Fundamenta Informaticae 96 (2009) 1–32
DOI 10.3233/FI-2009-185
IOS Press

$\mathcal{ALC} + \mathbf{T}$: a Preferential Extension of Description Logics

Laura Giordano
Dipartimento di Informatica
Università del Piemonte Orientale "A. Avogadro"
viale Teresa Michel, 11 - 15121, Alessandria, Italy
laura@mfn.unipmn.it

Valentina Gliozzi
Dipartimento di Informatica
Università degli Studi di Torino
c.so Svizzera 185, 10145 Torino, Italy
gliozzi@di.unito.it

Nicola Olivetti
LSIS-UMR CNRS 6168
Université "Paul Cézanne"
Aix en Provence, France
nicola.olivetti@univ-cezanne.fr

Gian Luca Pozzato[*]
Dipartimento di Informatica
Università degli Studi di Torino
c.so Svizzera 185, 10145 Torino, Italy
pozzato@di.unito.it

Abstract. We extend the Description Logic $\mathcal{ALC}$ with a "typicality" operator $\mathbf{T}$ that allows us to reason about the prototypical properties and inheritance with exceptions. The resulting logic is called $\mathcal{ALC} + \mathbf{T}$. The typicality operator is intended to select the "most normal" or "most typical" instances of a concept. In our framework, knowledge bases may then contain, in addition to ordinary ABoxes and TBoxes, subsumption relations of the form "$\mathbf{T}(C)$ is subsumed by P", expressing that typical C-members have the property P. The semantics of a typicality operator is defined by a set of postulates that are strongly related to Kraus-Lehmann-Magidor axioms of preferential logic **P**. We first show that $\mathbf{T}$ enjoys a simple semantics provided by ordinary structures equipped with a preference relation. This allows us to obtain a modal interpretation of the typicality operator. We show that the satisfiability of an $\mathcal{ALC}+\mathbf{T}$ knowledge base is decidable and it is precisely EXPTIME. We then present a tableau calculus for deciding satisfiability of $\mathcal{ALC} + \mathbf{T}$ knowledge bases. Our calculus gives a (suboptimal) nondeterministic-exponential time decision procedure for $\mathcal{ALC} + \mathbf{T}$. We finally discuss how to extend $\mathcal{ALC} + \mathbf{T}$ in order to infer defeasible properties of (explicit or implicit) individuals. We propose two alternatives: (i) a nonmonotonic completion of a knowledge base; (ii) a "minimal model" semantics for $\mathcal{ALC} + \mathbf{T}$ whose intuition is that minimal models are those that maximise typical instances of concepts.

Keywords: Description Logics, Prototypical Reasoning, Tableaux Calculi.

1. Introduction

The family of description logics (DLs, [1]) is one of the most important formalisms of knowledge representation. DLs are reminiscent of the early semantic networks and of frame-based systems. They offer

[*] Address for correspondence: Dipartimento di Informatica, Università degli Studi di Torino, c.so Svizzera 185, 10145 Torino, Italy

two key advantages: a well-defined semantics based on first-order logic and a good trade-off between expressivity and complexity. DLs have been successfully implemented by a range of systems and they are at the base of languages for the semantic web such as OWL. A DL knowledge base (KB) comprises two components: (i) the TBox, containing the definition of concepts (and possibly roles) and a specification of inclusion relations among them, and (ii) the ABox containing instances of concepts and roles, in other words, properties and relations of individuals. Since the primary objective of the TBox is to build a taxonomy of concepts, the need of representing prototypical properties and of reasoning about defeasible inheritance of such properties easily arises. The traditional approach is to handle defeasible inheritance by integrating some kind of nonmonotonic reasoning mechanism. This has led to study nonmonotonic extensions of DLs [2, 3, 6, 5, 4, 8, 10, 11, 23]. However, finding a suitable nonmonotonic extension for inheritance reasoning with exceptions is far from obvious.

In this work, we propose an approach to defeasible inheritance reasoning based on the typicality operator $\mathbf{T}$. The intended meaning is that, for any concept C, $\mathbf{T}(C)$ singles out the instances of C that are considered as "typical" or "normal". Thus assertions as "normally students do not pay taxes" [13] or "typically mammals inhabit land" [4] are represented by $\mathbf{T}(Student) \sqsubseteq \neg\, TaxPayer$ and $\mathbf{T}(Mammal) \sqsubseteq \exists Habitat.Land$.

Before entering in the technical details, let us sketch how we intend to use the typicality operator and what kind of inferential services we expect to profit. We assume that a KB comprises, in addition to the standard TBox and ABox, a set of assertions of the type $\mathbf{T}(C) \sqsubseteq D$, where $\mathbf{T}$ does not occur in D. The reasoning system should be able to infer prototypical properties as well as to ascribe defeasible properties to individuals. For instance, let the KB contain:

$$\mathbf{T}(ItalianFencer) \sqsubseteq \neg LovedByPeople$$
$$\mathbf{T}(ItalianFencer \sqcap OlympicGoldMedalist) \sqsubseteq LovedByPeople$$
$$\mathbf{T}(ItalianFencer \sqcap OlympicGoldMedalist \sqcap \exists\, TakePart.RealityShow) \sqsubseteq \neg LovedByPeople$$

corresponding to the assertions: normally an Italian fencer is not a people's favourite (fencing is not so popular in Italy...), but normally an Italian fencer who won a gold medal in an Olympic competition is a people's favourite, whereas normally an Italian Olympic gold medalist in fencing who has taken part to a reality show is not a people's favourite (because he has lost his passion in sport and his determination to obtain better and better results...). Observe that, if the same properties were expressed by ordinary inclusions, such as $ItalianFencer \sqsubseteq \neg LovedByPeople$, we would simply get that there are not Italian gold medalists in fencing and so on, thus the KB would collapse. This collapse is avoided as we do not assume that $\mathbf{T}$ is monotonic, that is to say $C \sqsubseteq D$ does not imply $\mathbf{T}(C) \sqsubseteq \mathbf{T}(D)$. Suppose next that the ABox contains the following facts about the individuals $oronzo$, $aldo$ and $luca$:

1. $ItalianFencer(oronzo)$
2. $ItalianFencer(aldo)$, $OlympicGoldMedalist(aldo)$
3. $ItalianFencer(luca)$, $OlympicGoldMedalist(luca), \exists\, TakePart.RealityShow(luca)$

Then the reasoning system should be able to infer the expected (defeasible) conclusions:

1. $\neg LovedByPeople(oronzo)$
2. $LovedByPeople(aldo)$
3. $\neg LovedByPeople(luca)$

As a further step, the system should be able to infer (defeasible) properties also of individuals implicitly introduced by existential restrictions. For instance, if the ABox further contains

$$\exists \mathit{HasChild}.\mathit{ItalianFencer}(\mathit{mario})$$

it should conclude (defeasibly)

$$\exists \mathit{HasChild}.\neg \mathit{LovedByPeople}(\mathit{mario})$$

Given the nonmonotonic character of the $\mathbf{T}$ operator, there is a difficulty with handling irrelevant information. For instance, given the KB as above, one should be able to infer as well:

$$\mathbf{T}(\mathit{ItalianFencer} \sqcap \mathit{SlimPerson}) \sqsubseteq \neg \mathit{LovedByPeople}$$
$$\mathbf{T}(\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \mathit{SlimPerson}) \sqsubseteq \mathit{LovedByPeople}$$

as $\mathit{SlimPerson}$ is irrelevant with respect to being loved by people or not. For the same reason, the conclusion about aldo being a favourite of the people or not should not be influenced by the addition of $\mathit{SlimPerson}(\mathit{aldo})$ to the ABox. We refer to this problem as the problem of Irrelevance.

In this paper we lay down the base of an extension of DL with a typicality operator. Our starting point is a monotonic extension of the basic $\mathcal{ALC}$ with the $\mathbf{T}$ operator. The operator is supposed to satisfy a set of postulates that are essentially a reformulation of Kraus, Lehmann, and Magidor (KLM) axioms of preferential logic, namely, the assertion $\mathbf{T}(C) \sqsubseteq P$ is equivalent to the conditional assertion $C \mid\!\sim P$ of KLM preferential logic **P**. It turns out that the semantics of the typicality operator can be equivalently specified by considering a preference relation (a strict partial order) on individuals: the typical members of a concept C are just the most preferred (or "most normal") individuals of C according to the preference relation. The preference relation is the only additional ingredient that we need in our semantics.

We assume that "most normal" members of a concept C always exist, whenever the concept C is non-empty. This assumption corresponds to the *Smoothness Condition* of KLM logics, or the well-known *Limit Assumption* in conditional logics. Taking advantage of this semantic setting, we can give a modal interpretation to the typicality operator: the modal operator $\Box$ has intuitively the same properties as in Gödel-Löb modal logic G of arithmetic provability.

From a computational viewpoint, we show that the extension of $\mathcal{ALC}$ with the $\mathbf{T}$ operator is decidable and we provide an EXPTIME complexity upper bound. Since reasoning in $\mathcal{ALC}$ alone with arbitrary TBox has already the same complexity, we can conclude that the extension by $\mathbf{T}$ is essentially inexpensive. We also define a tableau proof procedure for $\mathcal{ALC}$ with the $\mathbf{T}$ operator that has, however, a suboptimal upper bound NEXPTIME. Actually we conjecture that the tableau procedure can be made more efficient in order to match the EXPTIME upper bound, by means of optimization techniques developed for $\mathcal{ALC}$. This issue will be part of our future research.

From a knowledge representation viewpoint, however, the monotonic extension is not enough to perform inheritance reasoning of the kind described above. We need a way of inferring defeasible properties of individuals and a way of handling Irrelevance. In the last section, we discuss two different approaches:

- we can define a *completion* of an ABox: the idea is that each individual is assumed to be a typical member of the most specific concept to which it belongs. Such a completion allows to perform inferences as 1.,2.,3. above;

- we can strengthen the semantics of $\mathcal{ALC}+\mathbf{T}$ by proposing a *minimal model semantics*. Intuitively, the idea is to restrict our consideration to models that maximise typical instances of a concept.

The first proposal is computationally easy, but it presents some difficulties. The second proposal is computationally more expensive, but it is more powerful for inheritance reasoning.

2. The Description Logic $\mathcal{ALC}+\mathbf{T}$

We consider an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individuals $\mathcal{O}$. The language $\mathcal{L}$ of the logic $\mathcal{ALC}+\mathbf{T}$ is defined by distinguishing *concepts* and *extended concepts* as follows. Concepts: $A \in \mathcal{C}$ and $\top$ are *concepts* of $\mathcal{L}$; if $C, D \in \mathcal{L}$ and $R \in \mathcal{R}$, then $C \sqcap D, C \sqcup D, \neg C, \forall R.C, \exists R.C$ are *concepts* of $\mathcal{L}$. Extended concepts: if C is a concept, then C and $\mathbf{T}(C)$ are *extended concepts*, and all the boolean combinations of extended concepts are extended concepts of $\mathcal{L}$. A knowledge base is a pair (TBox,ABox). TBox contains subsumptions $C \sqsubseteq D$, where $C \in \mathcal{L}$ is an extended concept of the form either C' or $\mathbf{T}(C')$, and $D \in \mathcal{L}$ is a concept. ABox contains expressions of the form $C(a)$ and aRb where $C \in \mathcal{L}$ is an extended concept, $R \in \mathcal{R}$, and $a, b \in \mathcal{O}$.

In order to provide a semantics to the operator $\mathbf{T}$, we extend the definition of a model used in "standard" terminological logic $\mathcal{ALC}$[1]:

Definition 2.1. (Semantics of $\mathbf{T}$ with selection function)
A model is any structure $\langle \Delta, I, f_{\mathbf{T}} \rangle$, where: Δ is the domain; I is the extension function that maps each extended concept C to $C^I \subseteq \Delta$, and each role R to a $R^I \subseteq \Delta \times \Delta$. I is defined in the usual way (as for $\mathcal{ALC}$) and, in addition, $(\mathbf{T}(C))^I = f_{\mathbf{T}}(C^I)$. $f_{\mathbf{T}} : Pow(\Delta) \rightarrow Pow(\Delta)$ is a function satisfying the following properties (given $S \subseteq \Delta$):

$(f_{\mathbf{T}} - 1)$ $f_{\mathbf{T}}(S) \subseteq S$ $\qquad$ $(f_{\mathbf{T}} - 2)$ if $S \neq \emptyset$, then also $f_{\mathbf{T}}(S) \neq \emptyset$

$(f_{\mathbf{T}} - 3)$ if $f_{\mathbf{T}}(S) \subseteq R$, then $f_{\mathbf{T}}(S) = f_{\mathbf{T}}(S \cap R)$ $\qquad$ $(f_{\mathbf{T}} - 4)$ $f_{\mathbf{T}}(\bigcup S_i) \subseteq \bigcup f_{\mathbf{T}}(S_i)$

$(f_{\mathbf{T}} - 5)$ $\bigcap f_{\mathbf{T}}(S_i) \subseteq f_{\mathbf{T}}(\bigcup S_i)$

Intuitively, given the extension of some concept C, $f_{\mathbf{T}}$ selects the *typical* instances of C. $(f_{\mathbf{T}} - 1)$ requests that typical elements of S belong to S. $(f_{\mathbf{T}} - 2)$ requests that if there are elements in S, then there are also *typical* such elements. The following properties constrain the behavior of $f_{\mathbf{T}}$ with respect to $\cap$ and $\cup$ in such a way that they do not entail monotonicity. According to $(f_{\mathbf{T}} - 3)$, if the typical elements of S are in R, then they coincide with the typical elements of $S \cap R$, thus expressing a weak form of monotonicity (namely, *cautious monotonicity*). $(f_{\mathbf{T}} - 4)$ corresponds to one direction of the equivalence $f_{\mathbf{T}}(\bigcup S_i) = \bigcup f_{\mathbf{T}}(S_i)$, so that it does not entail monotonicity. Similar considerations apply to the equation $f_{\mathbf{T}}(\bigcap S_i) = \bigcap f_{\mathbf{T}}(S_i)$, of which only the inclusion $\bigcap f_{\mathbf{T}}(S_i) \subseteq f_{\mathbf{T}}(\bigcap S_i)$ is derivable. $(f_{\mathbf{T}} - 5)$ is a further constraint on the behavior of $f_{\mathbf{T}}$ with respect to arbitrary unions and intersections; it would be derivable if $f_{\mathbf{T}}$ were monotonic.

We can give an alternative semantics for $\mathbf{T}$ based on a preference relation. The idea is that there is a global preference relation among individuals and that the typical members of a concept C (i.e., those selected by $f_{\mathbf{T}}(C^I)$) are the minimal elements of C with respect to this relation. Observe that this notion is global, that is to say, it does not compare individuals with respect to a specific concept (something like

[1] We refer to [1] for a detailed description of the standard Description Logic $\mathcal{ALC}$.

y is more typical than x with respect to concept C). In this framework, an element $x \in \Delta$ is a *typical instance* of some concept C if $x \in C^I$ and there is no C-element in Δ *more typical* than x. The typicality preference relation is partial since it is not always possible to establish which element is more typical than which other. The following definition is needed before we provide the Representation Theorem.

Definition 2.2. Given a preference relation $<$, which is a strict partial order (i.e., an irreflexive and transitive relation) over a domain Δ, for all $S \subseteq \Delta$, we define $Min_<(S) = \{x : x \in S$ and $\nexists y \in S$ s.t. $y < x\}$. We say that $<$ satisfies the *Smoothness Condition* iff for all $S \subseteq \Delta$, for all $x \in S$, either $x \in Min_<(S)$ or $\exists y \in Min_<(S)$ such that $y < x$.

Now we are ready to provide the Representation Theorem, showing that, given a model with a selection function, we can define *on the same domain* a preference relation $<$ such that, for all $S \subseteq \Delta$, $f_{\mathbf{T}}(S) = Min_<(S)$. Notice that, as a difference with respect to related results (Theorem 3 in [18]), the relation is defined on the same domain Δ of $f_{\mathbf{T}}$. On the other hand, if $<$ is a strict partial order satisfying the Smoothness Condition, then the operator defined as $f_{\mathbf{T}}(S) = Min_<(S)$ satisfies the postulates of Definition 2.1. In order to give a formal proof, we also need the following lemma:

Lemma 2.1. $f_{\mathbf{T}}(S \cup R) \cap S \subseteq f_{\mathbf{T}}(S)$

Proof:
First, consider $f_{\mathbf{T}}((S \cup R) \cap S)$. Since $(S \cup R) \cap S = S$, it follows that $f_{\mathbf{T}}((S \cup R) \cap S) = f_{\mathbf{T}}(S)$. Hence $f_{\mathbf{T}}((S \cup R) \cap S) \subseteq f_{\mathbf{T}}(S) \cup (\Delta - S)$. Consider now $f_{\mathbf{T}}((S \cup R) \cap (\Delta - S))$. By $(f_{\mathbf{T}} - 1)$, it follows that $f_{\mathbf{T}}((S \cup R) \cap (\Delta - S)) \subseteq \Delta - S$, hence also $f_{\mathbf{T}}((S \cup R) \cap (\Delta - S)) \subseteq f_{\mathbf{T}}(S) \cup (\Delta - S)$. Finally, from $(f_{\mathbf{T}} - 4)$, also $f_{\mathbf{T}}(S \cup R) \subseteq f_{\mathbf{T}}(S) \cup (\Delta - S)$. From this, it can be easily derived that $f_{\mathbf{T}}(S \cup R) \cap S \subseteq f_{\mathbf{T}}(S)$. □

Theorem 2.1. (Representation Theorem)
Given any model $\langle \Delta, I, f_{\mathbf{T}} \rangle$, $f_{\mathbf{T}}$ satisfies postulates $(f_{\mathbf{T}} - 1)$ to $(f_{\mathbf{T}} - 5)$ above iff it is possible to define on Δ a strict partial order $<$, satisfying the Smoothness Condition, such that for all $S \subseteq \Delta$, $f_{\mathbf{T}}(S) = Min_<(S)$.

