gamma_1\}$;

3. $A_{\mathcal{T}}(\omega_1) = \{\gamma_1, \gamma_2\}, A^1_{\mathcal{T}} = \emptyset$;

4. $\Omega_1 = \omega_1$;

5. $\omega_0 = \Psi, \Gamma_0 = \emptyset, A_{\mathcal{T}}(\omega_0) = A^0_{\mathcal{T}} = \{\gamma_1\}, A_{\mathcal{T}}(\Omega_1) = \{\gamma_1, \gamma_2\}$;

6. $\omega_1 = \omega_0 \cup \{Consequent(\gamma_1)\}, \Gamma_1 = \{\gamma_1\}$;

7. $A_{\mathcal{T}}(\omega_1) = \{\gamma_1, \gamma_2\}, A^1_{\mathcal{T}} = \{\gamma_2\}$;

8. $\omega_2 = \omega_1 \cup \{Consequent(\gamma_2)\}, \Gamma_1 = \{\gamma_1, \gamma_2\}$;

9. $A_{\mathcal{T}}(\omega_2) = \{\gamma_1, \gamma_2\}, A^2_{\mathcal{T}} = \emptyset$;

10. $\Omega_2 = \omega_2$;

⋮ *a sequence similar to (5)-(10)*

16. $\Omega_3 = \Omega_2 = \Psi \cup \{F_2[i], F_3[i]\}$. ∎

The following correctness and completeness theorem states that all and only the extensions of a $\mathcal{TDL}^-$ theory $\mathcal{T}$ can be computed by the algorithm. Before proceeding to its proof, we need to state

Lemma 1 *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory with extension $\mathcal{E}$, and let Γ be a set of instantiated defaults of $\mathcal{T}$. Let us define $Consequent(\Gamma)$ to be equal to $\bigcup_{\gamma \in \Gamma}\{Consequent(\gamma)\}$. Then $\mathcal{E} = TC(\Psi \cup Consequent(A_{\mathcal{T}}(\mathcal{E})))$.* ∎

which can be shown by proceeding as for Reiter's Theorem 2.5, described in [18]. We can now state and prove the main theorem.

Theorem 1 (Correctness and completeness) *Let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory. $\mathcal{E}$ is an extension of $\mathcal{T}$ iff the application of the* EXT *algorithm to $\mathcal{T}$ has a converging computation such that $\Omega_n = \Omega_{n-1}$ and $TC(\Omega_n) = \mathcal{E}$.*

Proof.

("if") *Consider the last current state ω_{n_i}. It follows that*

1. $\omega_{n_i} = \Omega_n = \Omega_{n-1}$ *and* $TC(\Omega_n) = \mathcal{E}$;

2. $A^{n_i}_{\mathcal{T}} = \emptyset = (A_{\mathcal{T}}(\omega_{n_i}) \cap A_{\mathcal{T}}(\Omega_{n-1})) \setminus \Gamma_{n_i}$;

3. $\mathcal{E} = TC(\Psi \cup Consequent(A_{\mathcal{T}}(\mathcal{E})))$.

From $A_{\mathcal{T}}^{n_i} = \emptyset$ it follows that $A_{\mathcal{T}}(\mathcal{E}) \subseteq \Gamma_{n_i}$. We will show that $\mathcal{E}$ is an extension of $\mathcal{T}$ by proving that $\Theta(\mathcal{E}, \mathcal{T}) = \mathcal{E}$. By construction and from the definition of TC it follows that point (1.) and point (2.) of Definition 11 hold for $\mathcal{E}$.

Let us first prove that $\mathcal{E} \subseteq \Theta(\mathcal{E}, \mathcal{T})$. Suppose instead that $\mathcal{E} \not\subseteq \Theta(\mathcal{E}, \mathcal{T})$. Therefore, by point (3.), there exists $\gamma \in A_{\mathcal{T}}(\mathcal{E})$ such that $Consequent(\gamma) \in \mathcal{E}$ and $Consequent(\gamma) \notin \Theta(\mathcal{E}, \mathcal{T})$. From Definition 11 it follows that if $\gamma \in A_{\mathcal{T}}(\mathcal{E})$, then $Consequent(\gamma) \in \Theta(\mathcal{E}, \mathcal{T})$; a contradiction. Let us now prove that $\Theta(\mathcal{E}, \mathcal{T})) \subseteq \mathcal{E}$. Suppose instead that $\Theta(\mathcal{E}, \mathcal{T})) \not\subseteq \mathcal{E}$. Therefore, there exists $\gamma \in A_{\mathcal{T}}(\mathcal{E})$ such that $Consequent(\gamma) \in \Theta(\mathcal{E}, \mathcal{T})$, but $Consequent(\gamma) \notin \mathcal{E}$; a contradiction, since point (3.) holds.

