Reasoning About Typicality in Preferential Description Logics

Laura Giordano[⋆], Valentina Gliozzi[⊙], Nicola Olivetti[∗], Gian Luca Pozzato[⊙]

[⋆] Dip. di Informatica - Univ. Piemonte O. "A. Avogadro" - laura@mfn.unipmn.it
[⊙] Dip. di Informatica - Università di Torino - {gliozzi,pozzato}@di.unito.it
[∗] LSIS-UMR CNRS 6168 Univ. "P. Cézanne" - nicola.olivetti@univ-cezanne.fr

Abstract. In this paper we propose a nonmonotonic extension $\mathcal{ALC} + \mathbf{T}_{min}$ of the Description Logic $\mathcal{ALC}$ for reasoning about prototypical properties and inheritance with exception. The logic $\mathcal{ALC} + \mathbf{T}_{min}$ is built upon a previously introduced (monotonic) logic $\mathcal{ALC} + \mathbf{T}$, that is obtained by adding a typicality operator $\mathbf{T}$ to $\mathcal{ALC}$. The operator $\mathbf{T}$ is intended to select the "most normal" or "most typical" instances of a concept, so that knowledge bases may contain subsumption relations of the form"$\mathbf{T}(C)$ is subsumed by P", expressing that typical C-members have the property P. In order to perform nonmonotonic inferences, we define a "minimal model" semantics $\mathcal{ALC} + \mathbf{T}_{min}$ for $\mathcal{ALC} + \mathbf{T}$. The intuition is that preferred, or minimal models are those that maximise typical instances of concepts. By means of $\mathcal{ALC} + \mathbf{T}_{min}$ we are able to infer defeasible properties of (explicit or implicit) individuals. We also present a tableau calculus for deciding $\mathcal{ALC} + \mathbf{T}_{min}$ entailment.

1 Introduction

The family of description logics (DLs) is one of the most important formalisms of knowledge representation. They have a well-defined semantics based on first-order logic and offer 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: the TBox, containing the definition of concepts (and possibly roles), and a specification of inclusions relations among them, and the ABox containing instances of concepts and roles. Since the very 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 naturally 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 [1–3, 5, 6, 12]. However, finding a suitable nonmonotonic extension for inheritance with exceptions is far from obvious.

To give a brief account, [1] proposes the extension of DL with Reiter's default logic. However, the same authors have pointed out that this integration may lead to both semantical and computational difficulties. Furthermore, Reiter's default logic does not provide a direct way of modeling inheritance with exceptions. This has motivated the study of extensions of DLs with prioritized defaults [12, 2]. A more general approach is undertaken in [5], where an extension of DL is proposed with two epistemic operators. This extension, called $\mathcal{ALCK}_{\mathcal{NF}}$, allows one to encode Reiter's default logic as well as to express epistemic concepts and procedural rules. However, this extension has a rather complicated modal semantics, so that the integration with the existing systems requires significant changes to the standard semantics of DLs. [9] extends the work in [5] 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 [3] an extension of DL with circumscription is proposed to express prototypical properties with exceptions, by introducing "abnormality" predicates whose extension is minimized. The authors provide algorithms for checking satisfiability, subsumption and instance checking which are proved to have an optimal complexity, but are based on massive nondeterministic guessing. A calculus for circumscription in DL has not been developed yet. Moreover, the use of circumscription to model inheritance with exceptions is not that straightforward.

In this work, we propose a new nonmonotonic logic $\mathcal{ALC} + \mathbf{T}_{min}$ for defeasible reasoning in description logic. Our starting point is the monotonic logic $\mathcal{ALC} + \mathbf{T}$ introduced in [7], obtained by adding a typicality operator $\mathbf{T}$ to $\mathcal{ALC}$. The intended meaning of the operator $\mathbf{T}$, for any concept C, is that $\mathbf{T}(C)$ singles out the instances of C that are considered as "typical" or "normal". Thus assertions as "normally students do not pay taxes", or "typically users do not have access to confidential files" [3] are represented by $\mathbf{T}(Student) \sqsubseteq \neg TaxPayer$ and $\mathbf{T}(User) \sqsubseteq \neg \exists HasAccess.ConfidentialFile$. As shown in [7], the operator $\mathbf{T}$ is characterised by a set of postulates that are essentially a reformulation of KLM [10] axioms of preferential logic $\mathbf{P}$, namely the assertion $\mathbf{T}(C) \sqsubseteq P$ is equivalent to the conditional assertion $C \mid\!\sim P$ of $\mathbf{P}$. It turns out that the semantics of the typicality operator can be equivalently specified by a suitable modal logic.

The idea underlying the modal interpretation is that there is a global preference relation (a strict partial order) $<$ on individuals, so that typical instances of a concept C can be defined as the instances of C that are minimal with respect to $<$. In this modal logic, $<$ works as an accessibility relation R with $R(x, y) \equiv y < x$, so that we can define $\mathbf{T}(C)$ as $C \sqcap \Box \neg C$. The preference relation $<$ does not have infinite descending chains as we adopt the so-called Smoothness condition or Limit Assumption of conditional logics. As a consequence, the corresponding modal operator $\Box$ has the same properties as in Gödel-Löb modal logic G of arithmetic provability.