Proof:
("*Only if*" direction) Given $f_{\mathbf{T}}$ satisfying postulates ($f_{\mathbf{T}} - 1$) to ($f_{\mathbf{T}} - 5$), we define $<$ as follows: for all $x, y \in \Delta$, we let $x < y$ if $\forall S \subseteq \Delta$, if $y \in f_{\mathbf{T}}(S)$, then $x \notin S$ and $\exists R \subseteq \Delta$ such that $S \subset R$ and $x \in f_{\mathbf{T}}(R)$. We prove that $<$ is irreflexive, transitive, and satisfies the Smoothness Condition. Moreover, we prove that, for all $S \subseteq \Delta$, we have $f_{\mathbf{T}}(S) = Min_<(S)$:

1. $<$ is irreflexive and transitive. Irreflexivity follows from the definition of $<$. For transitivity, let (a) $x < y$ and (b) $y < z$. Let $z \in f_{\mathbf{T}}(S)$ for some S, then, by definition of $<$, $y \notin S$ and $\exists R$ s.t. $S \subset R$ and $y \in f_{\mathbf{T}}(R)$. Furthermore, $x \notin R$ and $\exists Q : R \subset Q$ and $x \in f_{\mathbf{T}}(Q)$. From this, we can conclude that $x \notin S$ (otherwise $x \in R$) and $S \subset Q$, and hence $x < z$.
2. $f_{\mathbf{T}}(S) \subseteq Min_<(S)$. Let $x \in f_{\mathbf{T}}(S)$. Suppose $x \notin Min_<(S)$, i.e., for some $y \in S$, $y < x$. By definition of $<$, $y \notin S$, contradiction, hence $x \in Min_<(S)$.
3. $Min_<(S) \subseteq f_{\mathbf{T}}(S)$. Let $x \in Min_<(S)$. Then $x \in S$, i.e., $S \neq \emptyset$. By ($f_{\mathbf{T}} - 2$), $f_{\mathbf{T}}(S) \neq \emptyset$. Suppose $x \notin f_{\mathbf{T}}(S)$. Consider $\bigcup R_i$ for all $R_i \subseteq \Delta$ s.t. $x \in f_{\mathbf{T}}(R_i)$. By ($f_{\mathbf{T}} - 5$), we have $x \in f_{\mathbf{T}}(\bigcup R_i)$. Consider now $f_{\mathbf{T}}(\bigcup R_i \cup S)$. We can show that $f_{\mathbf{T}}(\bigcup R_i \cup S) \not\subseteq \bigcup R_i$, since

otherwise, by ($f_{\mathbf{T}} - 3$) we would have $f_{\mathbf{T}}(\bigcup R_i \cup S) = f_{\mathbf{T}}(\bigcup R_i)$, and by Lemma 2.1 we would conclude $f_{\mathbf{T}}(\bigcup R_i) \cap S \subseteq f_{\mathbf{T}}(S)$, which contradicts the fact that $x \in f_{\mathbf{T}}(\bigcup R_i)$, but $x \notin f_{\mathbf{T}}(S)$. Consider hence $y \in f_{\mathbf{T}}(\bigcup R_i \cup S)$ s.t. $y \notin \bigcup R_i$. We can observe that $y < x$: indeed, $x \in f_{\mathbf{T}}(\bigcup R_i)$, whereas $y \notin \bigcup R_i$; however, $\bigcup R_i \subset \bigcup R_i \cup S$ and $y \in f_{\mathbf{T}}(\bigcup R_i \cup S)$, then $y < x$ by the definition of $<$. Furthermore, since $y \in f_{\mathbf{T}}(\bigcup R_i \cup S)$, by ($f_{\mathbf{T}} - 1$) we have that $y \in \bigcup R_i \cup S$ and, since $y \notin \bigcup R_i$, we conclude that $y \in S$. It follows that $x \notin Min_<(S)$, contradiction, hence $Min_<(S) \subseteq f_{\mathbf{T}}(S)$.

4. $<$ satisfies the Smoothness Condition. Let $S \neq \emptyset$ and $x \in S$. If $x \in f_{\mathbf{T}}(S)$ then by point 2 we have $x \in Min_<(S)$. If $x \notin f_{\mathbf{T}}(S)$, we can reason as for point 3 to conclude that there is $y \in f_{\mathbf{T}}(\bigcup R_i \cup S)$ s.t. $y \notin \bigcup R_i$ (hence $y \in S$), and $y < x$. By Lemma 2.1, we have $y \in f_{\mathbf{T}}(S)$, hence by point 2 we conclude $y \in Min_<(S)$.

("*If*" direction) Given a strict partial order $<$ satisfying the Smoothness Condition, we can define $f_{\mathbf{T}} : Pow(\Delta) \rightarrow Pow(\Delta)$ by letting $f_{\mathbf{T}}(S) = Min_<(S)$. It can be easily shown that $f_{\mathbf{T}}$ satisfies postulates ($f_{\mathbf{T}} - 1$) to ($f_{\mathbf{T}} - 5$). The proof is left to the reader.

□

Having the above Representation Theorem, from now on, we will refer to the following semantics:

Definition 2.3. (Semantics of $\mathcal{ALC} + \mathbf{T}$)
A model $\mathcal{M}$ is any structure $\langle \Delta, <, I \rangle$, where Δ and I are defined as in Definition 2.1, and $<$ is a strict partial order over Δ satisfying the Smoothness Condition (see Definition 2.2 above). I is the extension function that maps each extended concept C to $C^I \subseteq \Delta$, and each role R to a $R^I \subseteq \Delta \times \Delta$. I is defined in the usual way (as for $\mathcal{ALC}$) and, in addition, $(\mathbf{T}(C))^I = Min_<(C^I)$.

Definition 2.4. (Model satisfying a Knowledge Base)
Consider a model $\mathcal{M}$, as defined in Definition 2.3. We extend I so that it assigns to each individual a of $\mathcal{O}$ an element a^I of the domain Δ. Given a KB (TBox,ABox), we say that:

- $\mathcal{M}$ satisfies TBox if for all inclusions $C \sqsubseteq D$ in TBox, for all elements $x \in \Delta$, if $x \in C^I$, then $x \in D^I$.
- $\mathcal{M}$ satisfies ABox if: (i) for all $C(a)$ in ABox, we have that $a^I \in C^I$, (ii) for all aRb in ABox, we have that $(a^I, b^I) \in R^I$.

$\mathcal{M}$ satisfies a knowledge base if it satisfies both its TBox and its ABox.

Notice that the meaning of $\mathbf{T}$ consists of two parts: for any element x of the domain Δ, $x \in (\mathbf{T}(C))^I$ just in case (i) $x \in C^I$, and (ii) there is no $y \in C^I$ such that $y < x$. In order to formalize (ii) in the calculus that we present in Section 3, we introduce a new modality $\Box$ whose interpretation in $\mathcal{M}$ is defined as follows.

Definition 2.5. $(\Box C)^I = \{x \in \Delta \mid$ for every $y \in \Delta$, if $y < x$ then $y \in C^I\}$

The basic idea is simply to interpret the preference relation $<$ as an accessibility relation. By the Smoothness Condition, it turns out that the modality $\Box$ has the properties of Gödel-Löb modal logic of provability G. The Smoothness Condition ensures that typical elements of C^I exist whenever $C^I \neq \emptyset$, by preventing infinitely descending chains of elements. This condition therefore corresponds to the finite-chain condition on the accessibility relation (as in G). A similar correspondence has been presented in [14] to interpret the preference relation in KLM logics. The following relation between $\mathbf{T}$ and $\Box$ holds:

Proposition 2.1. For all $x \in \Delta$, we have $x \in (\mathbf{T}(C))^I$ iff $x \in C^I$ and $x \in (\Box\neg C)^I$

Since we only use $\Box$ to capture the meaning of $\mathbf{T}$, in the following we will always use $\Box$ followed by a negated concept, as in $\Box\neg C$.

Let us now show that the satisfiability of an $\mathcal{ALC} + \mathbf{T}$-knowledge base is a decidable problem and it is precisely EXPTIME-complete. In order to do this, we need some more definitions. First of all, we define the language $\mathcal{L}_{KB}$ of all subformulas of KB plus all boxed formulas $\Box\neg C$ such that $\mathbf{T}(C)$ occurs in KB:

Definition 2.6. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox), we define the language $\mathcal{L}_{KB}$ as follows: - if C is a subformula of KB, then $C \in \mathcal{L}_{KB}$ and $\neg C \in \mathcal{L}_{KB}$; - if $\mathbf{T}(C)$ occurs in KB, then $\Box\neg C \in \mathcal{L}_{KB}$ and $\neg\Box\neg C \in \mathcal{L}_{KB}$.

Let us now define the *saturation* of a set $\mathcal{A}$ of formulas of $\mathcal{L}_{KB}$:

Definition 2.7. Let KB=(TBox,ABox) be an $\mathcal{ALC} + \mathbf{T}$ knowledge base, and let $\mathcal{A}$ be a set of formulas of $\mathcal{L}_{KB}$, i.e., $\mathcal{A} \subseteq \mathcal{L}_{KB}$. We say that $\mathcal{A}$ is saturated if the following conditions hold:

- if $\neg C \in \mathcal{L}_{KB}$, then $C \in \mathcal{A}$ if and only if $\neg C \not\in \mathcal{A}$;
- if $C \sqcup D \in \mathcal{L}_{KB}$, then $C \sqcup D \in \mathcal{A}$ if and only if $C \in \mathcal{A}$ or $D \in \mathcal{A}$;
- if $C \sqcap D \in \mathcal{L}_{KB}$, then $C \sqcap D \in \mathcal{A}$ if and only if $C \in \mathcal{A}$ and $D \in \mathcal{A}$;
- if $\mathbf{T}(C) \in \mathcal{L}_{KB}$, then $\mathbf{T}(C) \in \mathcal{A}$ if and only if $C \in \mathcal{A}$ and $\Box\neg C \in \mathcal{A}$;
- if $\neg\mathbf{T}(C) \in \mathcal{L}_{KB}$, then $\neg\mathbf{T}(C) \in \mathcal{A}$ if and only if $\neg C \in \mathcal{A}$ or $\neg\Box\neg C \in \mathcal{A}$;
- if $C \sqsubseteq D \in$ TBox, then if $C \in \mathcal{A}$ then $D \in \mathcal{A}$.

Intuitively, given a set of formulas $\mathcal{A}$ of $\mathcal{L}_{KB}$, we say that it is saturated if it is free of obvious contradictions, i.e., for all $C \in \mathcal{L}_{KB}$, either $C \in \mathcal{A}$ or $\neg C \in \mathcal{A}$, but not both. In the following a saturated set $\mathcal{A} \subseteq \mathcal{L}_{KB}$ will also be called an *atom*, and the set of all atoms $\mathcal{A} \subseteq \mathcal{L}_{KB}$ will be denoted by $At(\mathcal{L}_{KB})$. It is easy to see that a set of concepts $S = \{C_1, C_2, \ldots, C_n\}$ of $\mathcal{L}_{KB}$ such that $C_1 \sqcap C_2 \sqcap \ldots \sqcap C_n$ is satisfiable in some model of the TBox, can be extended to a saturated set $\mathcal{A} \in At(\mathcal{L}_{KB})$.

We can show that, given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox) such that the TBox is consistent and the ABox is consistent w.r.t. the TBox, we can build a canonical model for KB by means of a construction whose temporal complexity is exponential in the size of the KB. The basic idea is that of building a canonical model whose domain is a subset of the set of atoms $At(\mathcal{L}_{KB})$.

Definition 2.8. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox), we define a *canonical* model $\mathcal{M} = \langle \Delta, <, I \rangle$ iteratively through the following steps:

(Step 1) Let:

- $\Delta = At(\mathcal{L}_{KB})$
- I be the following extension function:
  - *concepts*: given $\mathcal{A} \in \Delta$, let $\mathcal{A} \in C^I$ if and only if $C \in \mathcal{A}$;
  - *roles*: given $\mathcal{A} \in \Delta$ and $\mathcal{B} \in \Delta$, let $(\mathcal{A}, \mathcal{B}) \in R^I$ if and only if there exists $\exists R.C \in \mathcal{A}$ (resp. $\neg\forall R.C \in \mathcal{A}$) such that: (i) $C \in \mathcal{B}$ (resp. $\neg C \in \mathcal{B}$); (ii) for all $\forall R.D \in \mathcal{A}$ (resp. for all $\neg\exists R.D \in \mathcal{A}$), $D \in \mathcal{B}$ (resp. $\neg D \in \mathcal{B}$);

- $<_0$ be the following relation:
  - given $\mathcal{A} \in \Delta$ and $\mathcal{B} \in \Delta$, let $\mathcal{A} <_0 \mathcal{B}$ if and only if there exists $\neg\Box\neg C \in \mathcal{B}$ such that: (i) $C \in \mathcal{A}$ and $\Box\neg C \in \mathcal{A}$; (ii) for each $\Box\neg D \in \mathcal{B}$, $D \in \mathcal{A}$ and $\Box\neg D \in \mathcal{A}$;

(Step 2) Repeatedly update Δ, I and $<_0$ as follows:

- remove from Δ all $\mathcal{A}$ such that: $\exists R.C \in \mathcal{A}$ (resp. $\neg\forall R.C \in \mathcal{A}$) and there is no $\mathcal{B} \in \Delta$ such that $(\mathcal{A}, \mathcal{B}) \in R^I$ and $C \in \mathcal{B}$ (resp. $\neg C \in \mathcal{B}$);
- remove from Δ all $\mathcal{B}$ such that: $\neg\Box\neg C \in \mathcal{B}$ and there is no $\mathcal{A} \in \Delta$ such that $\mathcal{A} <_0 \mathcal{B}$ and $C \in \mathcal{A}$ and $\Box\neg C \in \mathcal{A}$;
- Update I and $<_0$ accordingly (by removing from each C^I all the atoms which have been removed from Δ; by removing from R^I all the pairs $(\mathcal{A}, \mathcal{B})$ such that either $\mathcal{A}$ or $\mathcal{B}$ has been removed from Δ, etc.)

until no further deletion of atoms from Δ is possible.

(Step 3) Define the preference relation $<$ as the transitive closure of $<_0$.

Observe that, (Step 2) above removes from the domain Δ those domain elements containing existential (or negated Box) concepts of which that domain element cannot be an instance. Due to the finiteness of the set Δ as defined in (Step 1), the number of iterations in (Step 2) must be finite.

It is easy to prove that, given the canonical model $\mathcal{M} = \langle \Delta, <, I \rangle$ of Definition 2.8, the function I is defined according to the semantics of $\mathcal{ALC} + \mathbf{T}$, namely:
- $\mathcal{A} \in (D \sqcup E)^I$ if and only if either $\mathcal{A} \in D^I$ or $\mathcal{A} \in E^I$;
- $\mathcal{A} \in (D \sqcap E)^I$ if and only if $\mathcal{A} \in D^I$ and $\mathcal{A} \in E^I$;
- $\mathcal{A} \in (\neg D)^I$ if and only if $\mathcal{A} \notin D^I$;
- $\mathcal{A} \in (\exists R.C)^I$ if and only if there is a $\mathcal{B} \in \Delta$ such that $(\mathcal{A}, \mathcal{B}) \in R^I$ and $\mathcal{B} \in C^I$;
- $(\mathbf{T}(C))^I = Min_<(C^I)$.
The proof of all cases is obvious apart from the last one, for which we prove the next two lemmas.

Lemma 2.2. Let KB=(TBox,ABox) be an $\mathcal{ALC} + \mathbf{T}$ knowledge base and let $\mathcal{M} = \langle \Delta, <, I \rangle$ be the canonical model of Definition 2.8. Given $\mathcal{A} \in \Delta$, if $\Box\neg C \in \mathcal{A}$, then for all $\mathcal{B} < \mathcal{A}$ we have that $\neg C \in \mathcal{B}$ and $\Box\neg C \in \mathcal{B}$.

Proof:
We distinguish two cases: (i) $\mathcal{B} <_0 \mathcal{A}$: by construction, for all $\Box\neg C \in \mathcal{A}$ we have that $\neg C \in \mathcal{B}$ and $\Box\neg C \in \mathcal{B}$ and we are done; (ii) $\mathcal{B} < \mathcal{A}$ has been obtained by the transitive closure of $<_0$ with $\mathcal{B} <_0 \mathcal{A}_1 <_0 \mathcal{A}_2 <_0 \cdots <_0 \mathcal{A}_{n-1} <_0 \mathcal{A}_n <_0 \mathcal{A}$. By construction, for all $\Box\neg C \in \mathcal{A}$, we have that $\neg C \in \mathcal{A}_n$ and $\Box\neg C \in \mathcal{A}_n$. For the same reason, we have that $\neg C \in \mathcal{A}_{n-1}$ and $\Box\neg C \in \mathcal{A}_{n-1}$, and so on, then $\neg C \in \mathcal{B}$ and $\Box\neg C \in \mathcal{B}$, and we are done.