("only if") *Let $\mathcal{E}$ be an extension of $\mathcal{T}$. By Lemma 1, it is true that $\mathcal{E} = TC(\Psi \cup Consequent(A_{\mathcal{T}}(\mathcal{E})))$. Since, by Definition 11, $\Theta(\mathcal{E}, \mathcal{T})) = \mathcal{E}$ is minimal, there exists a sequence $\Omega_0, \ldots, \Omega_{n-1}$, such that $\Omega_0 = \Psi$, and, for $1 \leq i \leq n-1$, $\Omega_{i-1} \subseteq \Omega_i \subseteq \Psi \cup Consequent(A_{\mathcal{T}}(\mathcal{E}))$, and if $Consequent(\gamma) \in \mathcal{A}_i$, then $\gamma \in A_{\mathcal{T}}(\Omega_{i-1})$. Now it easy to see that we can choose adequately $\gamma \in A_{\mathcal{T}}(\mathcal{E})$ such that the sequence $\Omega_0, \ldots, \Omega_{n-1}, \Omega_n$, with $\Omega_{n-1} = \Omega_n$, represents a converging series of approximations of the EXT algorithm.* ∎

The following is an example of a non-converging computation.

Example 3 *Consider the acyclic $\mathcal{TDL}^-$ theory $\mathcal{T}$ of Proposition 6. It turns out that each approximation Ω_i is such that $\Omega_{2k} = \Psi$ and $\Omega_{2k+1} = \Psi \cup \{F_3[i]\}$, for each $k \geq 0$. Therefore $\Omega_{2k} \neq \Omega_{2k+1}$ for each $k \geq 0$, and the computation never stops.* ∎

The next corollary follows from Theorem 1.

Corollary 1 *The set of extensions of an acyclic $\mathcal{TDL}^-$ theory is recursively enumerable.* ∎

4.2 Complexity issues

Finally, we discuss some issues related to the computational complexity of $\mathcal{TDL}^-$. In order to do so, first of all we will restrict our attention to TLs for which deciding if an instantiated default is applicable is computable in polynomial time; for all the other TLs, the problem of computing an extension is obviously intractable. For our purposes, let $\mathcal{T} = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory and F a frame. *F is a simple frame wrt Ψ* iff F is a conjunction of unary predicate symbols or negated unary predicate symbols which do not appear as names in Ψ. We will say that *$\mathcal{T}$ is a restricted $\mathcal{TDL}^-$ theory* iff

1. Ψ contains only assertions of the form $F[i]$, where F is a simple frame wrt Ψ;

2. Ψ contains only namings of the form $M \doteq F$, where F is a unary predicate symbol (or the negation of a unary predicate symbol) which is not a name in Ψ, or axioms of the form $M \lessdot F$, where M is not a name in Ψ;

3. Δ is a set of defaults of the form $M \mapsto S.F$, where F is a simple frame wrt Ψ.

It may be shown that, in the case of restricted $\mathcal{TDL}^-$ theories, deciding if an instantiated default is applicable is computable in polynomial time.

Unfortunately, notwithstanding the restrictions on $\mathcal{TDL}^-$ theories, the problem of computing an extension is a computationally hard problem.

Theorem 2 *Finding an extension of a restricted $\mathcal{TDL}^-$ theory (or determining that it has no extension) is NP-Hard.*

Proof. *The proof of this theorem depends on the following reduction of the NP-Complete 3SAT problem to the problem of determining an extension. For a propositional formula σ in 3CNF consider the restricted $\mathcal{TDL}^-$ theory $\mathcal{T}_\sigma$ obtained exactly from the expressions which appear in the following rules.*

1. *for every symbol P which occurs in σ, the following expressions appear in T_σ, where A is a new unary predicate symbol and i an individual constant: $A \mapsto P$, $A \mapsto (\neg P)$, $A[i]$;*

2. *for each clause $X \vee Y \vee Z$ of σ, the following expressions appear in $\mathcal{T}_\sigma$, where A, B, F and F_{xyz} are new unary predicate symbols: $A \doteq (\neg X)$ (also, we will substitute P for $(\neg(\neg P))$), $A \mapsto (F_{xyz} \sqcap ((\neg Y) \sqcap (\neg Z)))$, $F_{xyz} \mapsto (F \sqcap (\neg B))$, $F \lessdot F_{xyz}$ and $F \mapsto B$.*

We will show that σ is satisfiable iff T_σ has an extension. As a consequence, the problem of determining if a restricted $\mathcal{TDL}^-$ theory has an extension is also NP-Hard.