In our setting, 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 D is a concept not mentioning $\mathbf{T}$. For instance, let the KB contain: $\mathbf{T}(Student) \sqsubseteq \neg TaxPayer$; $\mathbf{T}(Student \sqcap Worker) \sqsubseteq TaxPayer$; $\mathbf{T}(Student \sqcap Worker \sqcap \exists HasChild.\top) \sqsubseteq \neg TaxPayer$, corresponding to the assertions: normally a student does not pay taxes, normally a working student pays taxes, but normally a working student having children does not pay taxes. Suppose further that the ABox contains alternatively the following facts about *john*: 1. $Student(john)$; 2. $Student(john)$, $Worker(john)$; 3. $Student(john)$, $Worker(john)$, $\exists HasChild.\top(john)$. We would like to infer the expected (defeasible) conclusion about *john* in each case: 1. $\neg\,TaxPayer(john)$, 2. $TaxPayer(john)$, 3. $\neg\,TaxPayer(john)$. Moreover, we would like to infer (defeasible) properties also of individuals implicitly introduced by existential restrictions, for instance, if the ABox further contains $\exists HasChild.(Student \sqcap Worker)(jack)$ it should derive (defeasibly) the "right" conclusion $\exists HasChild.TaxPayer(jack)$ in the latter. Finally, adding irrelevant information should not affect the conclusions. Given the KB as above, one should be able to infer as well $\mathbf{T}(Student \sqcap SportLover) \sqsubseteq \neg\,TaxPayer$ and $\mathbf{T}(Student \sqcap Worker \sqcap SportLover) \sqsubseteq TaxPayer$, as *SportLover* is irrelevant with respect to being a TaxPayer or not. For the same reason, the conclusion about *john* being a TaxPayer or not should not be influenced by adding $SportLover(john)$ to the ABox.

The monotonic logic $\mathcal{ALC} + \mathbf{T}$ is not sufficient to perform the kind of defeasible reasoning illustrated above. Concerning the example, we get for instance that: KB $\cup$ $\{Student(john), Worker(john)\}$ $\not\models$ $TaxPayer(john)$; KB $\not\models$ $\mathbf{T}(Student \sqcap SportLover) \sqsubseteq \neg\,TaxPayer$. In order to derive the conclusion about *john* we should know (or assume) that *john* is a typical working student, but we do not dispose of this information. Similarly, in order to derive that also a typical student who loves sport must not pay taxes, we must be able to infer or assume that a typical "student loving sport" is also a "typical student", since there is no reason why it should not be the case; this 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. In [7], we proposed a completion of the KB that adds, for each individual, the assumption that the individual is a typical member of the *most specific concept* to which it belongs. However, this solution presents some difficulties: (i) it is not clear how to take into account implicit individuals, (ii) the completion might be inconsistent, so that we must consider alternative maximal completions, (iii) 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 (this would require a kind of fixpoint definition).

In this work we follow another approach, rather than defining an ad-hoc mechanism to perform defeasible inferences or making nonmonotonic assumptions, we strenghten 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. Similarly to circumscription, where we must specify a set of minimised predicates, here we must specify a set of concepts $\mathcal{L}_T$ of which we want to maximise the set of typical instances (it may just be the set of all concepts occurring in the knowledge base). We call the new logic $\mathcal{ALC} + \mathbf{T}_{min}$ and we denote by $\models_{min}^{\mathcal{L}_T}$ semantic entailment determined by minimal models. Taking the KB of the examples above we obtain, for instance, KB $\cup$ $\{Student(john),\, Worker(john)\} \models^{\mathcal{L}_T}_{min}$ $TaxPayer(john)$; KB $\cup$ $\{\exists HasChild.(Student \sqcap Worker)(jack)\} \models^{\mathcal{L}_T}_{min} \exists HasChild.\neg TaxPayer(jack)$ and KB $\models^{\mathcal{L}_T}_{min}$ $\mathbf{T}(Student \sqcap SportLover) \sqsubseteq \neg TaxPayer$. As the second example shows, we are able to infer the intended conclusion also for the implicit individuals.

We provide a decision procedure for checking satisfiability and validity in $\mathcal{ALC} + \mathbf{T}_{min}$. Our decision procedure has the form of tableaux calculus, with a two-step tableau construction. The idea is that the top level construction generates open branches that are candidates to represent minimal models, whereas the auxiliary construction checks whether a candidate branch represents indeed a minimal model. Termination is ensured by means of a standard blocking mechanism. Our procedure can be used to determine constructively an upper bound of the complexity of $\mathcal{ALC} + \mathbf{T}_{min}$. Namely we obtain that checking query entailment for $\mathcal{ALC} + \mathbf{T}_{min}$ is in co-NExp$^{\mathrm{NP}}$.

2 The logic $\mathcal{ALC} + \mathbf{T}$

In this section, we summarize the characteristics of the original $\mathcal{ALC}+\mathbf{T}$, which is an extension of $\mathcal{ALC}$ by a typicality operator $\mathbf{T}$. Given 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 of $\mathcal{L}$, then C and $\mathbf{T}(C)$ are *extended concepts* of $\mathcal{L}$, 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 either a concept or an extended concept $\mathbf{T}(C')$, 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 $a, b \in \mathcal{O}$.