□

Lemma 2.3. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox) and given the canonical model $\mathcal{M} = \langle \Delta, <, I \rangle$ of Definition 2.8, we have that $(\mathbf{T}(C))^I = Min_<(C^I)$.

Proof:
First, we prove that $(\mathbf{T}(C))^I \subseteq Min_<(C^I)$. Consider $\mathcal{A} \in (\mathbf{T}(C))^I$. By the definition of the extension function I in Definition 2.8, we have that $\mathbf{T}(C) \in \mathcal{A}$ since $\mathcal{A} \in (\mathbf{T}(C))^I$. Since $\mathcal{A}$ is saturated (Definition 2.7), we have that $C \in \mathcal{A}$ and $\Box\neg C \in \mathcal{A}$, then (i) $\mathcal{A} \in C^I$ by construction of I in the model. Let us now consider each $\mathcal{B} \in \Delta$ such that $\mathcal{B} < \mathcal{A}$: since $\Box\neg C \in \mathcal{A}$, by Lemma 2.2 we have that $\neg C \in \mathcal{B}$ then, by the definition of the extension function I in the model, we have that (ii) $\mathcal{B} \not\in C^I$. We can conclude that (i) $\mathcal{A} \in C^I$ and (ii) for each $\mathcal{B} < \mathcal{A}$ we have $\mathcal{B} \not\in C^I$, i.e., $\mathcal{A} \in Min_<(C^I)$.
Second, we prove that $Min_<(C^I) \subseteq (\mathbf{T}(C))^I$. Consider $\mathcal{A} \in Min_<(C^I)$. By definition, we have that $\mathcal{A} \in C^I$ and there is not $\mathcal{B} < \mathcal{A}$ such that $\mathcal{B} \in C^I$. Since $\mathcal{A} \in C^I$, we have that (i) $C \in \mathcal{A}$ by the definition of I in Definition 2.8. By absurd, suppose that $\Box\neg C \not\in \mathcal{A}$: since $\mathcal{A}$ is saturated (Definition 2.7), $\neg\Box\neg C \in \mathcal{A}$. By construction of the canonical model, there exists $\mathcal{B}' < \mathcal{A}$ such that $C \in \mathcal{B}'$ and $\Box\neg C \in \mathcal{B}'$ then, by the definition of I, $\mathcal{B}' \in C^I$. Therefore, there exists $\mathcal{B}' < \mathcal{A}$ such that $\mathcal{B}' \in C^I$, against the hypothesis that $\mathcal{A} \in Min_<(C^I)$. We can conclude that (ii) $\Box\neg C \in \mathcal{A}$. Since $\mathcal{A}$ is saturated (Definition 2.7), $\mathbf{T}(C) \in \mathcal{A}$. We conclude that $\mathcal{A} \in (\mathbf{T}(C))^I$ by the definition of I in Definition 2.8. □

To show that the canonical model is indeed a model, we also need to show that the relation $<$ satisfies the properties of a preference relation.

Lemma 2.4. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox) and given the canonical model $\mathcal{M} = \langle \Delta, <, I \rangle$ of Definition 2.8, we have that $<$ is an irreflexive and transitive relation and satisfies the smoothness condition.

Proof:
Transitivity follows from the definition of $<$ from $<_0$. For the irreflexivity, suppose by absurd that $\mathcal{B} < \mathcal{B}$ for some $\mathcal{B} \in \Delta$. We distinguish two cases: (i) $\mathcal{B} <_0 \mathcal{B}$, then there exists $\neg\Box\neg C \in \mathcal{B}$ and, by construction, also $\Box\neg C \in \mathcal{B}$, against the fact that $\mathcal{B}$ is saturated; (ii) $\mathcal{B} < \mathcal{B}$ since $\mathcal{B} < \mathcal{A}_1 < \mathcal{A}_2 < \dots < \mathcal{A}_n < \mathcal{A} < \mathcal{B}$ for some $\mathcal{A}, \mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n \in \Delta$. Since $\mathcal{A} < \mathcal{B}$, there is $\neg\Box\neg C \in \mathcal{B}$ such that $\Box\neg C \in \mathcal{A}$. By Lemma 2.2, for all $\mathcal{B}' < \mathcal{A}$ we have that $\neg C \in \mathcal{B}'$ and $\Box\neg C \in \mathcal{B}'$, then also $\Box\neg C \in \mathcal{B}$ since $\mathcal{B} < \mathcal{A}$, once again against the saturation of $\mathcal{B}$. Concerning the smoothness condition, we have to show that, given a set S, if $S \neq \emptyset$, then also $Min_<(S) \neq \emptyset$. This immediately follows from the fact that $<$ does not contain infinite descending chains of elements of Δ; since $\Delta = At(\mathcal{L}_{KB})$ is finite, then we have only to show that $<$ is acyclic. This immediately follows from irreflexivity. □

We have seen that the canonical model $\mathcal{M}$ defined above is indeed an $\mathcal{ALC} + \mathbf{T}$ model. We will see later that $\mathcal{M}$ satisfies all the inclusion occurring in the TBox of the given KB. However, up to now we have not yet defined how the function I can map individuals occurring in the ABox into domain elements. This mapping should be defined in such a way that the assertions in the ABox are satisfied. How can we be sure that, for each individual a occurring in the Abox, we can find a proper element of the domain, namely an atom $\mathcal{A} \in \Delta$ such that for all the concept C with $C(a) \in$ABox, C belongs to $\mathcal{A}$?

We need to introduce some further Lemmas. Given an atom $\mathcal{A} \in At(\mathcal{L}_{KB})$, where $\mathcal{A} = \{C_1, \dots, C_n\}$, let us denote by $\hat{\mathcal{A}}$ the concept $C_1 \sqcap \dots \sqcap C_n$. The next lemma says that an atom which has not been included in the canonical model corresponds to a set of concepts which are not conjunctively satisfiable (with respect to the TBox).

Lemma 2.5. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox) and given the canonical model $\mathcal{M} = \langle \Delta, <, I \rangle$ of Definition 2.8, for all atoms $\mathcal{A} \in At(\mathcal{L}_{KB})$, if $\mathcal{A} \not\in \Delta$, then in all models $\mathcal{M}' = \langle \Delta', <', I' \rangle$ satisfying the TBox, $\hat{\mathcal{A}}^{I'} = \emptyset$.

Proof:
We give a sketch of the proof. If $\mathcal{A} \in At(\mathcal{L}_{KB})$ and $\mathcal{A} \not\in \Delta$, it is the case that, during the construction of the canonical model, atom $\mathcal{A}$ has been deleted from the domain Δ at some iteration step within (Step 2). The theses can be proved by induction on the order in which atoms have been deleted from Δ during the construction of the model.

Let us consider in detail the base case. Assume $\mathcal{A} \in At(\mathcal{L}_{KB})$ and $\mathcal{A}$ is deleted from Δ at the first iteration step. There are two possible cases: either (1) there is a concept $\exists R.C \in \mathcal{A}$ and there is no $\mathcal{B} \in \Delta$ with $(\mathcal{A}, \mathcal{B}) \in R^I$ and $C \in \mathcal{B}$ or (2) there is a concept $\neg\Box\neg C \in \mathcal{A}$ and there is no $\mathcal{B} \in \Delta$ such that $\mathcal{B} <_0 \mathcal{A}$ and $C \in \mathcal{B}$ and $\Box\neg C \in \mathcal{B}$. Let us consider case (1) (case (2) is similar). By construction of the canonical model, it must be the case that, there is no $\mathcal{B}$ in Δ such that $C \in \mathcal{B}$ and, for all for all $\forall R.D \in \mathcal{A}$ (resp. for all $\neg\exists R.D \in \mathcal{A}$), $D \in \mathcal{B}$ (resp. $\neg D \in \mathcal{B}$). Let $D_1, \ldots, D_k$ be all concepts D_i such that either $\forall R.D_i \in \mathcal{A}$ or $\neg\exists R.D_i \in \mathcal{A}$. It must be that there is no atom $\mathcal{B} \in \Delta$ such that $\{C, D_1, \ldots, D_k\} \subseteq \mathcal{B}$. As it is not the case that $\mathcal{B}$ has been rejected from Δ ($\mathcal{A}$ has been rejected at the first iteration step), then there is no atom $\mathcal{B} \in At(\mathcal{L}_{KB})$ such that $\{C, D_1, \ldots, D_k\} \subseteq \mathcal{B}$. Then the concept $C \sqcap D_1 \sqcap \ldots \sqcap D_k$ is not satisfiable with respect to the TBox, that is, in all models $\mathcal{M}' = \langle \Delta', <', I' \rangle$ satisfying the TBox, $(C \sqcap D_1 \sqcap \ldots \sqcap D_k)^{I'} = \emptyset$. Therefore, in all models $\mathcal{M}'$ satisfying the TBox, $\hat{\mathcal{A}}^{I'} = \emptyset$. We omit the inductive case.

□

Lemma 2.6. Let KB=(TBox,ABox) be an $\mathcal{ALC} + \mathbf{T}$ knowledge base and let $S \subseteq \mathcal{L}_{KB}$. If there is a model $\mathcal{M}' = \langle \Delta', <', I' \rangle$ satisfying the TBox and a domain element $x \in \Delta'$ such that $x \in C_i^{I'}$, for all $C_i \in S$, then there is an atom $\mathcal{A} \in \Delta$ such that $S \subseteq \mathcal{A}$.

Proof:
Assume, by absurdum, that there is no atom $\mathcal{A} \in \Delta$ such that $S \subseteq \mathcal{A}$. Let $\mathcal{A}_1, \ldots, \mathcal{A}_r$ be all the atoms in $At(\mathcal{L}_{KB})$ such that $S \subseteq \mathcal{A}_i$ (if S is not contained in any atom $\mathcal{A}_i \in At(\mathcal{L}_{KB})$, then $\hat{S}$ would not be satisfiable with respect to the TBox, against the hypothesis). By hypothesis, there is a model $\mathcal{M}' = \langle \Delta', <', I' \rangle$ satisfying the TBox and a domain element $x \in \Delta'$ such that $x \in C_i^{I'}$, for all $C_i \in S$. Hence, $\hat{S}^{I'} \neq \emptyset$. Then it is easy to see that, for some $i = 1, .., r$, $\hat{\mathcal{A}_i}^{I'} \neq \emptyset$. By Lemma 2.5, $\mathcal{A}_i \in \Delta$.

□

The last Lemma guarantees that, for a KB=(TBox,ABox), where the TBox is consistent and the ABox is consistent w.r.t. the TBox, given an individual a occurring in the ABox, it is always possible to find an element $\mathcal{A}$ of the domain Δ of the canonical model such that $\{C_1, .., C_n\} \subseteq \mathcal{A}$, where $C_1(a), .., C_n(a)$ are all the assertions concerning a in the ABox. Hence, the assertions of the form $C(a)$ can be made true in the canonical model, by defining $a^I = \mathcal{A}$, for such an $\mathcal{A}$. We have also to ensure that all the assertions of the form $R(a, b)$ can be made true in the canonical model by a proper choice of a^I and b^I.

Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox) and given an individual a explicitly mentioned in the ABox, we write $\sigma(a)$ to denote the set of concepts of which a is an instance, i.e., $\sigma(a) = \{C \in \mathcal{L}_{KB} \mid C(a) \in \text{ ABox}\}$.

Lemma 2.7. Let KB=(TBox,ABox) be an $\mathcal{ALC} + \mathbf{T}$ knowledge base, where the TBox is consistent and the ABox is consistent w.r.t. the TBox, and let $\mathcal{M} = \langle \Delta, <, I \rangle$ be a canonical model as in Definition 2.8. The following properties hold:

1. given an individual a explicitly mentioned in the ABox, there exists $\mathcal{A} \in \Delta$ such that $\sigma(a) \subseteq \mathcal{A}$;
2. if $R(a, b) \in$ ABox, there exist $\mathcal{A}, \mathcal{B} \in \Delta$ such that: (i) $\sigma(a) \subseteq \mathcal{A}$, (ii) $\sigma(b) \subseteq \mathcal{B}$ and, (iii) for all $\forall R.C \in \mathcal{A}$ (resp. for all $\neg\exists R.C \in \mathcal{A}$), we have that $C \in \mathcal{B}$ (resp. $\neg C \in \mathcal{B}$).

Proof:

1. The conclusion follows immediately from Lemma 2.6.
2. As the ABox is consistent w.r.t. the TBox, there must be a model $\mathcal{M}' = \langle \Delta', <', I' \rangle$ satisfying both the TBox and the ABox. In particular, there is a domain element $a^{I'} \in \Delta'$ such that $a^{I'} \in C^{I'}$, for all $C \in \sigma(a)$. Let us consider the set S_a of all the concepts (including those in $\sigma(a)$) of which $a^{I'}$ is an instance. By Lemma 2.6, there must be an atom $\mathcal{A} \in \Delta$ such that $S_a \subseteq \mathcal{A}$.

   Similarly, there is a domain element $b^{I'} \in \Delta'$ such that $b^{I'} \in C^{I'}$, for all $C \in \sigma(b)$. Let us consider the set S_b of all the concepts (including those in $\sigma(b)$) of which $b^{I'}$ is an instance. By Lemma 2.6, there must be an atom $\mathcal{B} \in \Delta$ such that $S_b \subseteq \mathcal{B}$.

   Finally, since $R(a, b) \in$ABox, then it must be that $(a^{I'}, b^{I'}) \in R^{I'}$. Let us consider any $\forall R.C \in \mathcal{A}$. It must be that in the model $\mathcal{M}'$, $a^{I'} \in (\forall R.D)^{I'}$. Then, it must be that $b^{I'} \in D^{I'}$. Hence, $D \in S_b$ and $D \in \mathcal{B}$. This proves our thesis, as we have found two atoms $\mathcal{A}, \mathcal{B} \in \Delta$ satisfying condition 2 above.

□

We can now define a model $\mathcal{M}'$ of a KB:

Definition 2.9. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox), where the TBox is consistent and the ABox is consistent w.r.t. the TBox, and given the canonical model $\mathcal{M} = \langle \Delta, <, I \rangle$ of Definition 2.8, we define the model $\mathcal{M}' = \langle \Delta, <, I' \rangle$ where I' extends I as follows: 1. by properly assigning to each individual a of the ABox an element $\mathcal{A} \in \Delta$, call it $a^{I'}$, such that (i) $\sigma(a) \subseteq \mathcal{A}$ and (ii) for all $R(a, b) \in$ ABox and for all $\forall R.C \in \mathcal{A}$ (resp. $\neg\exists R.C \in \mathcal{A}$) we have $C \in \mathcal{B}$ (resp. $\neg C \in \mathcal{B}$), where $\mathcal{B} = b^{I'}$, and 2. by adding $(a^{I'}, b^{I'}) \in R^{I'}$ for each $R(a, b) \in$ ABox.

It is easy to see that the addition of a pair $(a^{I'}, b^{I'})$ to $R^{I'}$ for each $R(a, b) \in$ ABox preserves the properties of the extension function in the model. In particular:

Lemma 2.8. Let KB=(TBox,ABox) be an $\mathcal{ALC} + \mathbf{T}$ knowledge base, where the TBox is consistent and the ABox is consistent w.r.t. the TBox, and let $\mathcal{M}' = \langle \Delta, <, I' \rangle$ be the model of Definition 2.9. Given $\mathcal{A} \in \Delta$ and $\mathcal{B} \in \Delta$, then if $(\mathcal{A}, \mathcal{B}) \in R^{I'}$ and $\mathcal{A} \in (\forall R.C)^{I'}$ (resp. $\mathcal{A} \in (\neg\exists R.C)^{I'}$), then also $\mathcal{B} \in C^{I'}$ (resp. $\mathcal{B} \in (\neg C)^{I'}$).

Proof:

We distinguish two cases. 1. $(\mathcal{A}, \mathcal{B}) \in R^I$, i.e. $\mathcal{A}$ and $\mathcal{B}$ are already R-related in the canonical model of Definition 2.8: this means that, for all $\forall R.C \in \mathcal{A}$ (resp. for all $\neg\exists R.C \in \mathcal{A}$) we have that $C \in \mathcal{B}$ (resp. $\neg C \in \mathcal{B}$). By construction of $\mathcal{M}'$, we can conclude that $\mathcal{B} \in C^{I'}$ (resp. $\mathcal{B} \in (\neg C)^{I'}$). 2. $(\mathcal{A}, \mathcal{B}) \notin R^I$, while $(\mathcal{A}, \mathcal{B}) \in R^{I'}$: $(\mathcal{A}, \mathcal{B})$ has been added to $R^{I'}$ in step 2 of the construction of $\mathcal{M}'$ to satisfy some $R(a, b) \in$ ABox. In such a case, the conclusion follows immediatley follows from part 2. of Lemma 2.7.