("if") *Suppose that T_σ has an extension $\mathcal{E}$. By rule (1.) it follows that every unary predicate symbol in σ or its negation appears in $\mathcal{E}$. Furthermore, it must not be the case that $F[i] \in \mathcal{E}$ because otherwise rules of type (2.) would have the consequence that $B[i] \in \mathcal{E}$ and thus no expression introduced by rules of type (2.) could create an assertion of type $F[i]$. Therefore, no set of expressions introduced by rules of type (2.), corresponding to clauses in σ, can all be applicable in $\mathcal{E}$. Therefore, every clause of σ is satisfied by a model which "agrees" with $\mathcal{E}$.*

("only if") *Suppose M is a model of σ. Let Ω be the set of assertions $X[i]$ such that M is a model of the literal X, together with $A[i]$. Then $\mathcal{E} = TC(\Omega)$ is an extension of $\mathcal{T}_\sigma$. Note that $\mathcal{E}$ is dependent on expressions introduced by rule (1.) and that none of expressions introduced by rule (2.) apply.* ∎

5 $\mathcal{TDL}^-$ and its relationships with MINEs

In this section we will discuss the relationships between $\mathcal{TDL}^-$ and the formalisms for Multiple Inheritance Networks with Exceptions (MINE). In particular, we will show that there exists a translation Ξ of a MINE Υ into a $\mathcal{TDL}^-$ theory $\Xi(\Upsilon)$ such that a formula $x \rightsquigarrow y$ of the MINE formalism belongs to a conclusion set $\mathcal{S}$ of Υ if and only if $\Xi(\Upsilon)$ has an extension $\mathcal{E}$ to which $\Xi(x \rightsquigarrow y)$ belongs.

5.1 Our MINE Language

Our language and inferential system for MINEs is a variation of one proposed by Horty [10], from which it differs only slightly in its definition of "inheritable path" (Definition 22, Case 1 - to be found later on in this section).

Definition 17 (MINE) *Let $\mathcal{C}$ be a finite set of* nodes*, and let $\mathcal{A}$ be a finite set of* positive edges $x \rightarrow y$ *and* negative edges $x \not\rightarrow y$*, where $x \in \mathcal{C}$ and $y \in \mathcal{C}$. Let "$\rightsquigarrow$" be the union of the "$\rightarrow$" and "$\not\rightarrow$" relations, i.e. $x \rightsquigarrow y$ iff either $x \rightarrow y$ or $x \not\rightarrow y$. A* Multiple Inheritance Network with Exceptions *(MINE) is a pair $\Upsilon = \langle \mathcal{C}, \mathcal{A} \rangle$ such that "$\rightsquigarrow$" is a strict partial order.* ∎

By Υ_C and Υ_A we will refer to the components $\mathcal{C}$ and $\mathcal{A}$ of Υ, respectively.

Given a MINE Υ, we want to characterize the set of conclusions that may be derived from Υ. However, in order to do so we have to introduce the notion of "path".

Definition 18 (Path) *A* chain *in a MINE Υ is a sequence $\pi = x_1 \rightsquigarrow \ldots \rightsquigarrow x_n$ of edges in Υ_A, for $n \geq 2$. A* positive path *in a MINE Υ is a chain $\pi = x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{n-1} \rightarrow x_n$ ($n \geq 2$) in Υ consisting of positive edges only. A* negative path *in a MINE Υ is a chain $\pi = x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{n-1} \not\rightarrow x_n$ ($n \geq 2$) in Υ consisting of a (possibly empty) sequence of positive edges followed by one negative edge. The node x_1 is called* initial node *of π, and will also be denoted by $B(\pi)$; the node x_n is called* terminal node *of π, and will also be denoted by $E(\pi)$. The* polarity *of an edge is given by its positive or negative sign, and the* polarity *of a path is given by the polarity of its last edge. The* conclusion *derived from the path is the edge formed by the initial node of a path, its terminal node and its polarity.* ∎

We will use metavariables $\pi, \pi_1, \pi_2, \ldots$ ranging on paths and metavariables $\Pi, \Pi_1, \Pi_2, \ldots$ ranging on sets of paths.

Intuitively, paths are those chains from which it is somehow possible to draw conclusions. For example, in the MINE $KB = \langle \{opus, penguin, bird, flies, swims\}, \{opus \rightarrow penguin, penguin \rightarrow bird, bird \rightarrow flies, penguin \not\rightarrow flies, flies \not\rightarrow swims\} \rangle$, the chain

$$opus \rightarrow penguin \rightarrow bird \rightarrow flies \tag{10}$$

is a positive path, from which we can draw the conclusion $opus \rightarrow flies$ ("As there is no information to the contrary, we assume that Opus flies"), while from the negative path

$$opus \rightarrow penguin \not\rightarrow flies \tag{11}$$

we may conclude $opus \not\rightarrow flies$ ("As there is no information to the contrary, we assume that Opus does not fly"). The chain

$$opus \rightarrow penguin \not\rightarrow flies \not\rightarrow swims \tag{12}$$

is not a path; in fact, no reasonable conclusion may be drawn from it with respect to the fact whether opus can swim or not. Similar empirical considerations suggest to define the notion of "conclusion" with respect to paths only, and not with respect to chains in general.