Definition 1 (Semantics of $\mathcal{ALC}+\mathbf{T}$). *A model $\mathcal{M}$ is any structure $\langle \Delta, <, I \rangle$, where Δ is the domain; $<$ is a strict partial order over Δ. For all $S \subseteq \Delta$, we define $Min_<(S) = \{a : a \in S$ and $\nexists b \in S$ s.t. $b < a\}$. We say that $<$ satisfies the* Smoothness Condition *i.e., for all $S \subseteq \Delta$, for all $a \in S$, either $a \in Min_<(S)$ or $\exists b \in Min_<(S)$ such that $b < a$. I is the extension function that maps each extended concept C to $C^I \subseteq \Delta$, and each role R to a $R^I \subseteq \Delta^I \times \Delta^I$. For concepts (built from operators of $\mathcal{ALC}$), C^I is defined in the usual way. For the $\mathbf{T}$ operator: $(\mathbf{T}(C))^I = Min_<(C^I)$. A model satisfying a Knowledge Base (TBox,ABox) is defined as usual. We assume the unique name assumption.*

Notice that the meaning of $\mathbf{T}$ can be split into two parts: for any a of the domain Δ, $a \in (\mathbf{T}(C))^I$ just in case (i) $a \in C^I$, and (ii) there is no $b \in C^I$ such that $b < a$. In order to isolate the second part of the meaning of $\mathbf{T}$ (for the purpose of the calculus that we will present later), we introduce a new modality $\Box$. The basic idea is simply to interpret the preference relation $<$ as an accessibility relation. By the Smoothness Condition, it turns out that $\Box$ has the properties as in 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). The interpretation of $\Box$ in $\mathcal{M}$ is as follows:

Definition 2. $(\Box C)^I = \{a \in \Delta \mid \textit{for every } b \in \Delta, \textit{ if } b < a \textit{ then } b \in C^I\}$

We have that a is a typical instance of C ($a \in (\mathbf{T}(C))^I$) iff $a \in (C \sqcap \Box\neg C)^I$. Since we only use $\Box$ to capture the meaning of $\mathbf{T}$, in the following we will always use the modality $\Box$ followed by a negated concept, as in $\Box\neg C$.

3 The logic $\mathcal{ALC} + \mathbf{T}_{min}$

The logic $\mathcal{ALC} + \mathbf{T}$ allows one to reason about typicality. As a difference with respect to standard $\mathcal{ALC}$, in $\mathcal{ALC} + \mathbf{T}$ we can consistently express, for instance, the fact that three different concepts, as *student*, *working student* and *working student with children*, have a different status as taxpayers. As we have seen in the introduction, this can be consistently expressed by including in a knowledge base the three formulas: $\mathbf{T}(\mathit{Student}) \sqsubseteq \neg \mathit{TaxPayer}$; $\mathbf{T}(\mathit{Student} \sqcap \mathit{Worker}) \sqsubseteq \mathit{TaxPayer}$; $\mathbf{T}(\mathit{Student} \sqcap \mathit{Worker} \sqcap \exists \mathit{HasChild}.\top) \sqsubseteq \neg \mathit{TaxPayer}$. Assume that *john* is an instance of the concept $\mathit{Student} \sqcap \mathit{Worker} \sqcap \exists \mathit{HasChild}.\top$. What can we conclude about *john*? If the ABox contains the assertion $(*)$ $\mathbf{T}(\mathit{Student} \sqcap \mathit{Worker} \sqcap \exists \mathit{HasChild}.\top)(\mathit{john})$, then, in $\mathcal{ALC} + \mathbf{T}$, we can conclude that $\neg \mathit{TaxPayer}(\mathit{john})$. However, in the absence of (*), we cannot derive $\neg \mathit{TaxPayer}(\mathit{john})$.

We would like to infer that individuals are typical instances of the concepts they belong to, if consistent with the KB. In order to maximize the typicality of instances, we define a preference relation on models, and we introduce a semantic entailment determined by minimal models. Informally, we prefer a model $\mathcal{M}$ to a model $\mathcal{N}$ if $\mathcal{M}$ contains more typical instances of concepts than $\mathcal{N}$.

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. 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 $\mathcal{M}^{\Box^-}_{\mathcal{L}_T} = \{(a, \neg\Box\neg C) \mid a \in (\neg\Box\neg C)^I, \text{ with } a \in \Delta, C \in \mathcal{L}_T\}$.
Let KB be a knowledge base and let $\mathcal{L}_T$ be a set of concepts occurring in KB.

Definition 3 (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: $\Delta_\mathcal{M} = \Delta_\mathcal{N}$ and $\mathcal{M}^{\Box^-}_{\mathcal{L}_T} \subset \mathcal{N}^{\Box^-}_{\mathcal{L}_T}$. 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 $\mathcal{M}' <_{\mathcal{L}_T} \mathcal{M}$.*

Definition 4 (Minimal Entailment in $\mathcal{ALC}+\mathbf{T}_{min}$). *A query F (see section below) 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* (see [7]), $\mathcal{ALC}+\mathbf{T}_{min}$ is *nonmonotonic.*

Consider the following example: KB= $\{\mathbf{T}(S) \sqsubseteq \neg P, S(a), W(a)\}$ and $\mathcal{L}_T = \{S, W\}$. We have KB $\models^{\mathcal{L}_T}_{min} \neg P(a)$. Indeed, there is a unique minimal model of KB on the domain $\Delta = \{a\}$, in which a is an instance of $\mathbf{T}(S)$ (as well as an instance of $\mathbf{T}(W)$), and hence $\neg P$ holds in a. Observe that $\neg P(a)$ is obtained in spite of the presence of the irrelevant property $W(a)$.

