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). Let us consider the well-known "penguin" example. It may be represented by means of the DL rule

\[
\frac{\text{Bird}(x): \text{Flies}(x)}{(x)}
\]

("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:

\[
\forall x \text{ Penguin}(x) \Rightarrow \text{Bird}(x)
\]

\[
\frac{\text{Penguin}(x): \neg \text{Flies}(x)}{\neg \text{Flies}(x)}
\]

There is a problem hidden in this set of expressions: in DL, the addition of the assertion $\text{Penguin}(\text{opus})$ to the set (1) \(\vdash\) (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:

\[
\frac{\text{Bird}(x): \text{Flies}(x) \land \neg \text{Penguin}(x)}{\text{Flies}(x)}
\]

\(^1\)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".
\[ \forall x \text{Penguin}(x) \Rightarrow \text{Bird}(x) \]

\[ \text{Penguin}(x) : \neg \text{Flies}(x) \]

\[ \neg \text{Flies}(x) \]

At this point, given \( \text{Penguin}(\text{opus}) \), rule (4) does not allow one to conclude \( \text{Flies}(\text{opus}) \); 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 \textit{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 \textit{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 \textit{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 \( \text{Bird}(x) \) is \textit{derivable} from \( \text{Penguin}(x) \) 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 \( \text{Flies}(x) \) from \( \text{Penguin}(x) \). In general, given a DL knowledge base, it is not even determined \textit{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 \( \text{Penguin}(x) \) and \( \text{Flies}(x) \)); in case several such chains exist, \textit{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 incremen-
tal) 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 Net-
works with Exceptions (MINEs), a popular, albeit less general, class of non-
monotonic KR languages oriented to the representation of taxonomic knowl-
edge (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 tax-
onomy: 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 typi-
cally 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 nonmono-
tonlic 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 knowl-
edge allowing the description of “prototypical” knowledge and resulting in
KBS of taxonomic form. We will call our logic \( \text{TDL}^- \) (Terminological De-
fault 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 $\text{Triangle}$, $\text{Polyhedron}$ and $\text{Regular polyhedron}$, denoting the sets of triangles, polyhedra and regular polyhedra, respectively, and the binary predicate symbol $\text{Side}$, denoting all those pairs of individuals $(x, y)$ 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 $(\forall \text{Side.Triangle})$, 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 \((-\text{Triangle})\), denoting the set of those individuals that are not triangles;

• the frame \((\text{Regular polyhedron} \cap (\forall \text{Side} \cdot \text{Triangle}))\), 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 \(i, i_1, i_2, \ldots\)), an alphabet \(MP\) of monadic predicate symbols (with metavariables \(M, M_1, M_2, \ldots\)) and an alphabet \(DP\) of dyadic predicate symbols (with metavariables \(D, D_1, D_2, \ldots\)). 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:

\[
F_1, F_2 \rightarrow (F_1 \cap F_2) \mid M \mid (\neg M) \mid (\forall S. F) \mid \bot \mid \top
\]

\[
S \rightarrow D
\]

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 \(I\) 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 \(I(i_1) \neq I(i_2)\) whenever \(i_1 \neq i_2\). We will say that \(I\) is an extension function iff \(I(\bot) = \emptyset\), \(I(\top) = \mathcal{D}\), \(I(\neg M) = \mathcal{D} \setminus I(M)\), \(I(F_1 \cap F_2) = I(F_1) \cap I(F_2)\), \(I(\forall S. F) = \{x \in \mathcal{D} \mid \forall y : (x, y) \in I(S) \Rightarrow y \in I(F)\}\).

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 \(I\) over a nonempty domain \(\mathcal{D}\) satisfies an assertion \(F[i]\) iff \(I(i) \in I(F)\), and satisfies an assertion \(S[i_1, i_2]\) iff \(\langle I(i_1), I(i_2)\rangle \in I(S)\).

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.
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 $M = F$ or of the form $M < F$, 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 $M = F$ iff $\mathcal{I}(M) = \mathcal{I}(F)$, and satisfies a naming $M < F$ iff $\mathcal{I}(M) \subseteq \mathcal{I}(F)$.