□

We are able to prove that:

Theorem 2.2. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox), where the TBox is consistent and the ABox is consistent w.r.t. the TBox, the model $\mathcal{M}' = \langle \Delta, <, I' \rangle$ of Definition 2.9 satisfies KB.

Proof:
First of all, Lemmas 2.3 and 2.8 show that I' is an extension function in the sense of Definition 2.1. Moreover, Lemma 2.4 shows that $<$ is an irreflexive and transitive relation and satisfies the smoothness condition. We have to show that both the TBox and the ABox are satisfiable in $\mathcal{M}'$:

- TBox: for each $C \sqsubseteq D \in$ TBox, we have to show that, for each $\mathcal{A} \in \Delta$, if $\mathcal{A} \in C^{I'}$ then also $\mathcal{A} \in D^{I'}$. Suppose $\mathcal{A} \in C^{I'}$. By construction, we have that $C \in \mathcal{A}$. Since $\mathcal{A}$ is saturated, we have also that $D \in \mathcal{A}$ and, by construction of the model, $\mathcal{A} \in D^{I'}$, and we are done;
- ABox: we have to show that, for each $C(a) \in$ ABox and for each $R(a, b) \in$ ABox, $a^{I'} \in C^{I'}$ and $(a^{I'}, b^{I'}) \in R^{I'}$. This follows by the construction of $\mathcal{M}'$ in Definition 2.9.

□

We conclude with a complexity analysis of $\mathcal{ALC} + \mathbf{T}$:

Theorem 2.3. Given an $\mathcal{ALC} + \mathbf{T}$ knowledge base KB=(TBox,ABox), the problem of deciding satisfiability of KB is EXPTIME-complete.

Proof:
First, we prove that the problem of deciding satisfiability of KB is in EXPTIME. If KB is satisfiable, we can build a model as defined in Definition 2.9. Let n be the length of the string representing KB: the cardinality of the language $\mathcal{L}_{KB}$ (Definition 2.6) is $O(n)$, then the number of different sets of formulas of $\mathcal{L}_{KB}$ is $O(2^n)$. Also the size of $At(\mathcal{L}_{KB})$ is $O(2^n)$. The size of each set of formulas of $At(\mathcal{L}_{KB})$ is $O(n)$.

The construction of the canonical model in Definition 2.9 starts with (Step 1) the definition of a domain Δ containing $O(2^n)$ atoms, and the definition of the binary relations $<_0$ and R^Is on $At(\mathcal{L}_{KB})$. The definition of each of these relations requires, at most, $O(2^n)$ steps. Indeed, the construction introduces at most $O(2^n) \times O(2^n)$ pairs $(\mathcal{A}, \mathcal{B})$ for each R^I (where the different roles are at most $O(n)$), and the same for $<_0$. Observe that the iterative step (Step 2) requires at most $O(2^n)$ iterations, since each iteration removes at least one atom from Δ. Moreover, each iteration requires at most $O(2^n)$ steps. It is easy to see that also (Step 3) and the assignment of a domain element to each individual occurring in the ABox, as done in Definition 2.9, require at most $O(2^n)$ steps.

The standard DL $\mathcal{ALC}$ is EXPTIME-complete [1]. Since our logic $\mathcal{ALC} + \mathbf{T}$ extends $\mathcal{ALC}$, it immediately follows that the problem of deciding satisfiability of KB is EXPTIME-hard. □

Let us finally consider other standard reasoning tasks for DLs. Given a KB, we consider:

- concept satisfiability: given a concept C, is there a model of KB assigning a non empty extension to C?
- subsumption: given two concepts C and D, is C more general than D in any model of KB?
- instance checking: given an individual a and a concept C, is a an instance of C in any model of KB?

Theorem 2.4. In $\mathcal{ALC}+\mathbf{T}$, given a KB=(TBox,ABox), the problems of *concept satisfiability, subsumption* and of *instance checking* are EXPTIME-complete.

Proof:
The problem of *concept satisfiability* can be reduced to the problem of satisfiability of KB'=(TBox,ABox'), where ABox'=ABox $\cup\{C(a)\}$, with a not occurring in ABox, therefore it is EXPTIME-complete. The problem of *concept satisfiability* extends the problem of satisfiability of a KB, therefore it is EXPTIME-complete. Concerning the problems of *subsumption* and of *instance checking*, let us consider the complementary problem of deciding whether KB $\not\models \alpha$, where α is either $C \sqsubseteq D$ or $C(a)$, respectively. These problems can be reduced to the problem of verifying the satisfiability of KB $\cup$ $\{\neg\alpha\}$[2], which is EXPTIME-complete. It follows that subsumption and instance checking are CO-EXPTIME-complete, then they are EXPTIME-complete since CO-EXPTIME=EXPTIME (see Chapter 7 in [22]). □

3. A Tableaux calculus for $\mathcal{ALC}+\mathbf{T}$

In this section we present a tableau calculus for deciding the satisfiability of a knowledge base. Given a KB (TBox,ABox), any concrete reasoning system should provide the usual reasoning services, namely, *satisfiability of the KB*, *concept satisfiability*, *subsumption*, and *instance checking*. It is well known that the latter three services are reducible to the satisfiability of a KB.

We introduce a labelled tableau calculus for our logic $\mathcal{ALC}+\mathbf{T}$, which enriches the labelled tableau calculus for $\mathcal{ALC}$ presented in [7]. The calculus is called $T^{\mathcal{ALC}+\mathbf{T}}$ and it is based on the notion of *constraint system*. We consider a set of *variables* drawn from a denumerable set $\mathcal{V}$. $T^{\mathcal{ALC}+\mathbf{T}}$ makes use of *labels*, which are denoted with $x, y, z, \ldots$. Labels represent either a variable or an individual of the ABox, that is to say an element of $\mathcal{O}\cup\mathcal{V}$.

A *constraint* is a syntactic entity of the form $x \xrightarrow{R} y$ or $y < x$ or $x : C$, where R is a role and C is either an extended concept or has the form $\Box\neg D$ or $\neg\Box\neg D$, where D is a concept. The ABox of an $\mathcal{ALC}+\mathbf{T}$-knowledge base can be translated into a set of constraints by replacing every membership assertion $C(a)$ with the constraint $a : C$ and every role aRb with the constraint $a \xrightarrow{R} b$ (Definition 3.1). A tableau is a tree whose nodes are pairs $\langle S \mid U\rangle$, where:

- S contains constraints (or *labelled* formulas) of the form $x : C$ or $x \xrightarrow{R} y$ or $y < x$;
- U contains formulas of the form $C \sqsubseteq D^L$, representing subsumption relations $C \sqsubseteq D$ of the TBox. L is a list of labels. As we will discuss later, this list is used in order to ensure the termination of the tableau calculus.

A node $\langle S \mid U\rangle$ is also called a *constraint system*. A branch is a sequence of nodes $\langle S_1 \mid U_1\rangle, \langle S_2 \mid U_2\rangle, \ldots, \langle S_n \mid U_n\rangle \ldots$, where each node $\langle S_i \mid U_i\rangle$ is obtained by its immediate predecessor $\langle S_{i-1} \mid U_{i-1}\rangle$ by applying a rule of $T^{\mathcal{ALC}+\mathbf{T}}$, having $\langle S_{i-1} \mid U_{i-1}\rangle$ as the premise and $\langle S_i \mid U_i\rangle$ as one of its conclusions. A branch is closed if one of its nodes is an instance of (Clash), otherwise it is open. We say that a tableau is closed if all its branches are closed.

Given a KB, we define its *corresponding constraint system* as follows:

[2] In case α has the form $C \sqsubseteq D$, for $\neg\alpha$ we mean $(C \sqcap \neg D)(b)$, given b not appearing in the KB.

Definition 3.1. (Corresponding constraint system)
Given an $\mathcal{ALC} + \mathbf{T}$-knowledge base (TBox,ABox), we define its *corresponding constraint system* $\langle S \mid U \rangle$ as follows: $S = \{a : C \mid C(a) \in ABox\} \cup \{a \xrightarrow{R} b \mid aRb \in ABox\}$ and $U = \{C \sqsubseteq D^{\emptyset} \mid C \sqsubseteq D \in TBox\}$.

Definition 3.2. (Model satisfying a constraint system)
Let $\mathcal{M} = \langle \Delta, <_{\mathcal{M}}, I \rangle$ be a model as defined in Definition 2.4. We define a function α which assigns to each variable of $\mathcal{V}$ an element of Δ and assigns every individual $a \in \mathcal{O}$ to $a^I \in \Delta$. $\mathcal{M}$ satisfies (i) $x : C$ under α if $\alpha(x) \in C^I$, (ii) $x \xrightarrow{R} y$ under α if $(\alpha(x), \alpha(y)) \in R^I$ and (iii) $y < x$ under α if $\alpha(y) <_{\mathcal{M}} \alpha(x)$. A constraint system $\langle S \mid U \rangle$ is satisfiable if there is a model $\mathcal{M}$ and a function α such that $\mathcal{M}$ satisfies under α every constraint in S and that, for all $C \sqsubseteq D \in U$ and for all x occurring in S, we have that if $\alpha(x) \in C^I$, then $\alpha(x) \in D^I$.

It can be easily shown that:

Proposition 3.1. Given an $\mathcal{ALC} + \mathbf{T}$-knowledge base, it is satisfiable if and only if its corresponding constraint system is satisfiable.

Therefore, in order to check the satisfiability of (TBox,ABox), we build its corresponding constraint system $\langle S \mid U \rangle$ and then we use $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$ to check the satisfiability of $\langle S \mid U \rangle$. In order to check a constraint system $\langle S \mid U \rangle$ for satisfiability, our calculus $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$ adopts the usual technique of applying the rules until either a contradiction is generated (Clash) or a model satisfying $\langle S \mid U \rangle$ can be obtained from the resulting constraint system.

In order to take into account the TBox, we use a technique of *unfolding*, similar to the one described in [7]. Given a node $\langle S \mid U \rangle$, for each subsumption $C \sqsubseteq D^L \in U$ and for each label x that appears in S, we add to S the constraint $x : \neg C \sqcup D$. As mentioned above, each formula $C \sqsubseteq D$ is equipped by the list L of labels in which it has been unfolded in the current branch. This is needed in order to avoid multiple unfolding of the same subsumption by using the same label, generating non-termination in a proof search.

Before introducing the rules of $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$ we need some more definitions. First, as in [7], we assume that labels are introduced in a tableau according to an ordering $\prec$, that is to say if y is introduced in the tableau, then $x \prec y$ for all labels x that are already in the tableau.

Given a tableau node $\langle S \mid U \rangle$ and a label x, we define $\sigma(\langle S \mid U \rangle, x) = \{C \mid x : C \in S\}$. Furthermore, we say that two labels x and y are *S-equivalent*, written $x \equiv_S y$, if they label the same set of concepts, i.e., $\sigma(\langle S \mid U \rangle, x) = \sigma(\langle S \mid U \rangle, y)$. Intuitively, S-equivalent labels can represent the same element in the model built by the rules of $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$. Last, we define $S^M_{x \to y} = \{y : \neg C, y : \Box\neg C \mid x : \Box\neg C \in S\}$. $S^M_{x \to y}$ is used in order to propagate both the argument $\neg C$ of a boxed formula $\Box\neg C$ and the boxed formula $\Box\neg C$ itself, due to transitivity.

The rules of $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$ are presented in Figure 1. Rules $(\exists^+)$, $(\forall^-)$, and $(\Box^-)$ are called *dynamic* since they introduce a new variable in their conclusions. The other rules are called *static*. We do not need any extra rule for the positive occurrences of the $\Box$ operator, since these are taken into account by the computation of $S^M_{x \to y}$. Intuitively, when the rule $(\Box^-)$ is applied to $x : \neg\Box\neg D$ generating a new label y such that $y < x$, then, for each $x : \Box\neg C$ in S, the rule propagates the boxed formula by adding (i) its argument $y : \neg C$ and (ii) $y : \Box\neg C$ since $<$ is transitive.

$$\langle S, x : C, x : \neg C \mid U \rangle \quad \text{(Clash)} \qquad\qquad \frac{\langle S, x : \neg\neg C \mid U \rangle}{\langle S, x : C \mid U \rangle}\,(\neg)$$

$$\frac{\langle S, x : \mathbf{T}(C) \mid U \rangle}{\langle S, x : C, x : \Box\neg C \mid U \rangle}\,(\mathbf{T}^+) \qquad\qquad \frac{\langle S, x : \neg\mathbf{T}(C) \mid U \rangle}{\langle S, x : \neg C \mid U \rangle \qquad \langle S, x : \neg\Box\neg C \mid U \rangle}\,(\mathbf{T}^-)$$

$$\frac{\langle S, x : \forall R.C, x \xrightarrow{R} y \mid U \rangle}{\langle S, x : \forall R.C, x \xrightarrow{R} y, y : C \mid U \rangle}\,(\forall^+) \ \text{if } y : C \notin S$$

$$\frac{\langle S, x : \exists R.C \mid U \rangle}{\langle S, x : \exists R.C, x \xrightarrow{R} y, y : C \mid U \rangle}\,(\exists^+) \ \begin{array}{r} y \text{ new} \\ \text{if } \nexists z \prec x \text{ s.t. } z \equiv_{S,x:\exists R.C} x \text{ and} \\ \nexists u \text{ s.t. } x \xrightarrow{R} u \in S \text{ and } u : C \in S \end{array}$$

$$\frac{\langle S, x : \neg\forall R.C \mid U \rangle}{\langle S, x : \neg\forall R.C, x \xrightarrow{R} y, y : \neg C \mid U \rangle}\,(\forall^-) \ \begin{array}{r} y \text{ new} \\ \text{if } \nexists z \prec x \text{ s.t. } z \equiv_{S,x:\neg\forall R.C} x \text{ and} \\ \nexists u \text{ s.t. } x \xrightarrow{R} u \in S \text{ and } u : \neg C \in S \end{array}$$

$$\frac{\langle S, x : \neg\exists R.C, x \xrightarrow{R} y \mid U \rangle}{\langle S, x : \neg\exists R.C, x \xrightarrow{R} y, y : \neg C \mid U \rangle}\,(\exists^-) \ \text{if } y : \neg C \notin S$$

$$\frac{\langle S, x : \neg\Box\neg C \mid U \rangle}{\langle S, x : \neg\Box\neg C, y < x, y : C, y : \Box\neg C, S^M_{x \rightarrow y} \mid U \rangle}\,(\Box^-) \ \begin{array}{r} y \text{ new} \\ \text{if } \nexists z \prec x \text{ s.t. } z \equiv_{S,x:\neg\Box\neg C} x \text{ and} \\ \nexists u \text{ s.t. } \{u < x, u : C, u : \Box\neg C, S^M_{x \rightarrow u}\} \subseteq S \end{array}$$

$$\frac{\langle S \mid U, C \sqsubseteq D^L \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \rangle}\,\text{(Unfold)} \ \text{if } x \text{ occurs in } S \text{ and } x \notin L$$

Figure 1. The calculus $T^{\mathcal{ALC}+\mathbf{T}}$. To save space, we omit the standard rules for $\sqcup$ and $\sqcap$.

The side conditions on the rules $(\exists^+)$, $(\forall^-)$, and $(\Box^-)$ are introduced in order to ensure a terminating proof search, by implementing the standard *blocking* technique described below. The rules of $T^{\mathcal{ALC}+\mathbf{T}}$ are applied with the following *standard strategy*: 1. apply a rule to a label x only if no rule is applicable to a label y such that $y \prec x$; 2. apply dynamic rules only if no static rule is applicable. This strategy ensures that the labels are considered one at a time according to the ordering $\prec$. Consider an application of a dynamic rule to a label x of a constraint system $\langle S \mid U \rangle$. For all $\langle S' \mid U' \rangle$ obtained from $\langle S \mid U \rangle$ by a sequence of rule applications, it can be easily shown that (i) no rule can be applied in $\langle S' \mid U' \rangle$ to a label y s.t. $y \prec x$ and (ii) $\sigma(\langle S \mid U \rangle, x) = \sigma(\langle S' \mid U' \rangle, x)$. The calculus so obtained is sound and complete with respect to the semantics described in Definition 3.2. In order to prove this, we first introduce the notion of *witness*:

Definition 3.3. (Witness)
Given a constraint system $\langle S \mid U \rangle$ and two labels x and y occurring in S, we say that x is a witness of y if the following conditions hold: 1. $x \equiv_S y$; 2. $x \prec y$; 3. there is no label z s.t. $z \prec x$ and z satisfies conditions 1. and 2., i.e., x is the least label satisfying conditions 1. and 2. w.r.t. $\prec$. We say that y is *blocked* by x in $\langle S \mid U \rangle$ if y has witness x.

By the strategy on the application of the rules described above and by Definition 3.3, we can prove the following Lemma:

Lemma 3.1. In any constraint system $\langle S \mid U \rangle$, if x is blocked, then it has exactly one witness.