Consider the knowledge base KB' obtained by adding to KB the formula $\mathbf{T}(S \sqcap W) \sqsubseteq P$ and to $\mathcal{L}_T$ concept $S \sqcap W$. From KB', $\neg P(a)$ is not derivable any more. Instead, we have that KB' $\models^{\mathcal{L}_T}_{min} P(a)$. KB' has a unique minimal model on the domain $\Delta = \{a, b\}$, in which a is an instance of $\mathbf{T}(S \sqcap W)$ and $\mathbf{T}(W)$, but is not an instance of $\mathbf{T}(S)$ (as there is a b, such that $b < a$ and S holds at b). This example shows that, in case of conflict (here, a cannot be both a typical instance of S and $S \sqcap W$), typicality in the more specific concept is preferred.

In general, a knowledge base KB may have no minimal model or more than one minimal model, with respect to a given $\mathcal{L}_T$. The following properties hold:

Proposition 1. *(i) If* KB *has a model, then* KB *has a minimal model with respect to any $\mathcal{L}_T$. (ii) Let us replace all inclusions of the form $\mathbf{T}(C) \sqsubseteq C'$ in* KB *with $C \sqsubseteq C'$, and call* KB' *the resulting knowledge base. If* KB $\models^{\mathcal{L}_T}_{min} F$ *then* KB' $\models_{\mathcal{ALC}+\mathbf{T}} F$*, where F is a query.*

4 Reasoning

In this section we present a tableau calculus for deciding whether a formula (*query*) F is minimally entailed from a knowledge base (TBox,ABox). We introduce a labelled tableau calculus called $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{min}$, which extends the calculus $\mathcal{T}^{\mathcal{ALC}+\mathbf{T}}$ presented in [7], and allows to reason about minimal models.

$\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{min}$ performs a two-phase computation in order to check whether a query F is minimally entailed from the initial KB. In particular, the procedure tries to build an open branch representing a minimal model satisfying KB $\cup \{\neg F\}$. A query F is either a formula of the form $C(a)$ or a subsumption relation $C \sqsubseteq D$ such that, for all $\mathbf{T}(C')$ occurring in F, $C' \in \mathcal{L}_T$. In the first phase, a tableau calculus, called $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$, simply verifies whether KB $\cup\{\neg F\}$ is satisfiable in an $\mathcal{ALC}+\mathbf{T}$ model, building candidate models. In case F has the form $C \sqsubseteq D$, then $\neg F$ corresponds to $(C \sqcap \neg D)(x)$, where x does not occur in KB. In the second phase another tableau calculus, called $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$, checks whether the candidate models found in the first phase are *minimal* models of KB, i.e. for each open branch of the first phase, $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ tries to build a "smaller" model of KB, i.e. a model whose individuals satisfy less formulas $\neg\Box\neg C$ than the corresponding candidate model. The whole procedure $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is described at the end of this section (Definition 8).

$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is based on the notion of a *constraint system*. We consider a set of *variables* drawn from a denumerable set $\mathcal{V}$. $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ makes use of labels, which are denoted with $x, y, z, \ldots$. Labels represent *objects*. An object is 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 $x : C$, 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. A constraint of the form $x \xrightarrow{R} y$ says that the object denoted by label x is related to the object denoted by y by means of role R; a constraint $x : C$ says that the object denoted by x is an instance of the concept C.

Let us now separately analyze the two components of the calculus $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$, starting with $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$.

A tableau of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ 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 $x \xrightarrow{R} y$, whereas 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 from its immediate predecessor $\langle S_{i-1} \mid U_{i-1}\rangle$ by applying a rule of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, 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. A tableau is closed if all its branches are closed.

In order to check the satisfiability of a KB, we build the corresponding constraint system $\langle S \mid U\rangle$, and we check its satisfiability.

Definition 5 (Corresponding constraint system). *Given a knowledge base* KB=(*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 6 (Model satisfying a constraint system). *Let $\mathcal{M}$ be a model as defined in Definition 1. 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 $x : C$ under α if $\alpha(x) \in C^I$ and $x \xrightarrow{R} y$ under α if $(\alpha(x), \alpha(y)) \in R^I$. 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 every constraint in S under α and that, for all $C \sqsubseteq D^L \in U$ and for all x occurring in S, we have that if $\alpha(x) \in C^I$ then $\alpha(x) \in D^I$.*

Proposition 2. *Given a knowledge base, it is satisfiable if and only if its corresponding constraint system is satisfiable.*

To verify the satisfiability of KB $\cup\{\neg F\}$, we use $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ to check the satisfiability of the constraint system $\langle S \mid U\rangle$ obtained by adding the constraint

$$\langle S, x : \neg C, x : C \mid U \rangle \ \text{(Clash)} \qquad \frac{\langle S \mid U, C \sqsubseteq D^L \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \rangle}\text{(Unfold)} \qquad \frac{\langle S, x : \neg\neg C \mid U \rangle}{\langle S, x : C \mid U \rangle}(\neg) \qquad \frac{\langle S, x : \mathbf{T}(C) \mid U \rangle}{\langle S, x : C, x : \Box\neg C \mid U \rangle}(\mathbf{T}^+)$$

if x occurs in S and $x \notin L$

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

If $x : \Box\neg C \notin S$ and $x : \neg\Box\neg C \notin S, C \in \mathcal{L}_T$
x occurs in S

if $y : C \notin S$

$$\frac{\langle S, x : \neg\Box\neg C \mid U \rangle}{\langle S, x : \neg\Box\neg C, y : C, y : \Box\neg C, S^M_{x\to y} \mid U \rangle \quad \langle S, x : \neg\Box\neg C, v_1 : C, v_1 : \Box\neg C, S^M_{x\to v_1} \mid U \rangle \ \cdots \ \langle S, x : \neg\Box\neg C, v_n : C, v_n : \Box\neg C, S^M_{x\to v_n} \mid U \rangle}(\Box^-)$$