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

- `Irregular_polyhedron = (Polyhedron \cap (\neg \text{Regular_polyhedron}))`
- `Regular_tetrahedron = (\text{Regular_polyhedron} \cap (\forall \text{Side.Triangle}))`
- `Regular_polygon < Polygon`

Namings of type $M = F$ define necessary and sufficient conditions for an individual to be in the denotation of $M$, while the conditions defined by namings of type $M < F$ 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 $\Omega, \Omega_1, \Omega_2, ...$ and $\Psi, \Psi_1, \Psi_2, ...$ ranging on T-sets. The notion of **satisfiability** of a T-set $\Omega$, and that of a **model** of a T-set $\Omega$, 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 $\Omega$ iff $\mathcal{I}$ satisfies all the assertions in $A$ and all the namings in $N$. A T-set $\Omega$ is satisfiable iff it has a model (otherwise, we say $\Omega$ is unsatisfiable).

---

4 For brevity of notation, in the rest of the paper we will sometimes consider a T-set $\Omega$ 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 $\Omega$, we say that $M_1$ subsumes $M_2$ in $\Omega$ when $\Omega$ allows us to draw the conclusion that $M_1$ is "more general" than $M_2$. We say that a T-set $\Omega$ supports an assertion $\alpha$ when $\Omega$ allows us to draw the conclusion that $\alpha$.

**Definition 7** Let $\Omega$ 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 $\Omega$ (written $M_1 \preceq_\Omega M_2$) iff for every model $I$ of $\Omega$ it is true that $I(M_1) \subseteq I(M_2)$. We say that $\Omega$ supports an assertion $\alpha$ (written $\Omega \models \alpha$) iff $\alpha$ is satisfied by all models of $\Omega$.

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 $\Omega$.

**Definition 8** Let $\Omega = \langle A, N \rangle$ be a satisfiable T-set. The transitive closure of $\Omega$, 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 $M \rightarrow S.F$, 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, $M \rightarrow S.F$ 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 $IU \rightarrow FM.I$ 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;

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 $\Psi$ is a satisfiable finite $T$-set, and $\Delta$ 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, 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 $F_1 \rightarrow S.F_2$; 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 $M \rightarrow S.F_2$.

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 $\mathcal{T} = (\Psi, \Delta)$ be a $\text{TDL}^-$ theory. Let $\Theta$ be an operator such that, for any satisfiable $\mathcal{T}$-set $\Omega$, $\Theta(\Omega, \mathcal{T})$ is the smallest satisfiable $\mathcal{T}$-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 $M_i \mapsto S.F_i \in \Delta$, for all the assertions $M_i[i_1] \in \Theta(\Omega, \mathcal{T})$ such that $S[i_1, i_2] \in \Theta(\Omega, \mathcal{T})$, it happens that $F_i[i_2] \in \Theta(\Omega, \mathcal{T})$ 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) $\Omega \cup \{(F_1 \cap F_2)[i_2]\}$ is unsatisfiable.

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

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 $M_i \mapsto S.F_i$ is “applicable” and $(3a) \div (3c)$ there is no evidence contradicting the conclusion of the default (i.e. $i_1$ does not belong to any subclass of $M_i$ which is the premise of a default whose conclusion would be inconsistent with $F_i$), 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 $\text{TDL}^-$ employs an implicit handling of exceptions.

**Example 1** Let $\mathcal{T} = (\Psi, \Delta)$ be the $\text{TDL}^-$ theory that formalizes our “Professors” example, with $\Psi = \{\text{IU}[b], \text{FM}[b,d], \text{STU}[p], \text{FM}[p,k]\}, \{\text{STU}<\text{IU}\}$ and $\Delta = \{\text{IU} \mapsto \text{FM}.i, \text{STU} \mapsto \text{FM}(-i)\}$. Let $\mathcal{E} = TC(\Psi \cup \{\text{(-I)}[k], \text{I}[d]\})$. 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.
\[ IU(x) \land FM(x, y) : I(y) \land \neg STU(x) \]
\[ (5) \]
\[ STU(x) \land FM(x, y) : \neg I(y) \]
\[ (6) \]
\[ \forall x \ STU(x) \Rightarrow IU(x) \]
\[ (7) \]
\[ IU(b) \land FM(b, d) \land STU(p) \land FM(p, k) \]
\[ (8) \]

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 \( T = \langle \Psi, \Delta \rangle \), an extension \( \mathcal{E} \) of \( T \), a \( T \)-set \( \Psi' \) and a set of defaults \( \Delta' \), such that for no extension \( \mathcal{E}' \) of \( T' = \langle \Psi \cup \Psi', \Delta \cup \Delta' \rangle \) it is true that \( \mathcal{E} \subseteq \mathcal{E}' \).