As we have seen, MINEs such as KB have paths from which contradictory conclusions may be derived; unfortunately, this generates situations of ambiguity in the representation of states of affairs all but ambiguous to us. In order to eliminate these situations, what we will do is to define which are, given a MINE Υ, the paths from which we may derive "reliable" conclusions, and that must consequently be given priority over others. For example, in a MINE such as $KB1$ this will allow us to give path (11) priority over path (10), hence to inhibit the undesired conclusion according to which opus flies. In our terminology, paths from which we may derive "reliable" conclusions will be called *inheritable* paths. To be inheritable, a path must satisfy a number of conditions; the first of them is that the path must be *constructible*.

Definition 19 (Constructibility) *A* context *is a pair* $\langle \Upsilon, \Pi \rangle$*, where* Υ *is a MINE and* Π *is a set of paths. A positive path* $\pi_1 = x \rightarrow \pi' \rightarrow u \rightarrow y$ *is* constructible *in a context* $\langle \Upsilon, \Pi \rangle$ *iff*

1. $x \rightarrow \pi' \rightarrow u \in \Pi$*, for some (possibly empty) positive path* π'*;*
2. $u \rightarrow y \in \Upsilon_A$*.*

A negative path $\pi_2 = x \rightarrow \pi' \rightarrow u \not\rightarrow y$ *is* constructible *in a context* $\langle \Upsilon, \Pi \rangle$ *iff*

1. $x \rightarrow \pi' \rightarrow u \in \Pi$*, for some (possibly empty) positive path* π'*;*

2. $u \not\rightarrow y \in \Upsilon_A$. ∎

However, a constructible path will not be classified as inheritable if the conclusion that could be drawn from it conflicts with the conclusion that could be drawn from another constructible path. This is made formal by the following definition.

Definition 20 (Conflicting paths) *For any (possibly empty) positive paths π' and π'', the paths $\pi_1 = x \rightarrow \pi' \rightarrow y$ and $\pi_2 = x \rightarrow \pi'' \not\rightarrow y$ are said to* conflict *with each other. A path π_1* is conflicting *in $\langle \Upsilon, \Pi \rangle$ iff Π contains a path π_2 that conflicts with π_1.* ∎

The second restriction governing inheritability is that a constructible path cannot be classified as inheritable in a context in which it is "preempted". This restriction is supposed to reflect the idea that an ideal reasoner should not classify an argument as persuasive whenever his context of reasoning provides him with a better reason for accepting a conflicting argument.

Definition 21 (Preemption) *A positive path $\pi = x \rightarrow \pi' \rightarrow u \rightarrow y$ is* preempted *in a context $\langle \Upsilon, \Pi \rangle$ iff there is a node $v \in \Upsilon_C$ such that*

1. $x \rightarrow \pi'' \rightarrow v \rightarrow \pi''' \rightarrow u \in \Pi$;
2. $v \not\rightarrow y \in \Upsilon_A$.

A negative path $\pi = x \rightarrow \pi' \rightarrow u \not\rightarrow y$ is preempted *in a context $\langle \Upsilon, \Pi \rangle$ iff there is a node $v \in \Upsilon_C$ such that*

1. $x \rightarrow \pi'' \rightarrow v \rightarrow \pi''' \rightarrow u \in \Pi$;
2. $v \rightarrow y \in \Upsilon_A$. ∎

At this point, we can assemble our three preliminary concepts into a formal definition of inheritability.

Definition 22 (Inheritability) *A path π is* inheritable *in a context $\langle \Upsilon, \Pi \rangle$ (written $\langle \Upsilon, \Pi \rangle \vdash \pi$) iff*

Case 1 : *$\pi \in \Upsilon_A$. Then $\langle \Upsilon, \Pi \rangle \vdash \pi$ iff π conflicts with no path in Π consisting of a single edge in Υ_A.*

Case 2 : *π is a compound path. Then $\langle \Upsilon, \Pi \rangle \vdash \pi$ iff*

1. *π is constructible in $\langle \Upsilon, \Pi \rangle$; and*
2. *π is not conflicting in $\langle \Upsilon, \Pi \rangle$; and*

3. π is not preempted in $\langle \Upsilon, \Pi \rangle$. ∎

Finally,

Definition 23 (Extension) *The set of paths Π is an* extension *of the MINE Υ iff $\Pi = \{\pi \mid \langle \Upsilon, \Pi \rangle \vdash \pi\}$. A* conclusion set *$\mathcal{S}$ of Υ is the set of conclusions of paths of an extension of Υ.* ∎

A set of conclusions that may be drawn from a MINE Υ is a set that represents the knowledge "implicitly" present in the KB.

As mentioned earlier on in this section, our inferential system for MINEs differs from Horty's [10] only for Definition 22, Case 1. In fact, in Horty's case, Case 1 of Definition 22 would read:

Case 1 : $\pi \in \Upsilon_A$. Then $\langle \Upsilon, \Pi \rangle \vdash \pi$

i.e. elements of Υ_A are always inheritable, even if they are conflicting in Υ_A. We have modified his definition in order to obtain extensions without conflicting paths; according to Horty's definitions, instead, extensions can include conflicts at the single edge level. Note that in [20] we have introduced the same condition in the definition of "derivable path" for skeptical MINEs (Definition 4, Case 1 of [20]).