If $\nexists u$ s.t. $u : C \in S, u : \Box\neg C \in S,$ and $S^M_{x\to u} \subseteq S$. $\forall v_i$ occurring in $S, x \neq v_i$. y new

$$\frac{\langle S, x : \exists R.C \mid U \rangle}{\langle S, x : \exists R.C, x \xrightarrow{R} y, y : C \mid U \rangle \qquad \langle S, x : \exists R.C, x \xrightarrow{R} v_1, v_1 : C \mid U \rangle \qquad \cdots \qquad \langle S, x : \exists R.C, x \xrightarrow{R} v_n, v_n : C \mid U \rangle}(\exists^+)$$

if $\nexists z \prec x$ s.t. $z \equiv_{S,x:\exists R.C} x$ and $\nexists u$ s.t. $x \xrightarrow{R} u \in S$ and $u : C \in S$
$\forall v_i$ occurring in $S, x \neq v_i$. y new

Fig. 1. The calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$. To save space, we omit the standard rules for $\sqcup$ and $\sqcap$, as well as the rules $(\forall^-)$ and $(\exists^-)$, dual to $(\exists^+)$ and $(\forall^+)$, respectively.

corresponding to $\neg F$ to S', where $\langle S' \mid U \rangle$ is the corresponding constraint system of KB. To this purpose, the rules of the calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ are applied until either a contradiction is generated (Clash) or a model satisfying $\langle S \mid U \rangle$ can be obtained from the resulting constraint system.

As in the calculus proposed 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 the tableau, we add to S the constraint $x : \neg C \sqcup D$ (*unfolding*). As mentioned above, each formula $C \sqsubseteq D$ is equipped with a list L of labels in which it has been unfolded in the current branch. This is needed to avoid multiple unfolding of the same subsumption by using the same label, generating infinite branches.

Before introducing the rules of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ we need some more definitions. First, as in [4], we define an ordering relation $\prec$ to keep track of the temporal ordering of insertion of labels in the tableau, 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 an object 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 represent the same element in the model built by $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$. Last, we define $S^M_{x\to y} = \{y : \neg C, y : \Box\neg C \mid x : \Box\neg C \in S\}$.

The rules of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ are presented in Figure 1. Rules $(\exists^+)$ and $(\Box^-)$ are called *dynamic* since they introduce a new variable in their conclusions. The other rules are called *static*. The side condition on $(\exists^+)$ is introduced in order to ensure a terminating proof search, by implementing the standard *blocking* technique described below. The (*cut*) rule ensures that, given any concept $C \in \mathcal{L}_T$, an open branch built by $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ contains either $x : \Box\neg C$ or $x : \neg\Box\neg C$ for each label x: this is needed in order to allow $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ to check the minimality of the model corresponding to the open branch, as we will discuss later.

The rules of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ 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 ($(\Box^-)$ first) only if no static rule is applicable. This strategy ensures that the labels are considered one at a time according to the ordering $\prec$. The calculus so obtained is sound and complete with respect to the semantics in Definition 6.

Theorem 1 (Soundness and Completeness of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$). *Given a constraint system $\langle S \mid U \rangle$, it is unsatisfiable iff it has a closed tableau in $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$.*

To ensure termination, we adopt the standard loop-checking machinery known as *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 there is no z occurring in S such that $z \prec x$ and $z \equiv_{S,x:\exists R.C} x$. In other words, if there is an "older" label z which is equivalent to x wrt $S, x : \exists R.C$, then $(\exists^+)$ is not applicable, since the condition and the strategy imply that the $(\exists^+)$ rule has already been applied to z. In this case, we say that x is *blocked* by z. By virtue of the properties of $\Box$, no other additional machinery is 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. 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). The (*cut*) rule does not affect termination, since it is applied only to the finitely many formulas belonging to $\mathcal{L}_T$.

Theorem 2 (Termination of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$). *Let $\langle S \mid U \rangle$ be a constraint system, then any tableau generated by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ is finite.*

Since $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ is sound and complete (Theorem 1), and since a KB is satisfiable iff its corresponding constraint system is satisfiable (Proposition 2), from Theorem 2 above it follows that checking whether a given KB (TBox,ABox) is satisfiable is a decidable problem. It is possible to prove that, with the calculus above, the satisfiability of a KB can be decided in nondeterministic exponential time in the size of the KB.

Let us now introduce the calculus $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ which, for each open branch $\mathbf{B}$ built by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, verifies if it is a minimal model of the KB.

Definition 7. *Given an open branch $\mathbf{B}$ of a tableau built from $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, we define: (i) $\mathcal{D}(\mathbf{B})$ as the set of objects occurring on $\mathbf{B}$; (ii) $\mathbf{B}^{\Box^-} = \{x : \neg\Box\neg C \mid x : \neg\Box\neg C$ occurs in $\mathbf{B}\}$.*

A tableau of $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ is a tree whose nodes are triples of the form $\langle S \mid U \mid K \rangle$, where $\langle S \mid U \rangle$ is a constraint system, whereas K contains formulas of the form $x : \neg\Box\neg C$, with $C \in \mathcal{L}_T$.