Since all the rules are invertible, we can assume that there is only one single tableau (there can be several that only differ as far as the labels' names are concerned). In order to prove the completeness of the calculus, we need to introduce the notion of saturated branch. Informally speaking, this is a branch in which all the rules of the calculus have been applied as much as possible. More formally:

Definition 3.4. (Saturated Branch)
A branch **B**=$\langle S_0 \mid U_0\rangle, \langle S_1 \mid U_1\rangle, \ldots, \langle S_i \mid U_i\rangle, \ldots$ is *saturated* if the following conditions hold: 1. for all $C \sqsubseteq D^L$ and for all labels x occurring in **B**, either $x : \neg C$ or $x : D$ belong to **B**; 2. if $x : \mathbf{T}(C)$ occurs in **B**, then $x : C$ and $x : \Box\neg C$ occur in **B**; 3. if $x : \neg\mathbf{T}(C)$ occurs in **B**, then either $x : \neg C$ or $x : \neg\Box\neg C$ occurs in **B**; 4. if $x : \neg\Box\neg C$ occurs in **B**, then either there is y such that $y < x$, $y : C$, $y : \Box\neg C$, and $S^M_{x\rightarrow y}$ occur in **B** or x is blocked by a witness w, and there is y such that $y < w$, $y : C$, $y : \Box\neg C$, and $S^M_{w\rightarrow y}$ occur in **B**; 5. if $x : \exists R.C$ occurs in **B**, then either there is y such that $x \xrightarrow{R} y$ and $y : C$ occur in **B** or x is blocked by a witness w, and $w \xrightarrow{R} y$ and $y : C$ occur in **B**; 6. if $x : \forall R.C$ and $x \xrightarrow{R} y$ occur in **B**, also $y : C$ occurs in **B**; 7. for $x : \neg\forall R.C$ and for $x : \neg\exists R.C$ the condition of saturation is defined symmetrically; 8. for the boolean rules the condition of saturation is defined in the usual way. For instance, if $x : C \sqcap D$ occurs in **B**, also $x : C$ and $x : D$.

By following the strategy on the order of application of the rules outlined above and by Lemma 3.1, the following Proposition holds:

Proposition 3.2. Any open branch can be expanded into an open *saturated* branch.

In order to show the completeness of $T^{\mathcal{ALC}+\mathbf{T}}$, given an open, saturated branch **B**, we explicitly add to **B** the relation $y < x$ if x is blocked and w is the witness of x and $y < w$ occurs in **B**. Before proving the completeness, we prove the following Lemma:

Lemma 3.2. In the tableau, there is no open branch **B** containing an infinite descending chain $\ldots x_2 < x_1 < x_0$.

Proof:
The only way to obtain an infinite descending chain $\ldots x_2 < x_1 < x_0$ would be to have either (i) a loop or (ii) an infinite set of distinct labels. We can show that neither (i) nor (ii) can occur. As far as (i) is concerned, suppose for absurd that there were a loop, that is to say there is an infinite descending chain $x < u < \cdots < y_i < \cdots < y < x$. The relation $x < u$ cannot have been inserted in the branch by the rule $(\Box^-)$, that can only introduce in the branch relations $x < u$ where x is a new label. Hence, $x < u$ must have been introduced because u is blocked by some witness w, and $x < w$ occurs in **B**. Notice, however, that in this case: 1. $x < w$ has been introduced by $(\Box^-)$ applied to some $w : \neg\Box\neg C$, hence, $x : \Box\neg C$ occurs in **B**; 2. it can be easily proved that also for all y_i and for u, we have that $y_i : \Box\neg C$ and $u : \Box\neg C$ belong to **B**; 3. since w is a witness of u, also $u : \neg\Box\neg C$ occurs in the branch **B**, which contradicts the hypothesis that **B** was open. Concerning (ii), suppose there were an infinite descending chain $< \cdots < x_i \cdots < x_0$. Each relation must be generated by a $\neg\Box\neg C$ that has not yet been used in the chain, either by an application of the rule $(\Box^-)$ to $\neg\Box\neg C$ in x_{i-1}, or by an application of the rule $(\Box^-)$ to $\neg\Box\neg C$ in the witness w of x_{i-1}. Indeed, if $\neg\Box\neg C$ had been previously used in the chain, say in introducing $x_i < x_{i-1}$, for each x_j such that $x_j < \cdots < x_i$, $x_j : \Box\neg C$ is in **B**, hence $x_j : \neg\Box\neg C$ cannot be in **B**, otherwise **B** would be closed, against the hypothesis. Notice however that the only formulas $\neg\Box\neg C$ that appear in the branch are derived from $\mathbf{T}(C)$ appearing in the initial constraint system. Since the number of such $\mathbf{T}(C)$ is finite, it follows that also the number of possible different $\neg\Box\neg C$ is finite, and the infinite descending chain cannot be generated.

□

With the above propositions at hand, we can show that:

Theorem 3.1. (Soundness and Completeness of $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$)
Given a constraint system $\langle S \mid U \rangle$, it is unsatisfiable iff it has a closed tableau.

Proof:
As usual, the soundness can be proven by induction on the height of the closed tableau for $\langle S \mid U \rangle$. By considering each rule of $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$, it can be easily shown that if the premise is satisfiable, so is (at least) one of its conclusion. The theorem follows by contraposition.

Concerning the completeness, we show the contrapositive, i.e., that if the tableau is open, then the starting constraint system is satisfiable. An open tableau contains an open branch that by Proposition 3.2 can be expanded into an open *saturated* branch. From such a branch, call it **B**, we define a canonical model $\mathcal{M} = \langle \Delta_B, <', I \rangle$ where: $\Delta_B = \{x : x$ is a label appearing in $\mathbf{B}\}$; $<'$ is the transitive closure of relation $<$ in **B**; I is an interpretation function s.t. for all atomic concepts A, $A^I = \{x$ such that $x : A$ occurs in $\mathbf{B}\}$. I is then extended to all concepts C in the standard way, according to the semantics of the operators. For role names R, $R^I = \{(x, y)$: either $x \xrightarrow{R} y$ occurs in **B** or x is blocked and w is the witness of x (by Lemma 3.1 such w exists) and $w \xrightarrow{R} y$ occurs in $\mathbf{B}\}$. We can show that:

- *$<'$ is irreflexive, transitive, and satisfies the Smoothness Condition.* Irreflexivity follows from the fact that the relation $<$ is either introduced by rule $(\Box^-)$ between a label already present in **B** and a new label or is explicitly added in case some $x : \neg\Box\neg C$ is on the branch and x is blocked. In this case, suppose by absurd that $x < x$ is added, this means that x is blocked by w, thus $w : \neg\Box\neg C$ belongs to **B**, as well as $x < w, x : C, x : \Box\neg C$ belong to **B**, but this contradicts the fact that **B** is open (both $x : \neg\Box\neg C$ and $x : \Box\neg C$ occur). Transitivity follows from definition of $<'$. The Smoothness Condition follows from transitivity of $<'$ together with the finiteness of chains of $<$ deriving from Lemma 3.2.

- *for all concepts C we have:* (a) *if $x : C$ occurs in* **B**, *then $x \in C^I$;* (b) *if $x : \neg C$ occurs in* **B**, *then $x \in (\neg C)^I$.* We reason by induction on the complexity of C. If C is a boolean combination of concepts, the proof is simple and left to the reader.

  If C is $\exists R.D$, then by saturation, either $x \xrightarrow{R} y, y : C$ occur in **B** or $w \xrightarrow{R} y, y : C$ occur in **B**, for w witness of x. In both cases, $(x, y) \in R^I$ by construction, and by inductive hypothesis $y \in C^I$, hence (a) follows. (b) can be proven similarly to the following case (a).

  If C is $\forall R.D$, then by saturation, for all y s.t. $x \xrightarrow{R} y$ occurs in **B**, also $y : D$ occurs in **B**, and the (a) holds by the inductive hypothesis. (b) can be proven similarly to the previous case (a).

  If C is $\Box\neg D$ and $x : \Box\neg D$ occurs in **B**, let $x_i <' x$, then there is $x_i < y < \cdots < u < x$, where each relation has been introduced either by $(\Box^-)$ or by the completion described before Lemma 3.2. Observe that, in both cases, $u : \neg D$ and $u : \Box\neg D$ are added to **B**, since they belong to the conclusion of the first application of $(\Box^-)$ generating u. For the same reason, further applications of $(\Box^-)$ introduce $y : \neg D$ and $y : \Box\neg D$. Therefore, also $x_i : \neg D$ occurs in **B**. By inductive hypothesis, $x_i \in (\neg D)^I$, hence $x \in (\Box\neg D)^I$. If $x : \neg\Box\neg D$, then by saturation there is y s.t. $y < x, y : D$ and $y : \Box\neg D$ occur in **B**. By definition of $<'$, we have that $y <' x$ and, by inductive hypothesis, $y \in D^I$ and $y \in (\Box\neg D)^I$. It follows that $x \in (\neg\Box\neg D)^I$.

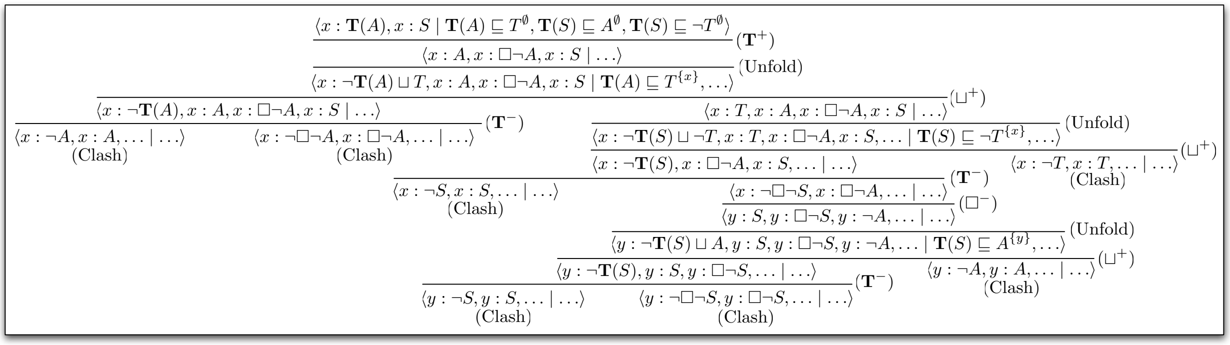
$\langle x : \mathbf{T}(A), x : S \mid \mathbf{T}(A) \sqsubseteq T^{\emptyset}, \mathbf{T}(S) \sqsubseteq A^{\emptyset}, \mathbf{T}(S) \sqsubseteq \neg T^{\emptyset}\rangle$
$(\mathbf{T}^{+})$
$\langle x : A, x : \Box\neg A, x : S \mid \ldots\rangle$
(Unfold)
$\langle x : \neg\mathbf{T}(A) \sqcup T, x : A, x : \Box\neg A, x : S \mid \mathbf{T}(A) \sqsubseteq T^{\{x\}}, \ldots\rangle$
$(\sqcup^{+})$
$\langle x : \neg\mathbf{T}(A), x : A, x : \Box\neg A, x : S \mid \ldots\rangle$
$(\mathbf{T}^{-})$
$\langle x : \neg A, x : A, \ldots \mid \ldots\rangle$
(Clash)
$\langle x : \neg\Box\neg A, x : \Box\neg A, \ldots \mid \ldots\rangle$
(Clash)
$\langle x : T, x : A, x : \Box\neg A, x : S \mid \ldots\rangle$
(Unfold)
$\langle x : \neg\mathbf{T}(S) \sqcup \neg T, x : T, x : \Box\neg A, x : S, \ldots \mid \mathbf{T}(S) \sqsubseteq \neg T^{\{x\}}, \ldots\rangle$
$(\sqcup^{+})$
$\langle x : \neg\mathbf{T}(S), x : \Box\neg A, x : S, \ldots \mid \ldots\rangle$
$\langle x : \neg T, x : T, \ldots \mid \ldots\rangle$
(Clash)
$(\mathbf{T}^{-})$
$\langle x : \neg S, x : S, \ldots \mid \ldots\rangle$
(Clash)
$\langle x : \neg\Box\neg S, x : \Box\neg A, \ldots \mid \ldots\rangle$
$(\Box^{-})$
$\langle y : S, y : \Box\neg S, y : \neg A, \ldots \mid \ldots\rangle$
(Unfold)
$\langle y : \neg\mathbf{T}(S) \sqcup A, y : S, y : \Box\neg S, y : \neg A, \ldots \mid \mathbf{T}(S) \sqsubseteq A^{\{y\}}, \ldots\rangle$
$(\sqcup^{+})$
$\langle y : \neg\mathbf{T}(S), y : S, y : \Box\neg S, \ldots \mid \ldots\rangle$
$\langle y : \neg A, y : A, \ldots \mid \ldots\rangle$
(Clash)
$(\mathbf{T}^{-})$
$\langle y : \neg S, y : S, \ldots \mid \ldots\rangle$
(Clash)
$\langle y : \neg\Box\neg S, y : \Box\neg S, \ldots \mid \ldots\rangle$
(Clash)

Figure 2. A closed tableau showing that $\mathbf{T}(\mathit{Adult}) \sqsubseteq \neg\mathit{Student}$ can be inferred from the TBox $\{\mathbf{T}(\mathit{Adult}) \sqsubseteq \mathit{TaxPayer}, \mathbf{T}(\mathit{Student}) \sqsubseteq \mathit{Adult}, \mathbf{T}(\mathit{Student}) \sqsubseteq \neg\,\mathit{TaxPayer}\}$. To save space, we use A for Adult, T for $\mathit{TaxPayer}$, and S for $\mathit{Student}$.

If C is $\mathbf{T}(D)$ and $x : \mathbf{T}(D)$ occurs in **B**, by saturation, both $x : D$ and $x : \Box\neg D$ occur in **B**, hence by inductive hypothesis $x \in D^{I}$ and $x \in (\Box\neg D)^{I}$, and by Proposition 2.1, $x \in (\mathbf{T}(D))^{I}$. If $x : \neg\mathbf{T}(D)$ occurs in **B**, then by saturation also $x : \neg D$ occurs in **B** or $x : \neg\Box\neg D$ occurs in **B**. By inductive hypothesis either $x \notin D^{I}$ or $x \notin (\Box\neg D)^{I}$. In both cases, by Proposition 2.1, we conclude that $x \in (\neg\mathbf{T}(D))^{I}$.

- for all $C \sqsubseteq D \in U$ and all labels x, either $x \in (\neg C)^{I}$ or $x \in D^{I}$, i.e., $C^{I} \subseteq D^{I}$. By saturation, either $x : \neg C$ occurs in **B** or $x : D$ occurs in **B**. The property follows by inductive hypothesis.
- $\mathcal{M}$ satisfies the starting constraint system: this is an immediate consequence of the previous points.

□

As an example, Figure 2 shows a derivation in $T^{\mathcal{ALC}+\mathbf{T}}$ of the fact that $\mathbf{T}(\mathit{Adult}) \sqsubseteq \neg\mathit{Student}$ can be inferred from the TBox $\{\mathbf{T}(\mathit{Adult}) \sqsubseteq \mathit{TaxPayer}, \mathbf{T}(\mathit{Student}) \sqsubseteq \mathit{Adult}, \mathbf{T}(\mathit{Student}) \sqsubseteq \neg\,\mathit{TaxPayer}\}$. In order to do so, $T^{\mathcal{ALC}+\mathbf{T}}$ checks whether the KB whose TBox is the one above and whose ABox contains an individual x which is both a typical Adult and a $\mathit{Student}$ (i.e., ABox=$\{\mathbf{T}(\mathit{Adult})(x), \mathit{Student}(x)\}$) is unsatisfiable. In detail, $T^{\mathcal{ALC}+\mathbf{T}}$ tries to build a closed tableau for the constraint system corresponding to the KB, namely, $\langle x : \mathbf{T}(\mathit{Adult}), x : \mathit{Student} \mid \mathbf{T}(\mathit{Adult}) \sqsubseteq \mathit{TaxPayer}^{\emptyset}, \mathbf{T}(\mathit{Student}) \sqsubseteq \mathit{Adult}^{\emptyset}, \mathbf{T}(\mathit{Student}) \sqsubseteq \neg\,\mathit{TaxPayer}^{\emptyset}\rangle$.

Let us conclude this section by analyzing termination and complexity of $T^{\mathcal{ALC}+\mathbf{T}}$. In general, non-termination in labelled tableau calculi can be caused by two different reasons: 1. some rules copy their principal formula in the conclusion(s), and can thus be reapplied over the same formula without any control; 2. dynamic rules may generate infinitely-many labels, creating infinite branches.

Concerning the first source of non-termination (point 1), the only rules copying their principal formulas in their conclusions are $(\forall^{+})$, $(\exists^{-})$, (Unfold), $(\forall^{-})$, $(\exists^{+})$, and $(\Box^{-})$. However, the side conditions on these rules avoid multiple applications on the same formula. Indeed, (Unfold) can be applied to a constraint system $\langle S \mid U, C \sqsubseteq D^{L}\rangle$ by using the label x only if it has not yet been applied to x in the current branch (i.e., x does not belong to L). Concerning $(\forall^{+})$, the rule can be applied to $\langle S, x : \forall R.C, x \xrightarrow{R} y \mid U\rangle$ only if $y : C$ does not belong to S. When $y : C$ is introduced in the branch, the rule will not further apply to $x : \forall R.C$. Similarly for $(\exists^{-})$, $(\exists^{+})$, $(\forall^{-})$, and $(\Box^{-})$.