5.2 Translation of MINEs into $\mathcal{TDL}^-$ theories

We are now able to give a translation of MINEs into $\mathcal{TDL}^-$. Let Υ be a MINE, and let $P(\Upsilon)$ be the set of all paths of Υ. For every path π_i in $P(\Upsilon)$, let us introduce two new monadic predicate symbols, M_i^+ and M_i^-, into MP. For every node x in Υ, let us introduce a new monadic predicate symbol M_x and an individual constant i_x.

Informally, what we want to do is to make $M_y[i_x]$ (resp. $\neg M_y[i_x]$) hold in an extension $\mathcal{E}$ of the translation $\Xi(\Upsilon)$ of Υ whenever $x \rightarrow y$ (resp. $x \not\rightarrow y$) holds in the corresponding extension of Υ.

Definition 24 *Let π_i and π_j be two positive paths of $P(\Upsilon)$; π_j is an* alternative *of π_i in $P(\Upsilon)$ (and vice versa) iff*

1. *$B(\pi_i) = B(\pi_j)$ and $E(\pi_i) = E(\pi_j)$;*
2. *there exists an edge in π_j which does not appear in π_i, or vice versa.* ∎

Informally, π_i is an alternative path of π_j whenever π_i and π_j are not the same path, and have the same initial and terminal nodes.

Definition 25 *Let $\pi = x_1 \rightarrow \ldots \rightarrow x_n$ $(n \geq 3)$ be a path in $P(\Upsilon)$. Then $\pi' = x_1 \rightarrow \ldots \rightarrow x_j$ $(2 \leq j \leq n)$ is a* subpath *of π in $P(\Upsilon)$. Let us also define $\pi^{\sqcap}$ as the $\mathcal{TDL}^-$ frame $(M_{x_1} \sqcap \ldots \sqcap M_{x_n})$.* ∎

Definition 26 (Translation) *Let Υ be a MINE. The* translation $\Xi(\Upsilon)$ of Υ into $\mathcal{TDL}^-$ *is defined as $\Xi(\Upsilon) = \langle \Psi, \Delta \rangle$, where*

$$
\begin{aligned}
\Psi \;=\; & \{M_x[i_x] \mid x \rightarrow y \in \Upsilon_A \text{ or } x \not\rightarrow y \in \Upsilon_A\} \cup && (13)\\
& \{Isa(i_x, i_x) \mid x \rightarrow y \in \Upsilon_A \text{ or } x \not\rightarrow y \in \Upsilon_A\} \cup && (14)\\
& \{M_x \lessdot M_j^+ \mid \pi_j = x \rightarrow y \in P(\Upsilon)\} \cup && (15)\\
& \{M_i^+ \lessdot M_j^+ \mid \pi_j = \pi_i \rightarrow x_n \in P(\Upsilon)\} \cup && (16)\\
& \{M_j^- \lessdot M_i^+ \mid \pi_j \text{ is an alternative of } \pi_i \text{ in } P(\Upsilon)\} && (17)
\end{aligned}
$$

$$
\begin{aligned}
\Delta \;=\; & \{M_x \mapsto Isa.M_y \mid x \rightarrow y \in \Upsilon_A\} \cup && (18)\\
& \{M_x \mapsto Isa.\neg M_y \mid x \not\rightarrow y \in \Upsilon_A\} \cup && (19)\\
& \{M_i^+ \mapsto Isa.\pi_i^{\sqcap} \sqcap M_n \mid \pi_j = \pi_i \rightarrow x_n \in P(\Upsilon)\} \cup && (20)\\
& \{M_i^+ \mapsto Isa.\pi_i^{\sqcap} \sqcap \neg M_n \mid \pi_j = \pi_i \not\rightarrow x_n \in P(\Upsilon)\} \cup && (21)\\
& \{M_i^+ \mapsto Isa.\pi_i^{\sqcap} \sqcap M_i^- \mid \pi_i \text{ is an alternative of } \pi_j \text{ in } P(\Upsilon)\} \cup && (22)\\
& \{M_i^- \mapsto Isa.\pi_j^{\sqcap} \sqcap M_n \mid \pi_k = \pi_j \rightarrow x_n \in P(\Upsilon) \text{ and} && (23)\\
& \qquad \pi_j \text{ is a subpath of } \pi_i\} \cup \\
& \{M_i^- \mapsto Isa.\pi_j^{\sqcap} \sqcap \neg M_n \mid \pi_k = \pi_j \not\rightarrow x_n \in P(\Upsilon) \text{ and} && (24)\\
& \qquad \pi_j \text{ is a subpath of } \pi_i\}
\end{aligned}
$$
∎

Instead of explaining every clause of the translation Ξ, we will try to convey the main idea underlying it.