The basic idea of $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ is as follows. Given an open branch $\mathbf{B}$ built by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ and corresponding to a model $\mathcal{M}^{\mathbf{B}}$ of KB $\cup\{\neg F\}$, $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ checks whether $\mathcal{M}^{\mathbf{B}}$ is a minimal model of KB by trying to build a model of KB which is preferred to $\mathcal{M}^{\mathbf{B}}$. Checking (un)satisfiability of $\langle S \mid U \mid \mathbf{B}^{\Box^-} \rangle$, where $\langle S \mid U \rangle$ is the corresponding constraint system of the initial KB, allows to verify whether the candidate model $\mathcal{M}^{\mathbf{B}}$ is minimal. More in detail, $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ tries to build an open branch containing all the objects appearing on $\mathbf{B}$, i.e. those in

$\langle S, x : C, x : \neg C \mid U \mid K \rangle$ (Clash) $\qquad \langle S \mid U \mid \emptyset \rangle$ (Clash)$_\emptyset$ $\qquad \langle S, x : \neg\Box\neg C \mid U \mid K \rangle$ (Clash)$_{\Box^-}$ if $x : \neg\Box\neg C \notin K$

$$\frac{\langle S \mid U, C \sqsubseteq D^L \mid K \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \mid K \rangle}\ (\text{Unfold}) \quad x \in \mathcal{D}(\mathbf{B}) \text{ and } x \notin L$$

$$\frac{\langle S, x : \exists R.C \mid U \mid K \rangle}{\langle S, x \xrightarrow{R} v_1, v_1 : C \mid U \mid K \rangle \quad \langle S, x \xrightarrow{R} v_2, v_2 : C \mid U \mid K \rangle \cdots \langle S, x \xrightarrow{R} v_n, v_n : C \mid U \mid K \rangle}\ (\exists^+)$$

If $\nexists u \in \mathcal{D}(\mathbf{B})$ s.t. $x \xrightarrow{R} u \in S$ and $u : C \in S$. $\forall v_i \in \mathcal{D}(\mathbf{B})$

$$\frac{\langle S, x : \neg\Box\neg C \mid U \mid K, x : \neg\Box\neg C \rangle}{\langle S, v_1 : C, v_1 : \Box\neg C, S^M_{x \to v_1} \mid U \mid K \rangle \quad \langle S, v_2 : C, v_2 : \Box\neg C, S^M_{x \to v_2} \mid U \mid K \rangle \quad \cdots \quad \langle S, v_n : C, v_n : \Box\neg C, S^M_{x \to v_n} \mid U \mid K \rangle}\ (\Box^-)$$

$\forall v_i \in \mathcal{D}(\mathbf{B}), x \neq v_i$

Fig. 2. The calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$.

$\mathcal{D}(\mathbf{B})$. To this aim, the dynamic rules use labels in $\mathcal{D}(\mathbf{B})$ instead of introducing new ones in their conclusions. The additional set K of a tableau node, initialized with $\mathbf{B}^{\Box^-}$, is used in order to ensure that any branch **B'** built by $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ is preferred to $\mathbf{B}$, that is **B'** only contains negated boxed formulas occurring in $\mathbf{B}$ and there exists at least a $x : \neg\Box\neg C$ that occurs in $\mathbf{B}$ and *does not occur* in **B'**. The rules of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ are shown in Figure 2. $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ also contains analogues of the following rules from $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ in Figure 1: $(\neg)$, $(\mathbf{T}^+)$, $(\mathbf{T}^-)$, (cut), $(\forall^+)$, where the rules in $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ include the additional component K.

More in detail, the rule $(\exists^+)$ is applied to a constraint system containing a formula $x : \exists R.C$; it introduces $x \xrightarrow{R} y$ and $y : C$ where $y \in \mathcal{D}(\mathbf{B})$, instead of y being a new label. The choice of the label y introduces a branching in the tableau construction. The rule (Unfold) is applied in the same way as in $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ to *all the labels of* $\mathcal{D}(\mathbf{B})$ (and not only to those appearing in the branch). The rule $(\Box^-)$ is applied to a node $\langle S, x : \neg\Box\neg C \mid U \mid K \rangle$, when $x : \neg\Box\neg C \in K$, i.e. when the formula $x : \neg\Box\neg C$ also belongs to the open branch $\mathbf{B}$. In this case, the rule introduces a branch on the choice of the individual $v_i \in \mathcal{D}(\mathbf{B})$ which is preferred to x and is such that C and $\Box\neg C$ hold in v_i. In case a tableau node has the form $\langle S, x : \neg\Box\neg C \mid U \mid K \rangle$, and $x : \neg\Box\neg C \notin K$, then $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ detects a clash, called (Clash)$_{\Box^-}$: this corresponds to the situation in which $x : \neg\Box\neg C$ does not belong to $\mathbf{B}$, while $S, x : \neg\Box\neg C$ is satisfiable in a model $\mathcal{M}$ only if $\mathcal{M}$ contains $x : \neg\Box\neg C$, and hence only if $\mathcal{M}$ is *not* preferred to the model represented by $\mathbf{B}$.

The calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ also contains the clash condition (Clash)$_\emptyset$. Since each application of $(\Box^-)$ removes the principal formula $x : \neg\Box\neg C$ from the set K, when K is empty all the negated boxed formulas occurring in $\mathbf{B}$ also belong to the current branch. In this case, the model built by $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ satisfies the same set of negated boxed formulas (for all individuals) as $\mathbf{B}$ and, thus, it is not preferred to the one represented by $\mathbf{B}$.