**Proof.** Let \( T \) be the \( \mathcal{TDL}^- \) theory of Example 1 and \( \mathcal{E} \) its (unique) extension. Let \( \Psi' = \{STU[b]\} \) and \( \Delta' = \emptyset \). It is easy to see that the only extension of \( T' = \langle \Psi \cup \Psi', \Delta \cup \Delta' \rangle \) is \( \mathcal{E}' = TC(\Psi \cup \Psi' \cup \{(-I)[k], (-I)[d]\}) \). 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 \( T = \langle \Psi, \Delta \rangle \) be a \( \mathcal{TDL}^- \) theory, and let \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) be extensions of \( T \). 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 \( T = \langle \Psi, \Delta \rangle \) be a \( \mathcal{TDL}^- \) theory. Let the set of conflicts \( C_T \) of \( T \) be the set
\[
C_T = \bigcup_{S} \{ \langle M_1, M_2 \rangle \mid \begin{array}{l}
M_1 \mapsto S.F_1 \in \Delta, M_2 \mapsto S.F_2 \in \Delta, \\
(F_1 \cap F_2) \text{ is unsatisfiable,} \\
\text{for some } i_1, i_2, S[i_1, i_2] \in \Psi, \\
M_1 \preceq_{\Psi} M_2
\end{array} \}
\]

We will say that \( T \) is conflict-free iff \( C_T = \emptyset \).
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 $\Psi_{DL}$ is the obvious translation of namings and assertions into first order logic and $\Delta_{DL}$ is obtained by translating defaults $M \models S.F \in \Delta$ into normal defaults

$$\frac{M(x) \land S(x,y) : F'(x)}{F'(x)}$$

(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 $\text{Tcl}(\mathcal{E}_{DL})$ is an extension of $\mathcal{T}_{DL}$, where $\text{Tcl}$ 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 $\langle \{ M_1[i], M_2[i], S_1[i, i], S_2[i, i], M_2 \models M_1 \}, \{ M_1 \models S_1.(\forall S_2(\neg M_3)), M_2 \models S_1, M_3 \} \rangle$. The default $M_2 \models S_1, M_3$ is exceptional to the default $M_1 \models S_1.(\forall S_2(\neg M_3))$ 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).
\[\Psi = \{\{M_k[i_j] \mid 1 \leq j \leq n, 1 \leq k \leq 2\} \cup \{S_k[i_j, i_j] \mid 1 \leq j, k \leq n\}, \emptyset\}\]

\[\Delta = \{M_1 \mapsto S_j.M_j \mid 1 \leq j \leq n\} \cup \{M_2 \mapsto S_j.(\neg M_j) \mid 1 \leq j \leq n\}\]

\(T\) 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), \(T\) has \(2^{n^2}\) extensions. Therefore, considering that \(|T| = |\Psi| + |\Delta|\) is \(O(n)\), the number of extensions of \(T\) is exponential with respect to the size of \(T\).

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 \(TDL^-\) 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:

\[
A_j(a_i) \quad \text{for } 1 \leq i \leq n \text{ and } 1 \leq j \leq 2
\]

\[
S_j(a_i, a_i) \quad \text{for } 1 \leq i, j \leq n
\]

\[
\frac{A_i(x) \cap P_i(x)}{P_i(x)} \quad \text{for } 1 \leq i \leq n
\]

\[
\frac{A_2(x) \cap \neg P_i(x)}{\neg P_i(x)} \quad \text{for } 1 \leq i \leq n
\]

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

**Proposition 6** There exists a \(TDL^-\) theory with no extensions.

This is proven by showing that the sample theory \(T = (\Psi, \Delta)\), with \(\Psi = \{\{M_1[i], S[i, i]\}, \{M_1 \equiv (F_1 \cap F_2), M_2 \equiv (F_1 \cap (F_2 \cap F_3))\}\}\) 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 \(TDL^-\) such that the existence of at least one extension is always guaranteed: in fact, removing from \(TDL^-\) the causes that are responsible for the non-existence of an extension for theory \(T\) would dramatically curtail the expressive power of the language itself.