Our aim is to capture in the translation Ξ the correct behaviour of the MINE inferential mechanism once it is confronted with the notorious problematic cases of inheritance reasoning.

For inheritable positive paths, the problematic case is the co-presence of the following types of paths:

- $\pi = x \rightarrow \pi_l \rightarrow u \rightarrow y$;
- $\pi' = x \rightarrow \pi_l' \rightarrow v \not\rightarrow y$ and
- $\pi'' = x \rightarrow \pi_l' \rightarrow v \rightarrow \pi_l'' \rightarrow u$.

Note that π'' is an alternative path of $x \rightarrow \pi_l \rightarrow u$, whereas π'' and $x \rightarrow \pi_l' \rightarrow v$ are subpaths of π. Then path π would be inheritable in a context iff

1. $x \rightarrow \pi_l \rightarrow u$ is inheritable;

2. there is no inheritable alternative path $\pi'' = x \rightarrow \pi'_l \rightarrow v \rightarrow \pi''_i \rightarrow u$ with $v \not\rightarrow y \in \Upsilon_A$.

Now observe that

1. Condition 1 is captured by Clause 16 and Clause 20 (applied to path π);

2. Condition 2 is captured by Clause 17 (applied to path π''), by Clause 22 (applied to path π'') and by Clause 24 (applied to path π' with $x \rightarrow \pi'_l \rightarrow v$ as subpath).

As for inheritable negative paths, the analogous of Condition 1 is captured by Clauses 16 and 21, whereas the analogous of Condition 2 is captured by Clauses 17, 22 and 23.

Finally, Clauses 13, 14, 15 and 18 (for positive edges) and Clause 19 (for negative edges) handle the inheritability of single edges (positive and negative).

We then have the translation theorem

Theorem 3 *Let Υ be a MINE. Then $\mathcal{S}$ is a conclusion set of Υ iff $\Xi(\Upsilon)$ has an extension $\mathcal{E}$ such that:*

1. *$x \rightarrow y \in \mathcal{S}$ iff $M_y[i_x] \in \mathcal{E}$ and $M_y \neq M_x$;*

2. *$x \not\rightarrow y \in \mathcal{S}$ iff $\neg M_y[i_x] \in \mathcal{E}$ and $M_y \neq M_x$.*

Proof (Sketch). *We will sketch the proof only for the "if" part of Clause 1 of the Theorem; the other cases are similar. Let $\mathcal{S}$ be a conclusion set of Υ, obtained from the extension Π of Υ. Let $\mathcal{E}$ be the set $TC(\Psi \cup \Lambda)$ where $\alpha \in \Lambda$ iff*

1. *$\alpha = M_y[i_x]$, $x \rightarrow y \in \mathcal{S}$ and $M_x \neq M_y$;*

2. *$\alpha = \neg M_y[i_x]$, $x \not\rightarrow y \in \mathcal{S}$ and $M_x \neq M_y$;*

3. *$\alpha = M_i^-[i_{B(\pi)}]$ and $\pi \in \Pi$.*

We now show that $\mathcal{E} = \Theta(\mathcal{E}, \mathcal{T})$; thus $x \rightarrow y \in \mathcal{S}$ implies that $M_y[i_x] \in \mathcal{E}$ and that $M_x \neq M_y$. First at all, it is easy to see that $\mathcal{E}$ satisfies Conditions 1 and 2 of Definition 11. Let us then show that $\mathcal{E} \subseteq \Theta(\mathcal{E}, \mathcal{T})$. For $M_y[i_x] \in \mathcal{E}$ we have $x \rightarrow y \in \mathcal{S}$. Thus, there is a path $\pi_i = x \rightarrow \sigma \rightarrow y \in \Pi$. It can be shown that Condition 3 of Definition 11 is satisfied for the default $M_i^+ \mapsto Isa.\pi_i^{\sqcap} \sqcap M_y$ and $M_i^+[i_x] \in \mathcal{E}$. Thus, $M_y[i_x] \in \Theta(\mathcal{E}, \mathcal{T})$. Finally, let us show that $\Theta(\mathcal{E}, \mathcal{T}) \subseteq \mathcal{E}$ Suppose instead that $\Theta(\mathcal{E}, \mathcal{T}) \not\subseteq \mathcal{E}$. Then it can be shown that there is $M_y[i_x] \notin \mathcal{E}$ such that $M_y[i_x] \in \Theta(\mathcal{E}, \mathcal{T})$ and $x \rightarrow y \in \mathcal{S}$. Therefore, $M_y[i_x] \in \mathcal{E}$; a contradiction. ∎

The sense of this theorem will be best illustrated by means of an example[7].