Concerning the second source of non-termination (point 2), we can prove that we only need to adopt the standard loop-checking machinery known as *blocking*, which ensures that the rules $(\exists^{+})$, $(\forall^{-})$, and

$(\Box^-)$ do not introduce infinitely-many labels on a branch. Thanks to the properties of $\Box$, no other additional machinery would be required to ensure termination. Indeed, we can show that the interplay between rules $(\mathbf{T}^-)$ and $(\Box^-)$ does not generate branches containing infinitely-many labels.

Let us discuss the termination in more detail. Without the side conditions on the rules $(\exists^+)$ and $(\forall^-)$, the calculus $T^{\mathcal{ALC}+\mathbf{T}}$ does not ensure a terminating proof search. Indeed, given a constraint system $\langle S \mid U\rangle$, it could be the case that $(\exists^+)$ is applied to a constraint $x : \exists R.C \in S$, introducing a new label y and the constraints $x \xrightarrow{R} y$ and $y : C$. If an inclusion $\mathbf{T}(\exists R.C) \sqsubseteq D$ belongs to U, then (Unfold) can be applied by using y, thus generating a branch containing $y : \neg\mathbf{T}(\exists R.C)$, to which $(\mathbf{T}^-)$ can be applied introducing $y : \neg\Box\neg(\exists R.C)$. An application of $(\Box^-)$ introduces a new variable z and the constraint $z : \exists R.C$, to which $(\exists^+)$ can be applied generating a new label u. (Unfold) can then be re-applied on $\mathbf{T}(\exists R.C) \sqsubseteq D$ by using u, incurring a loop. In order to prevent this source of non termination, we adopt the standard technique of *blocking*: the side condition of the $(\exists^+)$ rule says that this rule can be applied to a node $\langle S, x : \exists R.C \mid U\rangle$ only if x is not blocked. In other words, if there is a witness z of x, then $(\exists^+)$ is not applicable, since the condition and the strategy imply that the $(\exists^+)$ rule has already been applied to z. The same for $(\forall^-)$ and $(\Box^-)$.

As mentioned, another possible source of infinite branches could be determined by the interplay between rules $(\mathbf{T}^-)$ and $(\Box^-)$. However, even if we had no blocking on $(\Box^-)$ this could not occur, i.e., the interplay between these two rules does not generate branches containing infinitely-many labels. Intuitively, the application of $(\Box^-)$ to $x : \neg\Box\neg C$ adds $y : \Box\neg C$ to the conclusion, so that $(\mathbf{T}^-)$ can no longer consistently introduce $y : \neg\Box\neg C$. This is due to the properties of $\Box$ (no infinite descending chains of $<$ are allowed). More in detail, if (Unfold) is applied to $\mathbf{T}(C) \sqsubseteq D$ by using x, an application of $(\mathbf{T}^-)$ introduces a branch containing $x : \neg\Box\neg C$; when a new label y is generated by an application of $(\Box^-)$ on $x : \neg\Box\neg C$, we have that $y : \Box\neg C$ is added to the current constraint system. If (Unfold) and $(\mathbf{T}^-)$ are also applied to $\mathbf{T}(C) \sqsubseteq D$ on the new label y, then the conclusion where $y : \neg\Box\neg C$ is introduced is closed, by the presence of $y : \Box\neg C$. By this fact, we would not need to introduce any loop-checking machinery on the application of $(\Box^-)$. A detailed proof of termination of the calculus without blocking on $(\Box^-)$ can be found in [15]. However, in this paper we have introduced blocking also on $(\Box^-)$ for complexity reasons.

Theorem 3.2. (Termination of $T^{\mathcal{ALC}+\mathbf{T}}$)
Let $\langle S \mid U\rangle$ be a constraint system, then any tableau generated by $T^{\mathcal{ALC}+\mathbf{T}}$ is finite.

Our calculus $T^{\mathcal{ALC}+\mathbf{T}}$ gives a (suboptimal) nondeterministic-exponential time decision procedure for $\mathcal{ALC} + \mathbf{T}$:

Theorem 3.3. (Complexity of $T^{\mathcal{ALC}+\mathbf{T}}$)
Given an $\mathcal{ALC} + \mathbf{T}$-knowledge base (TBox,ABox), checking whether it is satisfiable by using $T^{\mathcal{ALC}+\mathbf{T}}$ is in NEXPTIME.

Proof:
We first show that the number of labels generated on a branch is at most exponential in the size of KB. Let n be the size of a KB. Given a constraint system $\langle S \mid U\rangle$, the number of extended concepts appearing in $\langle S \mid U\rangle$, including also all the ones appearing as a subformula of other concepts, is $O(n)$. As there are at most $O(n)$ concepts, there are at most $O(2^n)$ variables labelling distinct sets of concepts. Hence, there are $O(2^n)$ non-blocked variables in S.

Let m be the maximum number of direct successors of each variable $x \in S$, obtained by applying dynamic rules. m is bound by the number of $\exists R.C$ concepts ($O(n)$) plus the number of $\neg\forall R.C$ concepts ($O(n)$) plus the number of $\neg\Box\neg C$ concepts ($O(n)$). Then, there are at most $O(2^n \times m)$ *variables* in S, where $m \leq 3n$. The number of *individuals* in ABox is bound by n too, and each individual has at most m direct successors. The number of *labels* in S is then bound by $O((2^n + n) \times m)$, hence by $O(2^{2n})$.

For a given label x, the concepts labelled by x introduced in the branch (namely, all the possible subconcepts of the initial constraint system, as well as all boxed subconcepts) are $O(n)$. According to the standard strategy, after all static rules have been applied to a label x in phase 1, no other concepts labelled by x can be introduced later on a branch. Hence, the labelled concepts introduced on the branch is $O(n)$ for each label, and the number of all labelled concepts on the branch is $O(n \times 2^{2n})$. Therefore, a branch can contain at most an exponential number of applications of tableau rules.

The satisfiability of a KB can thus be solved by defining a procedure which nondeterministically generates an open branch of exponential size (in the size of KB). The problem is in NEXPTIME.

□

4. Extensions of $\mathcal{ALC} + \mathbf{T}$ for Reasoning about Typicality

Logic $\mathcal{ALC} + \mathbf{T}$ allows one to reason monotonically about typicality. In $\mathcal{ALC} + \mathbf{T}$ we can consistently express, for instance, the fact that *italian fencers* typically are not people's favorite, that *italian fencer olympic champions* typically are people's favorite, and that *italian fencer olympic champions taking part to a reality show* typically are not people's favorite.

What about the typical properties of an individual $aldo$ that we know being an italian fencer, who won the gold medal at an olympic competition and who has been part of a reality show? Of course, if we know that $aldo$ is a typical instance of the concept $ItalianFencer \sqcap OlympicGoldMedalist \sqcap \exists TakePart.RealityShow$, i.e., if the ABox contains the assertion

$$(*)\ \mathbf{T}(ItalianFencer \sqcap OlympicGoldMedalist \sqcap \exists TakePart.RealityShow)(aldo)$$

then, in $\mathcal{ALC} + \mathbf{T}$, we can conclude that $\neg LovedByPeople(aldo)$. However, in absence of (*), $\neg LovedByPeople(aldo)$ cannot be derived by the logic itself given the nonmonotonic nature of $\mathbf{T}$.

The basic monotonic logic $\mathcal{ALC} + \mathbf{T}$ is then too weak to enforce these extra assumptions, so that we need an additional mechanism to perform defeasible inferences.

We propose two alternatives. The first one, introduced in [13], is based on a nonmonotonic completion of a knowledge base. The second one, presented in [12], consists of a "minimal model" semantics for $\mathcal{ALC} + \mathbf{T}$ whose intuition is that minimal models are those that maximise typical instances of concepts.

4.1. Completion of a knowledge base

In general, we would like to infer that individuals have the properties which are typical of the most specific concept to which they belong. To this purpose, we define a completion of the knowledge base which adds to the ABox, for each individual a occurring in the ABox, the assertion that a is a typical instance of the most specific concept C to which it belongs.

Definition 4.1. (Completion of a Knowledge Base)
The KB (TBox,ABox') is the completion of the KB (TBox,ABox) if ABox' is obtained from ABox by adding to it, for all individual names a in the ABox, the assertion $\mathbf{T}(C_1 \sqcap \ldots \sqcap C_j)(a)$, where $C_1, \ldots, C_j$ are all the concepts C_i such that: (1) C_i is a subconcept of any concept occurring in (TBox,ABox); (2) C_i does not contain $\mathbf{T}$; (3) a is an instance of C_i, i.e., $C_i(a)$ is derivable in $\mathcal{ALC}$ from (TBox,ABox).

For instance, assuming that:

$$\begin{array}{l} \mathit{ItalianFencer}(\mathit{aldo}) \\ \mathit{OlympicGoldMedalist}(\mathit{aldo}) \\ \exists \mathit{TakePart}.\mathit{RealityShow}(\mathit{aldo}) \end{array}$$

are the only assertions concerning aldo derivable from the KB, the completion above would add:

$$\mathbf{T}(\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \exists \mathit{TakePart}.\mathit{RealityShow})(\mathit{aldo})$$

to the ABox, as $\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \exists \mathit{TakePart}.\mathit{RealityShow}$ is the most specific concept of which aldo is an instance. From this, we can conclude in $\mathcal{ALC} + \mathbf{T}$ that aldo is not a people's favourite.

The completion adds $\mathbf{T}(C_1 \sqcap \ldots \sqcap C_j)(a)$ by considering each $C_i(a)$ *derivable* in $\mathcal{ALC}$ from the KB, rather than considering only $C_i(a)$ in the ABox. This is needed, for instance, to infer that aldo is not loved by people from the KB containing $\mathit{ItalianFencer} \sqsubseteq \forall \mathit{HasChild}.\mathit{ItalianFencer}$, $\mathit{ItalianFencer}(\mathit{mario})$, and $\mathit{HasChild}(\mathit{mario}, \mathit{aldo})$.

Notice that the completion of the ABox only introduces $O(n)$ new assertions $\mathbf{T}(C_1 \sqcap \ldots \sqcap C_j)(a)$, one for each named individual a in the ABox. Furthermore, the size of the assertion $\mathbf{T}(C_1 \sqcap \ldots \sqcap C_j)(a)$ is $O(n^2)$ as $C_1, \ldots, C_j$ are all distinct subformulas of the initial formula ($O(n)$), and each C_i has size $O(n)$. Hence, after the completion construction, the size of the KB is polynomial in n. Moreover, for each individual a ($O(n)$) and for each concept C ($O(n)$), we have to check whether $C(a)$ is derivable in $\mathcal{ALC}$ from the KB, which is a problem in EXPTIME. Hence, the completion construction requires exponential time and produces a KB of size polynomial in the size of the original one:

Theorem 4.1. The problem of deciding satisfiability of the knowledge base after completion is EXPTIME-complete in the size of the original KB.

It is worth noticing that, given a consistent KB, its completion might be inconsistent. Suppose the ABox contains the information:

$$\mathit{LovedByPeople}(\mathit{aldo})$$

This would not cause an inconsistent completion of the KB. Indeed, in such a case, $\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \exists \mathit{TakePart}.\mathit{RealityShow} \sqcap \mathit{LovedByPeople}$ would be the most specific concept of which aldo is an instance, so that the assertion:

$$\mathbf{T}(\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \mathit{TakePart}.\mathit{RealityShow} \sqcap \mathit{LovedByPeople})(\mathit{aldo})$$

would be added in the completion of the KB. This does not allow to infer that $\neg \mathit{LovedByPeople}(\mathit{aldo})$. Hence, no inconsistency arises. However, if the KB contains:

$$\begin{array}{l}
\mathbf{T}(\mathit{FootballTeam}) \sqsubseteq \forall \mathit{HasMember}.\mathit{Rich} \\
\mathbf{T}(\mathit{FencingTeam}) \sqsubseteq \forall \mathit{HasMember}.\neg \mathit{Rich} \\
\mathit{FootballTeam}(\mathit{juventus}) \\
\mathit{FencingTeam}(\mathit{schermaTorino}) \\
\mathit{HasMember}(\mathit{juventus}, \mathit{alex}) \\
\mathit{HasMember}(\mathit{schermaTorino}, \mathit{alex})
\end{array}$$

we can observe that KB is consistent, whereas its completion, including also:

$$\begin{array}{l}
\mathbf{T}(\mathit{FootballTeam})(\mathit{juventus}) \\
\mathbf{T}(\mathit{FencingTeam})(\mathit{schermaTorino})
\end{array}$$

is not. In this case, we can consider two alternatives:

- given an inconsistent completion, we can choose to keep the original KB rather than the completed one;
- we could consider all maximal consistent KBs (extensions) that can be generated by adding, for all individuals, the relative most-specific concept assumptions. We could then perform either a skeptical or a credulous reasoning with respect to such extensions.

The completion process presents some difficulties. For instance, it is not clear how to take into account implicit individuals, as well as it is not clear whether and how the completion has to take into account concept instances that are inferred from previous typicality assumptions introduced by the completion itself. In order to deal with the last one, we would need some kind of fixpoint definition.

4.2. Minimal model semantics for $\mathcal{ALC} + \mathbf{T}$

As an alternative to the completion process described above, we propose another approach: rather than defining an ad-hoc mechanism to perform defeasible inferences or making nonmonotonic assumptions, we strengthen the semantics of the logic by proposing a minimal model semantics. Intuitively, the idea is to restrict our consideration to models that maximise typical instances of a concept.

In order to define the preference relation on models we take advantage of the modal semantics of $\mathcal{ALC} + \mathbf{T}$: the preference relation on models (with the same domain) is defined by comparing, for each individual, the set of modal (or more precisely $\Box$-ed) concepts containing the individual in the two models. Given a KB, we consider a finite set $\mathcal{L}_T$ of concepts occurring in the KB, the typicality of whose instances we want to maximize. This is similar to circumscription (see [4]), where we must specify a set of minimized predicates. The maximization of the set of typical instances will apply to individuals explicitly occurring in the ABox as well as to implicit individuals. We assume that the set $\mathcal{L}_T$ contains at least all concepts C such that $\mathbf{T}(C)$ occurs in the KB.

We have seen that a is a typical instance of a concept C ($a \in (\mathbf{T}(C))^I$) when it is an instance of C and there is not another instance of C preferred to a, i.e. $a \in (C \sqcap \Box\neg C)^I$. In the following, in order to maximize the typicality of the instances of C, we minimize the instances of $\neg\Box\neg C$. Notice that this is different from maximising the instances of $\mathbf{T}(C)$. We have adopted this solution since it allows to maximise the set of typical instances of C without affecting the extension of C (whereas maximising the extension of $\mathbf{T}(C)$ would imply maximising also the extension of C).

We define the set $\mathcal{M}^{\Box^-}_{\mathcal{L}_T}$ of negated boxed formulas holding in a model, relative to the concepts in $\mathcal{L}_T$. Given a model $\mathcal{M} = \langle \Delta, <, I \rangle$, let $\mathcal{M}^{\Box^-}_{\mathcal{L}_T} = \{(a, \neg\Box\neg C) \mid a \in (\neg\Box\neg C)^I,$ with $a \in \Delta, C \in \mathcal{L}_T\}$. Let KB be a knowledge base and let $\mathcal{L}_T$ be a set of concepts occurring in KB.

Definition 4.2. (Preferred and minimal models)
Given a model $\mathcal{M} = \langle \Delta_\mathcal{M}, <_\mathcal{M}, I_\mathcal{M} \rangle$ of KB and a model $\mathcal{N} = \langle \Delta_\mathcal{N}, <_\mathcal{N}, I_\mathcal{N} \rangle$ of KB, we say that $\mathcal{M}$ is preferred to $\mathcal{N}$ with respect to $\mathcal{L}_T$, and we write $\mathcal{M} <_{\mathcal{L}_T} \mathcal{N}$, if the following conditions hold:

- $\Delta_\mathcal{M} = \Delta_\mathcal{N}$;
- $\mathcal{M}^{\Box^-}_{\mathcal{L}_T} \subset \mathcal{N}^{\Box^-}_{\mathcal{L}_T}$.

A model $\mathcal{M}$ is a *minimal model* for KB (with respect to $\mathcal{L}_T$) if it is a model of KB and there is no a model $\mathcal{M}'$ of KB such that $\mathcal{M}' <_{\mathcal{L}_T} \mathcal{M}$.

Let us now define when a *query* F is minimally entailed in $\mathcal{ALC} + \mathbf{T}_{min}$ from a KB. A query F is either an inclusion $C \sqsubseteq D$ or a membership formula $C(a)$:

Definition 4.3. (Minimal Entailment in $\mathcal{ALC} + \mathbf{T}_{min}$)
A query F is minimally entailed from a knowledge base KB with respect to $\mathcal{L}_T$ if it holds in all models of KB minimal with respect to $\mathcal{L}_T$. We write KB $\models^{\mathcal{L}_T}_{min} F$.