Theorem 3 (Soundness and completeness of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$). *Given a* KB *and a query F, let $\langle S' \mid U \rangle$ be the corresponding constraint system of* KB*, and $\langle S \mid U \rangle$ the corresponding constraint system of* KB $\cup\{\neg F\}$*. An open branch* $\mathbf{B}$ *built by $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ for $\langle S \mid U \rangle$ is satisfiable by an injective mapping in a minimal model of* KB *iff the tableau in $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-} \rangle$ is closed.*

$\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ always terminates. Indeed, only a finite number of labels can occur on the branch (only those in $\mathcal{D}(\mathbf{B})$ which is finite). Moreover, the side conditions on the application of the rules copying their principal formulas in their conclusion(s) prevent the uncontrolled application of the same rules.

It is possible to show that the problem of verifying that a branch $\mathbf{B}$ represents a minimal model for KB in $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ is in NP in the size of $\mathbf{B}$.

The overall procedure $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is defined as follows:

Definition 8. *Let KB be a knowledge base whose corresponding constraint system is $\langle S \mid U\rangle$. Let F be a query and let S' be the set of constraints obtained by adding to S the constraint corresponding to $\neg F$. The calculus $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ checks whether a query F can be minimally entailed from a* KB *by means of the following procedure:* (phase 1) *the calculus $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ is applied to $\langle S' \mid U\rangle$; if, for each branch $\mathbf{B}$ built by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, either (i) $\mathbf{B}$ is closed or (ii)* (phase 2) *the tableau built by the calculus $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ for $\langle S \mid U \mid \mathbf{B}^{\Box^-}\rangle$ is open, then* KB $\models_{min}^{\mathcal{L}_T} F$*, otherwise* KB $\not\models_{min}^{\mathcal{L}_T} F$*.*

$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is therefore a sound and complete decision procedure for verifying if a formula F can be minimally entailed from a KB. We can also prove that:

Theorem 4 (Complexity of $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$). *The problem of deciding whether* KB $\models_{min}^{\mathcal{L}_T} F$ *is in* CO-NEXP$^{\text{NP}}$.

As an example, let KB contain $\{\mathbf{T}(C) \sqsubseteq \neg P, C(a), D(a)\}$. We show that KB $\models_{min}^{\mathcal{L}_T} \neg P(a)$ with $\mathcal{L}_T = \{C\}$. The tableau $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ is initialised with $\langle S \cup \{a : \neg\neg P\} \mid U\rangle$, where $S = \{a : C, a : D\}$ and $U = \{\mathbf{T}(C) \sqsubseteq \neg P^{\emptyset}\}$. We apply (Unfold), $(\sqcup^+)$ and then the $(\mathbf{T}^-)$ rule. Disregarding the nodes that contain a clash, the only left branch $\mathbf{B}$ contains (after the application of the $(\neg)$ rule) as lowest node: $\langle S' \mid U'\rangle$, where $U' = \{\mathbf{T}(C) \sqsubseteq \neg P^{\{a\}}\}$ and $S' = \{a : C, a : D, a : P, a : \neg\Box\neg C\}$. $\mathbf{B}$ may be further expanded. By applying $(\Box^-)$, (Unfold), $(\sqcup^+)$ and then $(\mathbf{T}^-)$ we can generate only one open extension of $\mathbf{B}$ and it contains: $S'' = \{a : C, a : D, a : P, a : \neg\Box\neg C, b : C, b : \Box\neg C, b : \neg P\}$. We now apply the procedure $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ to $\mathbf{B}$: the tableau is initialised with $\langle S \mid U \mid K\rangle$, where $K = \{a : \neg\Box\neg C\}$; its expansion contains a branch whose lowest node is $\langle S_1 \mid U' \mid K\rangle$, where $S_1 = \{a : C, a : D, a : \Box\neg C, a : \neg P\}$ and U' is as above. This node can be further expanded by applying (Unfold) wrt. $b \in \mathcal{D}(\mathbf{B})$ and $(\mathbf{T}^-)$ obtaining three nodes $\langle S_{1,1} \mid U'' \mid K\rangle$, $\langle S_{1,2} \mid U'' \mid K\rangle$, $\langle S_{1,3} \mid U'' \mid K\rangle$ where $S_{1,1} = S_1 \cup \{b : \neg C\}$, $S_{1,2} = S_1 \cup \{b : \neg\Box\neg C\}$, $S_{1,3} = S_1 \cup \{b : \neg P\}$ and $U'' = \{\mathbf{T}(C) \sqsubseteq \neg P^{\{a,b\}}\}$. The node $\langle S_{1,2} \mid U'' \mid K\rangle$ is closed by $(\text{Clash})_{\Box^-}$, the two others may be expanded by (cut) on $b : (\neg)\Box\neg C$, obtaining finally the following open nodes: $\langle S_{1,1} \cup \{b : \Box\neg C\} \mid U'' \mid K\rangle$ and $\langle S_{1,3} \cup \{b : \Box\neg C\} \mid U'' \mid K\rangle$. Since the two nodes are open, the tableau $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ for $\mathbf{B}$ is open, whence $\mathbf{B}$ is closed. Thus the whole procedure $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ verifies that KB $\models_{min}^{\mathcal{L}_T} \neg P(a)$.