Example 4 *Consider the MINE* $\Upsilon = \langle \{p, r, s, q\}, \{p \rightarrow r,\ p \rightarrow q,\ r \rightarrow q,\ q \rightarrow s,\ r \not\rightarrow s\rangle$. *The translation* Ξ *of* Υ *into* $\mathcal{TDL}^{-}$ *is* $\Xi(\Upsilon) = \langle \Psi, \Delta \rangle$, *where:*

- *the T-set* Ψ *is composed of the union of the following sets:*
  - $\{M_p[i_p], M_r[i_r], M_q[i_q]\}$, *from line (13) of Definition 26;*
  - $\{Isa[i_p, i_p], Isa[i_r, i_r], Isa[i_q, i_q]\}$, *from line (14) of Definition 26;*
  - $\{M_p{<\!\cdot}M^+_{pr}, M_p{<\!\cdot}M^+_{pq}, M_q{<\!\cdot}M^+_{qs}, M_r{<\!\cdot}M^+_{rq}\}$, *from line (15) of Definition 26;*
  - $\{M^+_{pq}{<\!\cdot}M^+_{pqs}, M^+_{pr}{<\!\cdot}M^+_{prq}, M^+_{prq}{<\!\cdot}M^+_{prqs}, M^+_{rq}{<\!\cdot}M^+_{rqs}\}$, *from line (16) of Definition 26;*
  - $\{M^-_{prq}{<\!\cdot}M^+_{pq}, M^-_{pq}{<\!\cdot}M^+_{prq}\}$, *from line (17) of Definition 26;*
- *the set of defaults* Δ *is composed of the union of the following sets:*
  - $\{M_p \mapsto Isa.M_r, M_p \mapsto Isa.M_q, M_r \mapsto Isa.M_q, M_q \mapsto Isa.M_s\}$, *from line (18) of Definition 26;*
  - $\{M_r \mapsto Isa.\neg M_s\}$, *from line (19) of Definition 26;*
  - $\{M^+_{pr} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q), M^+_{prq} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q \sqcap M_s), M^+_{pq} \mapsto Isa.(M_p \sqcap M_q \sqcap M_s), M^+_{rq} \mapsto Isa.(M_r \sqcap M_q \sqcap M_s)\}$, *from line (20) of Definition 26;*
  - $\{M^+_{pr} \mapsto Isa.(M_p \sqcap M_r \sqcap \neg M_s)\}$, *from line (21) of Definition 26;*
  - $\{M^+_{prq} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q \sqcap M^-_{prq}), M^+_{pq} \mapsto Isa.(M_p \sqcap M_q \sqcap M^-_{pq}), M^+_{prqs} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q \sqcap M_s \sqcap M^-_{prqs}), M^+_{pqs} \mapsto Isa.(M_p \sqcap M_q \sqcap M_s \sqcap M^-_{pqs})\}$, *from line (22) of Definition 26;*
  - $\{M^-_{prqs} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q), M^-_{prqs} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q \sqcap M_s), M^-_{prq} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q), M^-_{prq} \mapsto Isa.(M_p \sqcap M_r \sqcap M_q \sqcap M_s), M^-_{pqs} \mapsto Isa.(M_p \sqcap M_q \sqcap M_s), M^-_{rq} \mapsto Isa.(M_r \sqcap M_q \sqcap M_s)\}$, *from line (23) of Definition 26;*
  - $\{M^-_{prqs} \mapsto Isa.(M_p \sqcap M_r \sqcap \neg M_s), M^-_{prq} \mapsto Isa.(M_p \sqcap M_r \sqcap \neg M_s),\}$, *from line (24) of Definition 26.*

It can be easily shown that the set of paths $\Pi = \Upsilon \cup \{p \rightarrow r \rightarrow q, p \rightarrow r \not\rightarrow s\}$ *is the unique extension of* Υ, *and therefore* $\mathcal{S} = \Upsilon \cup \{p \not\rightarrow s\}$ *is the unique conclusion set of* Υ. *Let* $\mathcal{E}$ *be the (consequentially unique) extension of* $\Xi(\Upsilon)$. *It is easy to verify that* $M_s[i_p] \notin \mathcal{E}$ *and that* $\neg M_s[i_p] \in \mathcal{E}$, *consistently with Theorem 3.* ∎

[7] In this example, for brevity of notation we will indicate a path by means of the sequence of letters of the nodes belonging to it. So, the positive path $p \rightarrow q \rightarrow s$ will be indicated by pqs, while the negative path $p \rightarrow q \not\rightarrow s$ will be indicated by $\overline{pqs}$.

6 Conclusion

In this paper we have shown how we can extend terminological logics in such a way that they allow a brand of default reasoning that obeys the specialization principle, thus creating a formalism (which we have dubbed $\mathcal{TDL}^-$) that combines the tools for describing taxonomic organizations of complex objects which are typical of TLs, the ability to describe default information which is typical of general nonmonotonic formalisms, and the incrementality in KB construction which is typical of MINEs.