While the original $\mathcal{ALC} + \mathbf{T}$ is *monotonic*, $\mathcal{ALC} + \mathbf{T}_{min}$ is *nonmonotonic*. Consider the following example. Let KB contains:

$$\mathbf{T}(\mathit{ItalianFencer}) \sqsubseteq \neg \mathit{LovedByPeople}$$
$$\mathit{ItalianFencer}(\mathit{aldo})$$
$$\mathit{SlimPerson}(\mathit{aldo})$$

and let $\mathcal{L}_T = \{\mathit{ItalianFencer}, \mathit{SlimPerson}\}$. We have that:

KB $\models^{\mathcal{L}_T}_{min} \neg \mathit{LovedByPeople}(\mathit{aldo})$

Indeed, there is a unique minimal model of KB on the domain $\Delta = \{\mathit{aldo}\}$, in which aldo is an instance of $\mathbf{T}(\mathit{ItalianFencer})$ (as well as an instance of $\mathbf{T}(\mathit{SlimPerson})$), and hence $\neg \mathit{LovedByPeople}$ holds in aldo. Observe that $\neg \mathit{LovedByPeople}(\mathit{aldo})$ is obtained in spite of the presence of the irrelevant property $\mathit{SlimPerson}(\mathit{aldo})$.

Consider also the knowledge base KB' obtained by adding to KB the formula:

$$\mathbf{T}(\mathit{ItalianFencer} \sqcap \mathit{SlimPerson}) \sqsubseteq \mathit{LovedByPeople}$$

and let $\mathcal{L}_T = \{\mathit{ItalianFencer}, \mathit{SlimPerson}, \mathit{ItalianFencer} \sqcap \mathit{SlimPerson}\}$. From KB', $\neg \mathit{LovedByPeople}$ (aldo) is not derivable any more. Instead, we have that:

KB' $\models^{\mathcal{L}_T}_{min} \mathit{LovedByPeople}(\mathit{aldo})$

KB' has a unique minimal model on the domain $\Delta = \{aldo, gianni\}$, in which $aldo$ is an instance of $\mathbf{T}(ItalianFencer \sqcap SlimPerson)$ and $\mathbf{T}(SlimPerson)$, but is not an instance of $\mathbf{T}(ItalianFencer)$ (as there is $gianni$, such that $gianni < aldo$ and $ItalianFencer$ holds at $gianni$). This example shows that, in case of conflict (here, $aldo$ cannot be both a typical instance of $ItalianFencer$ and $ItalianFencer \sqcap SlimPerson$), typicality in the more specific concept is preferred.

In [12] we have also defined a tableaux calculus for computing minimal entailment and we have provided an upper bound on the complexity of the resulting logic: checking if a query is minimally entailed from a KB is in CO-NEXP$^{\text{NP}}$.

5. Related works

Several nonmonotonic extensions of DLs have been proposed in the literature. In the following, we try to summarize the main approaches proposed in the literature. We conclude this section by providing a comparison with the KLM framework.

5.1. Nonmonotonic extensions of DLs

In [2] it is proposed the extension of DL with Reiter's default logic. Intuitively, a KB comprises, in addition to TBox and ABox, a finite set of default rules whose prerequisites, justifications, and consequents are concepts. Default rules are used in order to formalize prototypical properties. Concerning the KB about the Italian fencer of the Introduction, the direct encoding by normal (open) defaults would be:

$$\frac{ItalianFencer : \neg LovedByPeople}{\neg LovedByPeople} \qquad \frac{ItalianFencer \sqcap OlympicGoldMedalist : LovedByPeople}{LovedByPeople}$$

$$\frac{ItalianFencer \sqcap OlympicGoldMedalist \sqcap \exists TakePart.RealityShow : \neg LovedByPeople}{\neg LovedByPeople}$$

The same authors have pointed out that this integration may lead to both semantical and computational difficulties, both caused by an unsatisfactory treatment of open defaults via Skolemization. Skolemization of the ABox and of the consequents of default rules is needed in order to capture some intuitive inferences. For instance, given the above defaults and ABox $= \{\exists HasChild.ItalianFencer(mario)\}$, the intuitive conclusion $(*)$ $\exists HasChild.\neg LovedByPeople(mario)$ could not be deduced by default. Skolemization of ABox yields to ABox' $= \{HasChild(mario, giacomo), ItalianFencer(giacomo)\}$, where $giacomo$ is a new Skolem constant. Then, the closed defaults obtained by instantiating open defaults with $giacomo$ are applicable and allow to conclude $(*)$. However, Skolemization may lead to counterintuitive inferences. As an example, let us consider ABox$_1$ $= \{\exists HasChild.(ItalianFencer \sqcap Bold)(mario)\}$ and ABox$_2$ $= \{\exists HasChild.(ItalianFencer \sqcap Bold)(mario), \exists HasChild.ItalianFencer\ (mario)\}$. It is easy to see that ABox$_1$ and ABox$_2$ are logically equivalent. Skolemization leads to ABox$'_1$ $= \{HasChild(mario, giacomo), (ItalianFencer \sqcap Bold)(giacomo)\}$ and ABox$'_2$ $= \{HasChild(mario, paolo), (ItalianFencer \sqcap Bold)(paolo), HasChild(mario, antonio), ItalianFencer(antonio)\}$. Consider the open default:

$$\frac{ItalianFencer : \neg Bold}{\neg Bold}$$

This default rule does not fire for $giacomo$ and $paolo$, since their being in $ItalianFencer \sqcap Bold$ is inconsistent with the justification $\neg Bold$. On the contrary, this default rule fires for $antonio$, since his being an $ItalianFencer$ is consistent with the justification. Therefore, $(**)$ $\exists HasChild.\neg Bold(mario)$ is a default conclusion of ABox$'_2$, whereas it is not of ABox$'_1$. In our setting, neither ABox$_1$ nor ABox$_2$ can be used to infer $(**)$ in $\mathcal{ALC} + \mathbf{T}$. Consider the model $\mathcal{M} = \langle \Delta, <, I \rangle$, where $\Delta = \{mario^I, ivo^I, adelmo^I\}$, $adelmo^I < ivo^I$ and I is as follows: $ItalianFencer^I = \{ivo^I, adelmo^I\}$, $Bold^I = \{ivo^I\}$, $HasChild^I = \{(mario^I, ivo^I)\}$. We have that $(\mathbf{T}(ItalianFencer))^I = Min_<$ $(ItalianFencer^I) = \{adelmo^I\}$, therefore $\mathcal{M}$ is a model for the KB whose ABox=$\{\exists HasChild.$ $ItalianFencer(mario), \exists HasChild.(ItalianFencer \sqcap Bold)(mario)\}$ and whose TBox=$\{\mathbf{T}($ $ItalianFencer) \sqsubseteq \neg Bold\}$. However, $\exists HasChild.\neg Bold(mario)$ does not hold in $\mathcal{M}$. Notice that this also occurs if we refer to the minimal entailment $\models^{\mathcal{L}_T}_{min}$ of $\mathcal{ALC} + \mathbf{T}_{min}$, since $\mathcal{M}$ is also a minimal model for the mentioned KB. For further examples about semantical difficulties arising when integrating DLs with open defaults, we refer to Section 3 in [2].
The treatment of open defaults via Skolemization may also lead to an undecidable default consequence relation, even if the underlying logic is decidable. For this reason, [2] proposes a restricted semantics for open default theories, in which default rules are only applied to individuals explicitly mentioned in the ABox.

The extension of DLs with Reiter's default, even if restricted to explicitly mentioned individuals, presents a further drawback, namely it inherits from general default logic the difficulty of modeling inheritance with exceptions giving precedence to more specific defaults in a direct way. For instance, the above formulation of the KB does not allow to infer correctly the expected conclusions, as it does not give priority to more specific information. Consider, for instance, the preceding KB containing the default rules above and whose ABox contains $ItalianFencer(aldo)$ as well as $OlympicGoldMedalist(aldo)$. Such a terminological default theory has two extensions, one containing $\neg LovedByPeople(aldo)$ and the other $LovedByPeople(aldo)$; the semantics does not allow to prefer the second one, in which the most specific default was applied. This behaviour represents a critic aspect in the general context of default logic, however it is more problematic in a DL framework where the emphasis lies on the hierarchical organization of the concepts. To attack this problem, one has to impose a priority on default application or to find a smarter (but ad hoc) encoding of defaults giving priority to more specific information. This has motivated the study of extensions of DLs with prioritized defaults [23, 3, 6, 5].

To give a brief account, in [23] the author introduces an extension of DLs to perform default inheritance reasoning, a kind of default reasoning specifically tailored to reason in presence of a taxonomy of concepts. This formalism allows for defaults $C \mapsto D$ as well as $C \mapsto R.D$, whose intuitive meanings are: "if a is an element of C and the assumption that a is an element of D is consistent, then assume that a is a D" and "if a is an element of C and b is related to a by the role R and the assumption that b is an element of D is consistent, then assume that b is a D", respectively. Specificity is handled by defining, for a given KB=(TBox,ABox) and an individual a occurring in the ABox, a preference relation $\preceq_{KB,a}$ over atomic concepts. As pointed out by the author, the relation $\preceq_{KB,a}$ can be seen as "the taxonomy induced by the strict and defeasible information, altogether", belonging to the KB. Extensions are computed by means of a fixpoint construction which takes into account the relation $\preceq_{KB,a}$. Intuitively, given a default $C \mapsto D$, for all $C(a)$ belonging to the extension under construction, $D(a)$ is added to the extension unless this leads to a contradiction or there is a concept E such that $E \preceq_{KB,a} C$, $C \not\preceq_{KB,a} E$ and either (i) $E \mapsto F$ is also a default of the theory and $(D \sqcap F)(a)$ is inconsistent or (ii) $E \mapsto R.F$ is a default of the theory, $R(a, b)$ is in the ABox and the addiction of both $D(a)$ and $F(b)$ leads to an inconsistency. Similarly for defaults of the form $C \mapsto R.D$.

In [6] priorities among defaults are addressed by ordered default theories. The basic idea is to consider a strict partial order on the set of defaults when computing extensions. However, this approach is restricted to prerequisite-free normal defaults only. Such a severe restriction has motivated the generalization of ordered default theories to normal defaults with prerequisites [5, 3]. Similarly to [6], priorities are given by an arbitrary partial order on defaults. As a difference with [23], priorities between defaults are induced by the position of their prerequisites in the concept hierarchy of the TBox, then the specificity is determined by the strict information, and not by the defaults. Intuitively, given a KB and a set $\mathcal{D}$ of defaults, terminological default rules are obtained by instatiating each $d \in \mathcal{D}$ by all constants occurring in the ABox. Let d_1 and d_2 be two defaults so obtained, having $C(a)$ and $D(b)$ as prerequisites, respectively: we have that $d_1 < d_2$, i.e. d_1 has a major priority with respect to d_2, if and only if (i) $a = b$, (ii) $C \sqsubseteq D$ follows from the KB and (iii) $D \sqsubseteq C$ does not follow from the KB. The consequent of a default d can be added to the extension if d is not delayed by a preferred default, in other words if there is no $d' < d$ which is *active*, i.e. applicable in the current set of formulas of the extension. The authors also describe an algorithm for computing extensions.

As for the proposal in [2], in order to avoid semantical and computational difficulties due to the treatment of open defaults via Skolemization, all these approaches adopt a semantics in which defaults are only applied to individuals explicitly mentioned in the ABox, thus introducing an asymmetric treatment of domain elements.

A more general approach is undertaken in [10], where Description Logics of minimal knowledge and negation as failure are proposed by augmenting DLs with two epistemic operators, K and A, interpreted according to Lifschitz's nonmonotonic logic MKNF [20, 21]. In particular, [10] studies the extension of $\mathcal{ALC}$, called $\mathcal{ALCK}_{\mathcal{NF}}$, which allows to capture Reiter's default logic, integrity constraints, procedural rules as well as role and concept closure. The paper provides a sound, complete and terminating tableau calculus for checking satisfiability of *simple* $\mathcal{ALCK}_{\mathcal{NF}}$ KBs, where in a simple KB the occurrences of the operator K within the scope of quantifiers are limited. The calculus uses triple exponential time in the size of the KB. For MKNF-DLs without quantifying-in (i.e., with no occurrences of epistemic operators in the scope of quantifiers), a general deductive method can be defined (see [9]), which is parametric with respect to the underlying DL. The authors prove that the problem of instance checking in a MKNF-DL without quantifying-in is decidable if and only if the problem of instance checking in the underlying DL is decidable. In particular, for the logic $\mathcal{ALCK}_{\mathcal{NF}}$ without quantifying-in the problem of instance checking is EXPTIME-complete as in the non-modal case. [17] extends the work in [10] by providing a translation of an $\mathcal{ALCK}_{\mathcal{NF}}$ KB to an equivalent *flat* KB and by defining a simplified tableau algorithm for flat KBs, which includes an optimized minimality check.

In both [10] and [9], the domain of epistemic interpretations is assumed to be countably-infinite and to be the same for all interpretations. Although this assumption restricts the semantics of first-order MKNF, nevertheless it allows an encoding of prerequisite-free defaults with an open semantics. [10] also provides an encoding of closed defaults by translating them into simple $\mathcal{ALCK}_{\mathcal{NF}}$ inclusions. Although [10, 9] introduce very general nonmonotonic DLs, they do not address specifically the problem of reasoning about inheritance with exceptions, nor the problem of specificity on which we focus in this paper.

In [4] the authors propose an extension of DL with circumscription. One of the motivating applications of circumscription is indeed to express prototypical properties with exceptions, and this is done by introducing "abnormality" predicates, whose extension is minimized. The basic idea is as follows: in order to express $\mathbf{T}(C) \sqsubseteq D$, the authors introduce the inclusion

$$C \sqsubseteq D \sqcup Ab_C$$

where Ab_C is the predicate to be minimized. Concerning again the KB of the Introduction, one has to introduce abnormality predicates (Ab_i), e.g.:

$\mathit{ItalianFencer} \sqsubseteq \neg \mathit{LovedByPeople} \sqcup Ab_1$
$\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqsubseteq \mathit{LovedByPeople} \sqcup Ab_2$
$\mathit{ItalianFencer} \sqcap \mathit{OlympicGoldMedalist} \sqcap \exists \mathit{TakePart}.\mathit{RealityShow} \sqsubseteq \neg \mathit{LovedByPeople} \sqcup Ab_3$

Then one has to establish which predicates are minimized, fixed, or variable (the so-called circumscription pattern). The basic idea of circumscription is indeed to consider only those models where the extension of abnormality predicates is minimal with respect to set inclusion.

Circumscription patterns in [4] also allow to express priorities among predicates to be minimized. As pointed out by the authors, these priorities usually reflect the taxonomy described by the TBox and, since the subsumption hierarchy is a partial order, priorities are assumed to form a partial order, too, as a difference with standard prioritized circumscription which assumes a total ordering.

The authors provide decidability and complexity results based on theoretical analysis. In detail, it is shown that reasoning is decidable under the restriction that only concepts can be circumscribed, whereas roles have to vary during circumscription. This also holds for expressive DLs such as $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$. Allowing roles to be fixed during minimization leads to an undecidability results even in the extension of basic $\mathcal{ALC}$.

As in our approach, the extension of DLs with circumscription avoids the restriction of reasoning about elements explicitly mentioned in the ABox. However, the authors do not provide a calculus for their logic.

5.2. Relations with KLM

We have already mentioned that the semantics of the typicality operator $\mathbf{T}$ of Definition 2.1 is strongly related with the semantics of nonmonotonic entailment in KLM preferential logic **P**. Let us make precise the relation between the two logics. KLM logic **P** is originally defined as a propositional logic, thus we restrict our analysis to the propositional level. The basic assertions of KLM logics are positive conditionals of the form $A \mid\!\sim B$, where A, B are propositional formulas. In the literature, a few kinds of languages or knowledge bases allowing conditionals have been considered: (i) a knowledge base is a set of conditionals, this is indeed the original setting by KLM [18]; (ii) a knowledge base is a set of conditionals and negated conditionals [19]; (iii) a knowledge base is or contains arbitrary boolean combinations of conditionals (and possibly propositional formulas) [14]. The axiomatization of **P** for the richer language (iii) is given by a refolmulation of KLM postulates through the following set of axioms, where $\vdash_{PC}$ denotes validity in classical propositional calculus:

- REF. $A \mid\!\sim A$ (reflexivity)
- LLE. If $\vdash_{PC} A \leftrightarrow B$, then $\vdash (A \mid\!\sim C) \rightarrow (B \mid\!\sim C)$ (left logical equivalence)
- RW. If $\vdash_{PC} A \rightarrow B$, then $\vdash (C \mid\!\sim A) \rightarrow (C \mid\!\sim B)$ (right weakening)
- CM. $((A \mid\!\sim B) \wedge (A \mid\!\sim C)) \rightarrow (A \wedge B \mid\!\sim C)$ (cautious monotonicity)
- AND. $((A \mid\!\sim B) \wedge (A \mid\!\sim C)) \rightarrow (A \mid\!\sim B \wedge C)$

- OR. $((A \mathrel{|\!\sim} C) \wedge (B \mathrel{|\!\sim} C)) \rightarrow (A \vee B \mathrel{|\!\sim} C)$

To axiomatize the restricted languages (i) and (ii), axioms must be suitably replaced by inference rules. We establish here a mapping between **P** and $\mathcal{ALC} + \mathbf{T}$ for the language (ii), that is a knowledge base will be henceforth a set of positive and negative conditionals. The most general case (iii) is problematic, since $\mathcal{ALC} + \mathbf{T}$ does not allow for disjunctions of subsumption relations.

A model $\mathcal{M}$ for **P** is defined similarly to Definition 2.3, but the language is slightly different.