### 4 Computing an extension of a \(TDL^-\) theory

In this section we will discuss the properties of the EXT (nondeterministic) algorithm that computes (when it exists) an extension of a \(TDL^-\) 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 $|=\ $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 $|=\ $, and that the decision of $|=\ $can in turn be reduced to deciding if $\Omega$ 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 definitum is defined, either directly or indirectly, in terms of itself) is unsatisfiable. By considering acyclic $\mathcal{TDL}^-$

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

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

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

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

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

We will write $A_T(\Omega)$ as short for the set of applicable defaults of $T$ in $\Omega$.

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

Proposition 7 Let $T = \langle \Psi, \Delta \rangle$ be an acyclic $\mathcal{TDL}^-$ theory and $\Omega$ a finite and acyclic set of namings and assertions. Then the set of instantiated defaults of $T$ and $A_T(\Omega)$ are recursive sets.
Let $T = \langle \Psi, \Delta \rangle$ 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
    $A_T^i \leftarrow (A_T(\omega_i) \cap A_T(\Omega_{n-1})) \setminus \Gamma_i$;
    if not empty($A_T^i$)
      then choose $\gamma$ from $A_T^i$;
        $\Gamma_{i+1} \leftarrow \Gamma_i \cup \{ \gamma \}$;
        $\omega_{i+1} \leftarrow \omega_i \cup \{ \text{Consequent(}\gamma\text{)} \}$;
    endif
    $i \leftarrow i + 1$;
until empty($A_T^{i-1}$);
$\Omega_n \leftarrow \omega_{i-1}$;
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 $\Omega_n$ is built from the first component $\Psi$ of an acyclic $\mathcal{TDL}^-$ theory $T$ 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 $T = \langle \Psi, \Delta \rangle$ be a $\mathcal{TDL}^-$ theory, with

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

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 $T$. The application of the EXT algorithm to $T$ would yield the following sequence of steps:

1. $\Omega_0 = \omega_0 = \Psi, \Gamma_0 = \emptyset, A_T(\omega_0) = A_T(\Omega_0) = A_T^0 = \{ \gamma_1 \}$;
2. $\omega_1 = \omega_0 \cup \{\text{Consequent}(\gamma_1)\}, \Gamma_1 = \{\gamma_1\}$;
3. $A_T(\omega_1) = \{\gamma_1, \gamma_2\}, A_T^1 = \emptyset$;
4. $\Omega_1 = \omega_1$;
5. $\omega_0 = \Psi, \Gamma_0 = \emptyset, A_T(\omega_0) = A_T^0 = \{\gamma_1\}, A_T(\Omega_1) = \{\gamma_1, \gamma_2\}$;
6. $\omega_1 = \omega_0 \cup \{\text{Consequent}(\gamma_1)\}, \Gamma_1 = \{\gamma_1\}$;
7. $A_T(\omega_1) = \{\gamma_1, \gamma_2\}, A_T^1 = \{\gamma_2\}$;
8. $\omega_2 = \omega_1 \cup \{\text{Consequent}(\gamma_2)\}, \Gamma_1 = \{\gamma_1, \gamma_2\}$;
9. $A_T(\omega_2) = \{\gamma_1, \gamma_2\}, A_T^2 = \emptyset$;
10. $\Omega_2 = \omega_2$;

\[\vdots\] 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 $T$ can be computed by the algorithm. Before proceeding to its proof, we need to state

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

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

**Proof.**

("if") Consider the last current state $\omega_n$. It follows that

1. $\omega_n = \Omega_n = \Omega_{n-1}$ and $TC(\Omega_n) = E$;
2. $A_T^\omega = \emptyset = (A_T(\omega_n) \cap A_T(\Omega_{n-1})) \setminus \Gamma_n$;
3. $E = TC(\Psi \cup \text{Consequent}(A_T(E)))$.  

17
From $A^\omega_T = \emptyset$ it follows that $A_T(\mathcal{E}) \subseteq \Gamma_n$. 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 $\mathcal{TC}$ 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 $\gamma \in A_T(\mathcal{E})$ such that $\text{Consequent}(\gamma) \in \mathcal{E}$ and $\text{Consequent}(\gamma) \not\in \Theta(\mathcal{E}, \mathcal{T})$. From Definition 11 it follows that if $\gamma \in A_T(\mathcal{E})$, then $\text{Consequent}(\gamma) \subseteq \Theta(\mathcal{E}, \mathcal{T})$; a contradiction. Let us now prove that $\Theta(\mathcal{E}, \mathcal{T})) \not\subseteq \mathcal{E}$. Suppose instead that $\Theta(\mathcal{E}, \mathcal{T})) \not\subseteq \mathcal{E}$. Therefore, there exists $\gamma \in A_T(\mathcal{E})$ such that $\text{Consequent}(\gamma) \in \Theta(\mathcal{E}, \mathcal{T})$, but $\text{Consequent}(\gamma) \not\in \mathcal{E}$; a contradiction, since point (3.) holds.