This has been obtained by relying on the notion of "extension of a $\mathcal{TDL}^-$ theory", a notion that has been defined in the style pioneered by Reiter in his Default Logic, i.e. as a fixpoint of a consequence relation. We have also studied a number of properties of $\mathcal{TDL}^-$ related to issues such as the existence and the uniqueness of extensions, the relationships of $\mathcal{TDL}^-$ with Default Logic and MINEs, and the complexity of $\mathcal{TDL}^-$ reasoning.

The language of $\mathcal{TDL}^-$ has been designed with the aim of providing a minimal framework allowing to study the interaction of terminological and default information in a meaningful way. Quite obviously, extensions to this framework may be conceived that enable the expression of default information of a nature different from the one considered here.

References

[1] Fahiem Bacchus. A modest but semantically well founded inheritance reasoner. In *Proceedings of IJCAI-89, 11th International Joint Conference on Artificial Intelligence*, pages 1104–1105, Detroit, MI, 1989.

[2] Craig Boutilier. On the semantics of stable inheritance reasoning. Technical Report KRR-TR-89-11, Department of Computer Science, University of Toronto, Toronto, Ontario, 1989.

[3] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in knowledge representation.* Morgan Kaufmann, Los Altos, CA, 1985.

[4] Gerhard Brewka. The logic of inheritance in frame systems. In *Proceedings of IJCAI-87, 10th International Joint Conference on Artificial Intelligence*, pages 483–488, Milano, Italy, 1987.

[5] James P. Delgrande. A semantics for a class of inheritance networks. In *Proceedings of CSCSI/SCEIO-90, 8th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 54–60, Ottawa, Ontario, 1990.

[6] Jon Doyle and Drew McDermott. Nonmonotonic logic I. *Artificial Intelligence*, 13:41–72, 1980. [a] Appears also in [7], pp. 111–126.

[7] Matthew L. Ginsberg, editor. *Readings in nonmonotonic reasoning.* Morgan Kaufmann, Los Altos, CA, 1987.

[8] John Haugeland, editor. *Mind design.* The MIT Press, Cambridge, MA, 1981.

[9] Bernhard Hollunder. Hybrid inferences in KL-ONE-based knowledge representation systems. In *Proceedings of the 14th German Workshop on Artificial Intelligence*, pages 38–47, Eringerfeld, FRG, 1990.

[10] John F. Horty. A credulous theory of mixed inheritance. In Maurizio Lenzerini, Daniele Nardi, and Maria Simi, editors, *Inheritance hierarchies in knowledge representation and programming languages*, pages 13–28. Wiley, Chichester, GB, 1991.

[11] John F. Horty, Richmond H. Thomason, and David S. Touretzky. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence*, 42:311–348, 1990.

[12] Thirunarayan Krishnaprasad, Michael Kifer, and D.S. Warren. On the declarative semantics of inheritance networks. In *Proceedings of IJCAI-89, 11th International Joint Conference on Artificial Intelligence*, pages 1099–1103, Detroit, MI, 1989.

[13] Maurizio Lenzerini, Daniele Nardi, and Maria Simi, editors. *Inheritance hierarchies in knowledge representation and programming languages.* Wiley, Chichester, GB, 1991.

[14] Vladimir Lifschitz, editor. *Programs with common sense - Papers by John McCarthy.* Ablex, Norwood, NJ, 1990.

[15] John McCarthy. Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980. [a] Appears also in [7], pp. 145–151. [b] Appears also in [14], pp. 142–155.

[16] Marvin Minsky. A framework for representing knowledge. In Patrick J. Winston, editor, *The psychology of computer vision*, pages 211–277. McGraw-Hill, New York, NY, 1975. [a] An extended version appears also in [3], pp. 245–262, and in [8], pp. 95–128.

[17] Robert A. Nado and Richard E. Fikes. Semantically sound inheritance for a formally defined frame language with defaults. In *Proceedings of AAAI-87, 6th Conference of the American Association for Artificial Intelligence*, pages 443–448, Seattle, WA, 1987.

[18] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980. [a] Appears also in [7], pp. 68–93.

[19] R. Bruce Roberts and Ira P. Goldstein. The FRL manual. Technical Report 409, The Artificial Intelligence Laboratory, MIT, Cambridge, MA, 1977.

[20] Fabrizio Sebastiani and Umberto Straccia. Incremental acquisition of knowledge for non-monotonic reasoning. *Computers and Artificial Intelligence*, 13(4):377–396, 1994.

[21] Bart Selman and Hector J. Levesque. The tractability of path-based inheritance. In *Proceedings of IJCAI-89, 11th International Joint Conference on Artificial Intelligence*, pages 1140–1145, Detroit, MI, 1989. [a] Appears also in [13], pp. 83–96.

[22] David S. Touretzky, John F. Horty, and Richmond H. Thomason. A clash of intuitions: the current state of nonmonotonic multiple inheritance systems. In *Proceedings of IJCAI-87, 10th International Joint Conference on Artificial Intelligence*, pages 476–482, Milano, Italy, 1987.