5 Discussion and Conclusion

We have proposed a nonmonotonic extension called $\mathcal{ALC} + \mathbf{T}_{min}$ of $\mathcal{ALC}$ for reasoning about prototypical properties in Description Logic framework. The extension is obtained by adding first a typicality operator, originally defined in

[7], to $\mathcal{ALC}$. This extension, called $\mathcal{ALC} + \mathbf{T}$, provides a monotonic extension of $\mathcal{ALC}$ that enjoys a simple modal semantics. One advantage of the use of a typicality operator is that we can express prototypical properties directly in the form "the most typical instances of concept C have the property P" (corresponding to $\mathbf{T}(C) \sqsubseteq P$), as opposed to rather complicated encodings within other nonmonotonic formalisms. However, $\mathcal{ALC} + \mathbf{T}$ is not sufficient to perform defeasible reasoning. For this reason in the present work we develop a preferential semantics, called $\mathcal{ALC} + \mathbf{T}_{min}$. This nonmonotonic extension allows one to perform defeasible reasoning in particular in the context of inheritance with exceptions to some extent. We have then developed a procedure for deciding query-entailment in $\mathcal{ALC} + \mathbf{T}_{min}$. The procedure has the form of a two-phase tableau calculus for generating $\mathcal{ALC} + \mathbf{T}_{min}$ minimal models. The procedure is sound, complete, and terminating, whereby giving a decision procedure for deciding $\mathcal{ALC} + \mathbf{T}_{min}$ entailment in co-NExp$^{\mathrm{NP}}$.

We plan to extend this work in several directions. First of all, the tableau procedure we have described can be optimised in many ways. For instance, we guess that the calculus $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, dealing with the monotonic logic $\mathcal{ALC} + \mathbf{T}$, can be made more efficient by applying standard techniques such as caching. More precisely, we expect to obtain an Exp decision procedure. Furthermore, the (cut) rule could be applied in $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ by distinguishing if $x : \neg\Box\neg C$ belongs to K or not: in the second case, the branch where $x : \neg\Box\neg C$ is introduced is closed by $(\text{Clash})_{\Box^-}$. Although the calculus $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ provides a decision procedure, we have still to determine the exact complexity of $\mathcal{ALC} + \mathbf{T}_{min}$ itself.

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: $\mathbf{T}(\mathit{Student}) \sqsubseteq \neg \mathit{HasIncome}$, $\mathbf{T}(\mathit{Student}) \sqsubseteq \exists \mathit{Owns}.\mathit{LibraryCard}$, $\mathit{PhDStudent} \sqsubseteq \mathit{Student}$, $\mathbf{T}(\mathit{PhDStudent}) \sqsubseteq \mathit{HasIncome}$. Our semantics does not support the inference $\mathbf{T}(\mathit{PhDStudent}) \sqsubseteq \exists \mathit{Owns}.\mathit{LibraryCard}$, 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 $\mathbf{T}(\mathit{Student}) \sqsubseteq \neg \mathit{HasIncome} \sqcap \exists \mathit{Owns}.\mathit{LibraryCard}$ which is contradicted by $\mathbf{T}(\mathit{PhDStudent}) \sqsubseteq \mathit{HasIncome}$. 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. This problem is perhaps better addressed by probabilistic extensions of description logics such as [8].

KLM logics, which are at the base of our semantics, are related to probabilistic reasoning. In [8], 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 ours in further research.

Moreover, we intend to investigate the precise relation of $\mathcal{ALC} + \mathbf{T}_{min}$ with other formalisms for nonmonotonic reasoning, first of all with circumscription. To this regard, Moinard in [11] has shown that several kinds of preferential entailment (in propositional logic) can be translated into generalised forms of circumscription by extending the vocabulary, as a matter of fact, to an exponentially larger vocabulary. It might be worth investigating if a similar encoding works in our case. More concretely, we may wonder if there is a direct translation of knowledge bases from $\mathcal{ALC}+\mathbf{T}_{min}$ to circumscription and viceversa. As a starting point, concerning the direction from $\mathcal{ALC}+\mathbf{T}_{min}$ to circumscription, we guess that the translation should map every subsumption relation $\mathbf{T}(C_i) \sqsubseteq Q_i$ of a KB into a subsumption relation of the kind $C_i \sqsubseteq Q_i \sqcup Abn_{C_i}$, where, for each concept C_i, Abn_{C_i} is a distinct abnormality concept name to minimize. Then we may ask whether there exists a circumscription pattern [3] CP where (i) all concept names different from Abn_{C_i} and all roles vary and (ii) concept names Abn_{C_i} are minimised according to a suitable partial order (to be discovered), such that the queries entailed by the KB in $\mathcal{ALC} + \mathbf{T}_{min}$ coincide with the queries entailed by the translated KB under circumscription with respect to pattern CP. We shall deal with this and related questions in future research.

References

1. F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180, 1995.
2. F. Baader and B. Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. of Automated Reasoning (JAR)*, 15(1):41–68, 1995.
3. P. A. Bonatti, C. Lutz, and F. Wolter. Description logics with circumscription. In *Proc. of KR*, pages 400–410, 2006.
4. M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. Artif. Int. Research (JAIR)*, 1:109–138, 1993.
5. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
6. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *KR 2004, 141-151.*
7. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Preferential Description Logics. In LPAR 2007. LNCS(LNAI), vol. 4790, pp. 257-272, 2007.
8. R. Giugno and T. Lukasiewicz. P-$\mathcal{SHOQ}$(D): A Probabilistic Extension of $\mathcal{SHOQ}$(D) for Probabilistic Ontologies in the Semantic Web. In JELIA 2002. LNCS(LNAI), vol. 2424, pp. 86-97.
9. P. Ke and U. Sattler. Next Steps for Description Logics of Minimal Knowledge and Negation as Failure. In DL2008, 2008.
10. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
11. Y. Moinard. General Preferential Entailments as Circumscriptions. In ECSQARU 2001. LNCS(LNAI), vol. 2143, pp. 532-543, 2005.
12. U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *Proc. of IJCAI*, pages 676–681, 1993.