Università degli studi di Torino

Tesi di Laurea Magistrale

Thesis title

Candidato:

Luca Violanti

Abstract

Contents

1. Introduction

In this thesis we present the design and implementation of a distributed theorem prover for the non-monotonic description logic $\mathcal{ALC} + \mathbf{T}_{min}$.
Description Logics (DL) are a family of formal knowledge representation languages. They are a decidable fragment of the first-order logic formalism used to provide semantics to representation structures. DLs are used in artificial intelligence for formal reasoning on the concepts of an application domain. It is of particular importance in providing a logical formalism for ontologies and the Semantic Web. The most notable applications outside information science is in bioinformatics and in the codification of medical knowledge. A DL knowledge base (KB) comprises two components: the TBox, containing the definition of concepts (and possibly roles) and a specification of inclusion 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 arise. The traditional approach is to handle defeasible inheritance by integrating some kind of nonmonotonic extension of DLs.

2. Description Logics

While formal, logic-based approaches to representing and working with knowledge occur throughout human history, the advent and widespread adoption of programmable computing devices in the 20th century has led to intensified studies of both theoretical and practical aspects of knowledge representation and automated reasoning. Rooted in early AI approaches, Description Logics (DLs) have developed into one of the main knowledge representation formalisms. The maturity of the field is also reflected by the adoption of description logics as prior specification paradigm for ontological descriptions - culminating in the standardization of the OWL web ontology language by the World Wide Web Consortium (W3C) - as well as the availability of highly optimized and readily deployable (yet open source) tools for automated inferencing. Thanks to this "dissemination path", DLs constitute the theoretical backbone for information systems in many disciplines, among which life sciences can be seen as the "early adopters"[**? ? ?**].

3. Description Logics for typicality

3.1. The logic $\mathcal{ALC} + \mathbf{T}$

In this section, we recall the original $\mathcal{ALC}+\mathbf{T}$, which is an extension of $\mathcal{ALC}$ by a typicality operator $\mathbf{T}$ introduced in [**?**]. Given an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individual constants $\mathcal{O}$, the language $\mathcal{L}$ of the logic $\mathcal{ALC}+\mathbf{T}$ is defined by distinguishing *concepts* and *extended concepts* as follows:

- (Concepts)
  - $A \in \mathcal{C}$, $\top$ and $\bot$ 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 $\mathcal{L}$
  - 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 $C', D \in \mathcal{L}$ are concepts. 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"

Definition 1 (Semantics of T with selection function). A model is any structure

$$\langle \Delta, I, f_{\mathbf{T}} \rangle$$

where:

- Δ is the domain, whose elements are denoted with $x, y, z, \ldots$;
- 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 assigns to each atomic concept $A \in \mathcal{C}$ a set $A^I \subseteq \Delta$ and it is extended to arbitrary extended concepts as follows:
  - $\top^I = \Delta$
  - $\bot^I = \emptyset$
  - $(\neg C)^I = \Delta \backslash C^I$
  - $(C \sqcap D)^I = C^I \cap D^I$
  - $(C \sqcup D)^I = C^I \cup D^I$
  - $(\forall R.C)^I = \{x \in \Delta \mid \forall y.(x, y) \in R^I \rightarrow y \in C^I\}$
  - $(\exists R.C)^I = \{x \in \Delta \mid \exists y.(x, y) \in R^I \text{ and } y \in C^I\}$
  - $(\mathbf{T}(C))^I = f_{\mathbf{T}}(C^I)$
- Given $S \subseteq \Delta$, $f_{\mathbf{T}}$ is a function $f_{\mathbf{T}} : Pow(\Delta) \rightarrow Pow(\Delta)$ satisfying the following properties:
  - $(f_{\mathbf{T}} - 1)$ $f_{\mathbf{T}}(S) \subseteq S$;
  - $(f_{\mathbf{T}} - 2)$ if $S \neq \emptyset$, then also $f_{\mathbf{T}}(S) \neq \emptyset$;
  - $(f_{\mathbf{T}} - 3)$ if $f_{\mathbf{T}}(S) \subseteq R$, then $f_{\mathbf{T}}(S) = f_{\mathbf{T}}(S \cap R)$;
  - $(f_{\mathbf{T}} - 4)$ $f_{\mathbf{T}}(\bigcup S_i) \subseteq \bigcup f_{\mathbf{T}}(S_i)$;
  - $(f_{\mathbf{T}} - 5)$ $\bigcap f_{\mathbf{T}}(S_i) \subseteq f_{\mathbf{T}}(\bigcup S_i)$.