Definition 5.1. (Semantics of $\mathrel{|\!\sim}$)
A **P** model $\mathcal{M}$ for a propositional language $\mathcal{L}$ has the form $\mathcal{M} = (\Delta, <, I)$, where Δ and $<$ are as in Definition 2.3, $I : \mathit{VarProp} \mapsto \mathit{Pow}(\Delta)$. We define:

$$\mathcal{M} \models A \mathrel{|\!\sim} B \text{ iff } Min_<(A^I) \subseteq B^I \text{ and } \mathcal{M} \models \neg(A \mathrel{|\!\sim} B) \text{ iff } \mathcal{M} \not\models A \mathrel{|\!\sim} B.$$

Let KB be a set of positive or negated conditionals F. $\mathcal{M} \models$ KB iff $\mathcal{M} \models F$ for every $F \in$ KB.

There is an obvious correspondence between **P** models and $\mathcal{ALC} + \mathbf{T}$ models. In the following, given an $\mathcal{ALC} + \mathbf{T}$ model $\mathcal{M} = \langle \Delta, <, I \rangle$, we write $\mathcal{M} \models C \sqsubseteq D$ to say that $C^I \subseteq D^I$.

Lemma 5.1. (a) Let $\mathcal{M} = (\Delta, <, I)$ be a **P** model for a propositional language $\mathcal{L}$, then the model $\mathcal{M}^+ = (\Delta, <, I)$, with $\mathbf{T}(A)^I = Min_<(A^I)$ is an $\mathcal{ALC} + \mathbf{T}$ model for $\mathcal{L}$ that satisfies:

$$\mathcal{M} \models A \mathrel{|\!\sim} B \text{ iff } \mathcal{M}^+ \models \mathbf{T}(A) \sqsubseteq B.$$

(b) Let $\mathcal{M} = (\Delta, <, I)$ be an $\mathcal{ALC} + \mathbf{T}$ model for $\mathcal{L}$ (the propositional variables are the concepts names), then the model $\mathcal{M}^- = (\Delta, <, I)$, with $\mathcal{M}^- \models A \mathrel{|\!\sim} B$ iff $Min_<(A^I) \subseteq B^I$ is a **P** model for $\mathcal{L}$ that satisfies:

$$\mathcal{M}^- \models A \mathrel{|\!\sim} B \text{ iff } \mathcal{M} \models \mathbf{T}(A) \sqsubseteq B.$$

We can generalize this correspondence to knowledge bases. Let KB be a finite knowledge base in **P**:

$$\text{KB} = \{A_1 \mathrel{|\!\sim} B_1, \dots, A_m \mathrel{|\!\sim} B_m, \neg(C_1 \mathrel{|\!\sim} D_1), \dots, \neg(C_n \mathrel{|\!\sim} D_n)\}$$

We define the $\mathcal{ALC} + \mathbf{T}$ translation KB$^+$ of KB as follows:

$$\{\mathbf{T}(A_1) \sqsubseteq B_1, \dots, \mathbf{T}(A_m) \sqsubseteq B_m, \mathbf{T}(C_1)(i_1), \neg D_1(i_1), \dots, \mathbf{T}(C_n)(i_n), \neg D_n(i_n)\}$$

where $i_1, \dots, i_n$ are distinct individuals. KB and its translation KB$^+$ are related as expressed in the following lemma.

Lemma 5.2. Let KB and KB$^+$ as above. (a) Let $\mathcal{M} = (\Delta, <, I)$ be a **P** model of KB, then there is an $\mathcal{ALC} + \mathbf{T}$ model $\mathcal{M}^+$ of KB$^+$ such that:

$$\mathcal{M} \models A \mathrel{|\!\sim} B \text{ iff } \mathcal{M}^+ \models \mathbf{T}(A) \sqsubseteq B.$$

(b) Let $\mathcal{M} = (\Delta, <, I)$ be a $\mathcal{ALC} + \mathbf{T}$ model of KB$^+$ then there is a **P** model $\mathcal{M}^-$ of KB such that:

$$\mathcal{M}^- \models A \mathrel{|\!\sim} B \text{ iff } \mathcal{M} \models \mathbf{T}(A) \sqsubseteq B.$$

Proof:
(a) Let $\mathcal{M} = (\Delta, <, I)$ be a model of KB. We have in particular that $\mathcal{M} \models \neg(C_j \mathrel{|\!\sim} D_j)$ for $j = 1, \ldots, n$; thus $Min_<(C_j^I) - D_j^I \neq \emptyset$ for $j = 1, \ldots, n$. Therefore there are elements $d_j \in Min_<(C_j^I) - D_j^I$, for $j = 1, \ldots, n$. We define $\mathcal{M}^+ = (\Delta, <, I)$, by stipulating $i_j^I = d_j$ for $j = 1, \ldots, n$ and $\mathbf{T}(A)^I = \min_<(A^I)$. It is immediate to see that $\mathcal{M} \models A \mathrel{|\!\sim} B$ iff $\mathcal{M}^+ \models \mathbf{T}(A) \sqsubseteq B$ and that $\mathcal{M}^+ \models$ KB$^+$.

(b) Let $\mathcal{M} = (\Delta, <, I)$ be an $\mathcal{ALC} + \mathbf{T}$ model of KB$^+$. It is easily seen that $\mathcal{M} \not\models \mathbf{T}(C_j) \sqsubseteq D_j$, for $j = 1, \ldots, n$. Thus the model $\mathcal{M}^-$ obtained by stipulating $\mathcal{M}^- \models A \mathrel{|\!\sim} B$ iff $\mathcal{M} \models \mathbf{T}(A) \sqsubseteq B$ and by omitting the interpretation of the individuals i_j is a **P** model of KB. □

By the previous lemma we easily obtain the following proposition that summarizes the relation between **P** and $\mathcal{ALC} + \mathbf{T}$ with respect to reasoning tasks:

Proposition 5.1. Let KB be a **P** knowledge base and KB$^+$ its translation into $\mathcal{ALC} + \mathbf{T}$. Then:
(a) for any conditional $A \mathrel{|\!\sim} B$, KB $\models A \mathrel{|\!\sim} B$ in **P** iff KB$^+ \models \mathbf{T}(A) \sqsubseteq B$ in $\mathcal{ALC} + \mathbf{T}$. Thus, (b) KB is satisfiable in **P** iff KB$^+$ is satisfiable in $\mathcal{ALC} + \mathbf{T}$.

This semantic correspondence between **P** and $\mathcal{ALC} + \mathbf{T}$ helps to understand intuitively the relation between the axioms of **P** and the semantic conditions of Definition 2.1[3]. We illustrate the correspondence by means of the translation $A \mathrel{|\!\sim} B$ as $\mathbf{T}(A) \sqsubseteq B$ and the axioms of **P**:

- REF. $\mathbf{T}(A) \sqsubseteq A$ corresponding to $(f_\mathbf{T} - 1)$
- LLE. by taking $C = \mathbf{T}(B)$, we get: if $\vdash_{PC} A \leftrightarrow B$ then $\mathbf{T}(A) \sqsubseteq \mathbf{T}(B)$. This property gives the independence from the syntax and is implicitly satisfied by any semantics.
- CM. by taking $C = \mathbf{T}(A)$, $\mathbf{T}(A) \sqsubseteq B$ implies $\mathbf{T}(A \sqcap B) \sqsubseteq \mathbf{T}(A)$. The other inclusion is derivable as well. They jointly give $(f_\mathbf{T} - 3)$.
- OR. by taking $C = \mathbf{T}(A) \sqcup \mathbf{T}(B)$, we get $\mathbf{T}(A \sqcup B) \sqsubseteq \mathbf{T}(A) \sqcup \mathbf{T}(B)$. This corresponds to the finitary version of $(f_\mathbf{T} - 4)$.

On the other hand there is no way to derive from **P** (a finitary version of) condition $(f_\mathbf{T} - 5)$, since in **P** we cannot express the fact that something is both a typical C and a typical D. This suggests that the conditional operator $\mathrel{|\!\sim}$ is too weak to deal with the notion of typicality.

To end this section we mention that the family of KLM logics contains other interesting members, notably the stronger logic **R**, known as Rational Preferential Logic. This system is obtained by adding to **P** the axiom/rule of rational monotonicity:

$$A \mathrel{|\!\sim} C \wedge \neg(A \mathrel{|\!\sim} \neg B) \rightarrow ((A \wedge B) \mathrel{|\!\sim} C)$$

That is to say, from $A \mathrel{|\!\sim} C$ we can conclude $(A \wedge B) \mathrel{|\!\sim} C$ unless we can derive $A \mathrel{|\!\sim} \neg B$. For a discussion and a justification of this property we refer to the literature [19]. We also mention that many (but not all) systems of probabilistic entailment satisfy this property. The semantics of rational logic **R** is well-understood: the rational monotonicity principle corresponds to the additional property of *modularity* of

[3] In this work we have given two equivalent (via the Representation Theorem 2.1) semantical characterizations of $\mathcal{ALC} + \mathbf{T}$, but we have not addressed the problem of giving a full axiomatization of the logic of typicality. The reason is that it is not our main concern: we are interested in reasoning with respect to a knowledge base and not in deriving abstract properties of the typicality operator. However, the problem of the axiomatization may be posed; to deal with it, we should first ask ourself what kind of assertions we allow in the language: inclusion relations or arbitrary combinations of them?

the preference relation. We could think of developing a logic of typicality built upon rational logic **R** rather than the weaker preferential logic **P** as we do. What is the impact of rational monotonicity to reason about typicality? Translating the axiom into a property of the typicality operator we obtain:

$$\neg(\mathbf{T}(A) \sqcap B \sqsubseteq \bot) \text{ implies } \mathbf{T}(A \sqcap B) \sqsubseteq \mathbf{T}(A)$$

Thus it is sufficient that there is *one* individual that is a typical A and that has the property B, to conclude that *all* typical A and Bs are typical A. For instance if $aldo$ is a typical italian fencer and he won a gold medal in an Olympic competition (these might be assertions in the ABox), then all typical italian fencers winning the gold medal are typical italian fencers. This seems rather arbitrary and counterintuitive. All of this means that the logic **R** is too strong and unsuitable to reason about typicality.

6. Conclusions and Future Work

We have proposed an extension of $\mathcal{ALC}$ for reasoning about typicality in the Description Logic framework. For the resulting logic, called $\mathcal{ALC} + \mathbf{T}$, we have defined a calculus for deciding the satisfiability of a general knowledge base. The calculus, called $T^{\mathcal{ALC}+\mathbf{T}}$, is analytic, terminating, and allows us to decide the satisfiability of a knowledge base in $\mathcal{ALC} + \mathbf{T}$ in nondeterministic exponential time. The work is an extended and revised version of the preliminary contribution presented in [13].

The basic monotonic logic $\mathcal{ALC} + \mathbf{T}$ results to be too weak to perform defeasible inferences. We have shown how to address this problem by presenting two different approaches. On the one hand, we have proposed a *completion* mechanism, whose objective is to complete the ABox by means of typicality assumptions, in order to infer prototypical properties of the individuals explicitly mentioned in the ABox. On the other hand, we have developed a preferential semantics. This nonmonotonic extension of $\mathcal{ALC} + \mathbf{T}$ allows for defeasible reasoning in presence of inheritance with exceptions.

We plan to extend this work in several directions. First of all, the tableau procedure we have described can be optimised in many ways. In particular, we believe that the calculus $T^{\mathcal{ALC}+\mathbf{T}}$ can be made more efficient by applying standard techniques such as caching.

From the point of view of knowledge representation, a limit of our logic is the unability to handle inheritance of multiple properties in case of exceptions as in the example: $\mathbf{T}(Student) \sqsubseteq \neg HasIncome$, $\mathbf{T}(Student) \sqsubseteq \exists Owns.LibraryCard$, $PhDStudent \sqsubseteq Student$, $\mathbf{T}(PhDStudent) \sqsubseteq HasIncome$. Our semantics does not support the inference $\mathbf{T}(PhDStudent) \sqsubseteq \exists Owns.LibraryCard$, that is, PhDStudents typically own a library card, as we might want to conclude (since having an income has nothing to do with owning a library card). The reason why our semantics fails to support this inference is that the first two inclusions are obviously equivalent to the single one $\mathbf{T}(Student) \sqsubseteq \neg HasIncome \sqcap \exists Owns.$ $LibraryCard$ which is contradicted by $\mathbf{T}(PhDStudent) \sqsubseteq HasIncome$. To handle this type of inferences we would need a tighter semantics where the truth of $\mathbf{T}(C) \sqsubseteq P$ is no longer a function of $\mathbf{T}(C)$ and P or a smarter (and less direct) encoding of the knowledge. Observe that the same problem arises for instance with circumscription, where we would need at least different abnormality predicates *for each pair* of concept-defeasible property.

KLM logics, which are at the base of our semantics, are related to probabilistic reasoning. In [16], the notion of conditional constraint allows typicality assertions to be expressed (with a specified interval of probability values). In order to perform defeasible reasoning, a notion of minimal entailment is introduced based on a *lexicographic preference* relation on probabilistic interpretations. We plan to compare in details this probabilistic approach to our approach in further research.

Acknowledgements. The work has been partially supported by Regione Piemonte, Project "ICT4Law - *ICT Converging on Law: Next Generation Services for Citizens, Enterprises, Public Administration and Policymakers*".

References

[1] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., Eds.: *The Description Logic Handbook - Theory, Implementation, and Applications, 2nd edition,* Cambridge, 2007.

[2] Baader, F., Hollunder, B.: Embedding Defaults into Terminological Knowledge Representation Formalisms, *J. Autom. Reasoning*, **14**(1), 1995, 149–180.

[3] Baader, F., Hollunder, B.: Priorities on Defaults with Prerequisites, and Their Application in Treating Specificity in Terminological Default Logic, *J. of Automated Reasoning (JAR)*, **15**(1), 1995, 41–68.

[4] Bonatti, P. A., Lutz, C., Wolter, F.: Description Logics with Circumscription., *Proc. of KR*, 2006.

[5] Brewka, G.: Adding Priorities and Specificity to Default Logic, *Proceedings of the European Workshop on Logics in Artificial Intelligence* (C. MacNish, D. Pearce, L. Moniz Pereira, Eds.), 838, Springer, Verlag, 1994.

[6] Brewka, G., Junker, U.: Handling partially ordered defaults in TMS, *Proceedings of the 1st European Conference on Symbolic and Quantitative Approaches for Uncertainty, ECSQARU, Marseille, France*, 548, Springer, 1991.

[7] Buchheit, M., Donini, F. M., Schaerf, A.: Decidable Reasoning in Terminological Knowledge Representation Systems, *J. Artif. Int. Research (JAIR)*, **1**, 1993, 109–138.

[8] Donini, F. M., Lenzerini, M., Nardi, D., Nutt, W., Schaerf, A.: An Epistemic Operator for Description Logics, *Artif. Intell.*, **100**(1-2), 1998, 225–274.

[9] Donini, F. M., Nardi, D., Rosati, R.: Autoepistemic Description Logics, *Proceedings of IJCAI (1)*, 1997.

[10] Donini, F. M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure, *ACM Trans. Comput. Log.*, **3**(2), 2002, 177–225.

[11] Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining Answer Set Programming with Description Logics for the Semantic Web, *KR 2004, 141-151*.

[12] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G. L.: Reasoning About Typicality in Preferential Description Logics, *Proceedings of JELIA 2008.* LNCS(LNAI), vol. 5293, pp. 192-205 (S. Hólldobler, C. Lutz, H. Wansing, Eds.), Springer, Heidelberg.

[13] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G. L.: Preferential Description Logics, *Proceedings of LPAR 2007.* LNCS(LNAI), vol. 4790, pp. 257-272, 2007.

[14] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G. L.: Analytic Tableaux Calculi for KLM Logics of Nonmonotonic Reasoning, *ACM Transactions on Computational Logics (TOCL)*, **10**(3), 2009.

[15] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G. L.: On Extending Description Logics for Reasoning About Typicality: a First Step, *TR Dip. di Informatica*, `http://www.di.unito.it/∼pozzato/tr09.pdf`, 2009.

[16] Giugno, R., Lukasiewicz, T.: P-$\mathcal{SHOQ}$(D): A Probabilistic Extension of $\mathcal{SHOQ}$(D) for Probabilistic Ontologies in the Semantic Web, *Proceedings of JELIA 2002.* LNCS(LNAI), vol. 2424, pp. 86-97, 2002.

[17] Ke, P., Sattler, U.: Next Steps for Description Logics of Minimal Knowledge and Negation as Failure, *Proceedings of DL2008*, 2008.

[18] Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic Reasoning, Preferential Models and Cumulative Logics, *Artificial Intelligence*, **44**(1-2), 1990, 167–207.

[19] Lehmann, D., Magidor, M.: What does a conditional knowledge base entail?, *Artificial Intelligence*, **55**(1), 1992, 1–60.

[20] Lifschitz, V.: Nonmonotonic Databases and Epistemic Queries, *Proceedings of IJCAI*, 1991.

[21] Lifschitz, V.: Minimal Belief and Negation as Failure, *Artificial Intelligence*, **70**(1-2), 1994, 53–72.

[22] Papadimitriou, C. H.: *Computational Complexity*, Addison-Wesley Publishing Company, 1994.

[23] Straccia, U.: Default Inheritance Reasoning in Hybrid KL-ONE-Style Logics, *Proc. of IJCAI*, 1993.