(“only if”) Let $\mathcal{E}$ be an extension of $\mathcal{T}$. By Lemma 1, it is true that $\mathcal{E} = \mathcal{TC}(\Psi \cup \text{Consequent}(A_T(\mathcal{E})))$. 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 $1 \leq i \leq n-1$, $\Omega_{i-1} \subseteq \Omega_i \subseteq \Psi \cup \text{Consequent}(A_T(\mathcal{E}))$, and if $\text{Consequent}(\gamma) \in A_T(\Omega_i)$, then $\gamma \in A_T(\Omega_{i-1})$. Now it easy to see that we can choose adequately $\gamma \in A_T(\mathcal{E})$ such that the sequence $\Omega_0, \ldots, \Omega_{n-1}$, 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 $\Omega_i$ 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 $\Psi$ iff $F$ is a conjunction of unary predicate symbols or negated unary predicate symbols which do not appear as names in $\Psi$. We will say that $\mathcal{T}$ is a restricted $\mathcal{TDL}^-$ theory iff

1. $\Psi$ contains only assertions of the form $F[i]$, where $F$ is a simple frame wrt $\Psi$.
2. $\Psi$ contains only namings of the form $M \equiv F$, where $F$ is a unary predicate symbol (or the negation of a unary predicate symbol) which is not a name in $\Psi$, or axioms of the form $M \prec F$, where $M$ is not a name in $\Psi$;

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

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 $\sigma$ in 3CNF consider the restricted $\mathcal{TDL}^-$ theory $T_\sigma$, obtained exactly from the expressions which appear in the following rules.

1. for every symbol $P$ which occurs in $\sigma$, the following expressions appear in $T_\sigma$, 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 $X \lor Y \lor Z$ of $\sigma$, the following expressions appear in $T_\sigma$, where $A$, $B$, $F$ and $F_{xyz}$ are new unary predicate symbols: $A \equiv (\neg X)$ (also, we will substitute $P$ for $(\neg(\neg P)))$, $A \mapsto (F_{xyz} \cap (\neg Y \cap (\neg Z)))$, $F_{xyz} \mapsto (F \cap (\neg B))$, $F \prec F_{xyz}$ and $F \mapsto B$.

We will show that $\sigma$ is satisfiable iff $T_\sigma$ 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 $T_\sigma$ has an extension $E$. By rule (1.) it follows that every unary predicate symbol in $\sigma$ or its negation appears in $E$. Furthermore, it must not be the case that $F[i] \in E$ because otherwise rules of type (2.) would have the consequence that $B[i] \in E$ 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 $\sigma$, can all be applicable in $E$. Therefore, every clause of $\sigma$ is satisfied by a model which "agrees" with $E$.

("only if") Suppose $M$ is a model of $\sigma$. Let $\Omega$ be the set of assertions $X[i]$ such that $M$ is a model of the literal $X$, together with $A[i]$. Then $E = TC(\Omega)$ is an extension of $T_\sigma$. Note that $E$ is dependent on expressions introduced by rule (1.) and that none of expressions introduced by rule (2.) apply.
5 \textit{TDL}^- and its relationships with MINEs

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

5.1 Our MINE Language

Our language and inferential system for MINEs is a variation of one proposed by Horty \cite{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 \( C \) be a finite set of nodes, and let \( A \) be a finite set of positive edges \( x \rightarrow y \) and negative edges \( x \not\rightarrow y \), where \( x, y \in C \). Let \( \sim \) be the union of the \( \rightarrow \) and \( \not\rightarrow \) relations, i.e. \( x \leadsto y \) iff either \( x \rightarrow y \) or \( x \not\rightarrow y \). A Multiple Inheritance Network with Exceptions (MINE) is a pair \( (C, A) \) such that \( \sim \) is a strict partial order.

By \( \Upsilon_C \) and \( \Upsilon_A \) we will refer to the components \( C \) and \( A \) of \( \Upsilon \), respectively.

Given a MINE \( \Upsilon \), we want to characterize the set of conclusions that may be derived from \( \Upsilon \). However, in order to do so we have to introduce the notion of “path”.

Definition 18 (Path) A chain in a MINE \( \Upsilon \) is a sequence \( \pi = x_1 \leadsto \ldots \leadsto x_n \) of edges in \( \Upsilon_A \), for \( n \geq 2 \). A positive path in a MINE \( \Upsilon \) is a chain \( \pi = x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{n-1} \rightarrow x_n \) \((n \geq 2)\) in \( \Upsilon \) consisting of positive edges only. A negative path in a MINE \( \Upsilon \) is a chain \( \pi = x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{n-1} \not\rightarrow x_n \) \((n \geq 2)\) in \( \Upsilon \) consisting of a (possibly empty) sequence of positive edges followed by one negative edge. The node \( x_1 \) is called initial node of \( \pi \), and will also be denoted by \( B(\pi) \); the node \( x_n \) is called terminal node of \( \pi \), 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 = \{\{opus, penguin, bird, flies, swims\}, \{opus \rightarrow penguin, penguin \rightarrow bird, bird \rightarrow flies, penguin \not\rightarrow flies, flies \not\rightarrow swims\}\}, the chain
opus $\rightarrow$ penguin $\rightarrow$ bird $\rightarrow$ flies \hspace{1cm} (10)

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

\[ \text{opus} \rightarrow \text{penguin} \not\rightarrow \text{flies} \hspace{1cm} (11) \]

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

\[ \text{opus} \rightarrow \text{penguin} \not\rightarrow \text{flies} \not\rightarrow \text{swims} \hspace{1cm} (12) \]

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 \( \mathcal{Y} \), 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 \textit{inheritable} paths. To be inheritable, a path must satisfy a number of conditions; the first of them is that the path must be \textit{constructible}.

\textbf{Definition 19 (Constructibility)} A context is a pair \( \langle \mathcal{Y}, \Pi \rangle \), where \( \mathcal{Y} \) is a MINE and \( \Pi \) is a set of paths. A positive path \( \pi_1 = x \rightarrow \pi' \rightarrow u \rightarrow y \) is constructible in a context \( \langle \mathcal{Y}, \Pi \rangle \) iff

1. \( x \rightarrow \pi' \rightarrow u \in \Pi \), for some (possibly empty) positive path \( \pi' \);
2. \( u \rightarrow y \in \mathcal{Y}_A \).

A negative path \( \pi_2 = x \rightarrow \pi' \rightarrow u \not\rightarrow y \) is constructible in a context \( \langle \mathcal{Y}, \Pi \rangle \) iff

1. \( x \rightarrow \pi' \rightarrow u \in \Pi \), for some (possibly empty) positive path \( \pi' \);
2. $u \not\rightarrow y \in \gamma_A$.  

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 $\pi'$ and $\pi''$, 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 $\pi_1$ is conflicting in $\langle \gamma, \Pi \rangle$ iff $\Pi$ contains a path $\pi_2$ that conflicts with $\pi_1$.

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 $\langle \gamma, \Pi \rangle$ iff there is a node $v \in \gamma_C$ such that

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

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

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

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

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

**Case 1**: $\pi \in \gamma_A$. Then $\langle \gamma, \Pi \rangle \vdash \pi$ iff $\pi$ conflicts with no path in $\Pi$ consisting of a single edge in $\gamma_A$.

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

1. $\pi$ is constructible in $\langle \gamma, \Pi \rangle$; and
2. $\pi$ is not conflicting in $\langle \gamma, \Pi \rangle$; and
3. \( \pi \) is not preempted in \( \langle \gamma, \Pi \rangle \).

Finally,

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

A set of conclusions that may be drawn from a MINE \( \gamma \) 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**: \( \pi \in \gamma_A \). Then \( \langle \gamma, \Pi \rangle \vdash \pi \)
i.e. elements of \( \gamma_A \) are always inheritable, even if they are conflicting in \( \gamma_A \). 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 \( \mathcal{T}\mathcal{D}\mathcal{L}^- \) theories

We are now able to give a translation of MINEs into \( \mathcal{T}\mathcal{D}\mathcal{L}^- \). Let \( \gamma \) be a MINE, and let \( P(\gamma) \) be the set of all paths of \( \gamma \). For every path \( \pi_i \) in \( P(\gamma) \), let us introduce two new monadic predicate symbols, \( M_i^+ \) and \( M_i^- \), into \( MP \). For every node \( x \) in \( \gamma \), 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 \( \Xi(\gamma) \) of \( \gamma \) whenever \( x \rightarrow y \) (resp. \( x \not\rightarrow y \)) holds in the corresponding extension of \( \gamma \).

**Definition 24** Let \( \pi_i \) and \( \pi_j \) be two positive paths of \( P(\gamma) \); \( \pi_j \) is an alternative of \( \pi_i \) in \( P(\gamma) \) (and vice versa) iff

1. \( B(\pi_i) = B(\pi_j) \) and \( E(\pi_i) = E(\pi_j) \);

2. there exists an edge in \( \pi_j \) which does not appear in \( \pi_i \), or vice versa.

Informally, \( \pi_i \) is an alternative path of \( \pi_j \) whenever \( \pi_i \) and \( \pi_j \) 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 $\pi$ in $P(\Upsilon)$. Let us also define $\pi^\pi$ as the TDLc frame $(M_{x_1} \cap \ldots \cap M_{x_n})$.

Definition 26 (Translation) Let $\Upsilon$ be a MINE. The translation $\Xi(\Upsilon)$ of $\Upsilon$ into TDLc is defined as $\Xi(\Upsilon) = \langle \Psi, \Delta \rangle$, where

\[
\Psi = \{ M_x \models \forall x \mid x \rightarrow y \in \Upsilon_A \text{ or } x \not\rightarrow y \in \Upsilon_A \} \cup \{ \text{Isa}(i_x, i_x) \mid x \rightarrow y \in \Upsilon_A \text{ or } x \not\rightarrow y \in \Upsilon_A \} \cup \{ M_x \mathtt{<} M_j^+ \mid \pi_j = x \rightarrow y \in P(\Upsilon) \} \cup \{ M_i^+ \mathtt{<} M_j^+ \mid \pi_j = \pi_i \rightarrow x \in P(\Upsilon) \} \cup \{ M_j^- \mathtt{<} M_i^+ \mid \pi_j \text{ is an alternative of } \pi_i \text{ in } P(\Upsilon) \} \]

\[
\Delta = \{ M_x \models \text{Isa}. M_y \mid x \rightarrow y \in \Upsilon_A \} \cup \{ M_x \models \text{Isa}. \neg M_y \mid x \not\rightarrow y \in \Upsilon_A \} \cup \{ M_i^+ \models \text{Isa}. \pi_j^\pi \cap M_n \mid \pi_j = \pi_i \rightarrow x \in P(\Upsilon) \} \cup \{ M_i^+ \models \text{Isa}. \pi_j^\pi \cap \neg M_n \mid \pi_j = \pi_i \not\rightarrow x \in P(\Upsilon) \} \cup \{ M_i^+ \models \text{Isa}. \pi_j^\pi \cap M_i^- \mid \pi_j \text{ is an alternative of } \pi_i \text{ in } P(\Upsilon) \} \cup \{ M_i^- \models \text{Isa}. \pi_j^\pi \cap M_i \mid \pi_k = \pi_j \rightarrow x \in P(\Upsilon) \} \text{ and } \\
\begin{align*}
\pi_j & \text{ is a subpath of } \pi_i \} \cup \\
\{ M_i^- \models \text{Isa}. \pi_j^\pi \cap \neg M_n \mid \pi_k = \pi_j \not\rightarrow x \in P(\Upsilon) \} \text{ and } \\
\pi_j & \text{ is a subpath of } \pi_i \}
\]

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

Our aim is to capture in the translation $\Xi$ 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:

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

Note that $\pi''$ is an alternative path of $x \rightarrow \pi_i \rightarrow u$, whereas $\pi''$ and $x \rightarrow \pi_i' \rightarrow v$ are subpaths of $\pi$. Then path $\pi$ would be inheritable in a context iff
1. \(x \rightarrow \pi_i \rightarrow u\) is inheritable;

2. there is no inheritable alternative path \(\pi'' = x \rightarrow \pi'_i \rightarrow v \rightarrow \pi''_i \rightarrow u\) 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 \(\pi\));

2. Condition 2 is captured by Clause 17 (applied to path \(\pi''\)), by Clause 22 (applied to path \(\pi''\)) and by Clause 24 (applied to path \(\pi'\) with \(x \rightarrow \pi'_i \rightarrow v\) 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 \(\Upsilon\) be a MINE. Then \(S\) is a conclusion set of \(\Upsilon\) iff \(\exists(\Upsilon)\) has an extension \(E\) such that:

1. \(x \rightarrow y \in S\) iff \(M_y[i_x] \in E\) and \(M_y \neq M_x\);

2. \(x \not\rightarrow y \in S\) iff \(-M_y[i_x] \in E\) 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 \(S\) be a conclusion set of \(\Upsilon\), obtained from the extension \(\Pi\) of \(\Upsilon\). Let \(E\) be the set \(TC(\Psi \cup \Lambda)\) where \(\alpha \in \Lambda\) iff

1. \(\alpha = M_y[i_x], \ x \rightarrow y \in S\) and \(M_x \neq M_y\);

2. \(\alpha = -M_y[i_x], \ x \not\rightarrow y \in S\) and \(M_x \neq M_y\);

3. \(\alpha = M^{-}_i[i_{B(\bar{\pi})}]\) and \(\bar{\pi} \in \Pi\).

We now show that \(E = \Theta(E, T)\); thus \(x \rightarrow y \in S\) implies that \(M_y[i_x] \in E\) and that \(M_x \neq M_y\). First at all, it is easy to see that \(E\) satisfies Conditions 1 and 2 of Definition 11. Let us then show that \(E \subseteq \Theta(E, T)\). For \(M_y[i_x] \in E\) we have \(x \rightarrow y \in S\). 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 \(M^{+}_i \leftarrow Isa.\pi_i \cap M_y\) and \(M^{+}_i[i_x] \in E\). Thus, \(M_y[i_x] \in \Theta(E, T)\). Finally, let us show that \(\Theta(E, T) \subseteq E\) Suppose instead that \(\Theta(E, T) \not\subseteq E\). Then it can be shown that there is \(M_y[i_x] \notin E\) such that \(M_y[i_x] \in \Theta(E, T)\) and \(x \rightarrow y \in S\). Therefore, \(M_y[i_x] \in E\); a contradiction. \(\blacksquare\)
The sense of this theorem will be best illustrated by means of an example:

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

- the $T$-set $\Psi$ 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;
  - $\{Isa[i_p, i_r], Isa[i_r, i_q], Isa[i_q, i_s]\}$, from line (14) of Definition 26;
  - $\{M_p \prec M_{pr}, M_p \prec M_{pq}, M_q \prec M_{pq}, M_r \prec M_{rq}\}$, from line (15) of Definition 26;
  - $\{M_{pq} \prec M_{pq}, M_{pr} \prec M_{pq}, M_{pr} \prec M_{pq}, M_{pq} \prec M_{pq}\}$, from line (16) of Definition 26;
  - $\{M_{pq} \prec M_{pq}, M_{pq} \prec M_{pq}\}$, from line (17) of Definition 26;

- the set of defaults $\Delta$ is composed of the union of the following sets:
  - $\{M_p \rightarrow Isa.M_r, M_p \rightarrow Isa.M_q, M_r \rightarrow Isa.M_q, M_q \rightarrow Isa.M_q\}$, from line (18) of Definition 26;
  - $\{M_p \rightarrow Isa.\neg M_s\}$, from line (19) of Definition 26;
  - $\{M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_s), M_{pq} \rightarrow Isa.(M_p \cap M_q \cap M_q \cap M_{pq}), M_{pr} \rightarrow Isa.(M_p \cap M_q \cap M_q \cap M_{pq})\}$, from line (20) of Definition 26;
  - $\{M_{pr} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_{pi})\}$, from line (21) of Definition 26;
  - $\{M_{pq} \rightarrow Isa.(M_p \cap M_q \cap M_{pq} \cap M_{pq}), M_{pq} \rightarrow Isa.(M_p \cap M_q \cap M_q \cap M_{pq}), M_{pq} \rightarrow Isa.(M_p \cap M_q \cap M_q \cap M_{pq})\}$, from line (22) of Definition 26;
  - $\{M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_4), M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_4), M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_4)\}$, from line (23) of Definition 26;
  - $\{M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_4), M_{pq} \rightarrow Isa.(M_p \cap M_r \cap M_q \cap M_4)\}$, from line (24) of Definition 26.

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

---

7In 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 $pq$, while the negative path $p \rightarrow q \not\rightarrow s$ will be indicated by $\overline{pq}$. 

---

26
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


28