Intuitively, given the extension of some concept C, the selection function $f_{\mathbf{T}}$ 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 $\bigcap f_{\mathbf{T}}(S_i) \subseteq f_{\mathbf{T}}(\bigcap S_i)$ holds. $(f_{\mathbf{T}} - 5)$ 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.

In [?], we have shown that one can give an equivalent, alternative semantics for **T** based on a *preference relation* semantics rather than on a selection function semantics. The idea is that there is a global, irreflexive and transitive 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. For this reason, we cannot express the fact that y is more typical than x with respect to concept C, whereas x is more typical than y with respect to another concept D. All what we can say is that either x is incomparable with y or x is more typical than y or y is more typical than x. 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 given two element which one of the two is more typical. Following KLM, the preference relation also satisfies a *Smoothness Condition*, which is related to the well known *Limit Assumption* in Conditional Logics [?] [1]; this condition ensures that, if the extension C^I of a concept C is not empty, then there is at least one *minimal* element of C^I. This is stated in a rigorous manner in the following definition:

Definition 2. Given an irreflexive and transitive relation $<$ over a domain Δ, called *preference relation*, for all $S \subseteq \Delta$, we define

$$Min_<(S) = \{x \in S \mid \nexists y \in S \text{ s.t. } y < x\}$$

We say that $<$ satisfies the *Smoothness Condition* if for all $S \subseteq \Delta$, for all $x \in S$, either $x \in Min_<(S)$ or $\exists y \in Min_<(S)$ such that $y < x$.

The following representation theorem is proved in [?]:

Theorem 1 (Theorem 2.1 in [?])**.** *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 there exists an irreflexive and transitive relation $<$ on Δ, satisfying the Smoothness Condition, such that for all $S \subseteq \Delta$, $f_{\mathbf{T}}(S) = Min_<(S)$.*

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

Definition 3 (Semantics of $\mathcal{ALC} + \mathbf{T}$). A model $\mathcal{M}$ of $\mathcal{ALC} + \mathbf{T}$ is any structure

$$\langle \Delta, I, < \rangle$$

where:

- Δ is the domain;
- $<$ is an irreflexive and transitive relation over Δ satisfying the Smoothness Condition (Definition **??**)
- 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 assigns to each atomic concept $A \in \mathcal{C}$ a set $A^I \subseteq \Delta$. Furthermore, I is extended as in Definition **??** with the exception of $(\mathbf{T}(C))^I$, which is defined as
$$(\mathbf{T}(C))^I = Min_<(C^I).$$

[1] More precisely, the Limit Assumption entails the Smoothness Condition (i.e. that there are no infinite $<$ descending chains). Both properties come for free in finite models.

Let us now introduce the notion of satisfiability of an $\mathcal{ALC}+\mathbf{T}$ knowledge base. In order to define the semantics of the assertions of the ABox, we extend the function I to individual constants; we assign to each individual constant $a \in \mathcal{O}$ a *distinct* domain element $a^I \in \Delta$, that is to say we enforce the *unique name assumption*. As usual, the adoption of the unique name assumption greatly simplifies reasoning about prototypical properties of individuals denoted by different individual constants. Considering the example of department staff having lunches, if (in addition to the TBox) the ABox only contains the following facts about Greg and Sara:

$DepartmentMember(greg)$
$DepartmentMember(sara)$, $TemporaryWorker(sara)$

we would like to infer that Greg takes his lunches at the restaurant, whereas Sara does not; but without the unique name hypothesis, we cannot get this conclusion since Greg and Sara might be the same individual. To perform useful reasoning we would need to extend the language with equality and make a case analysis according to possible identities of individuals. While this is technically possible, we prefer to keep the things simple here by adopting the unique name assumption.

Definition 4 (Model satisfying a Knowledge Base). Consider a model $\mathcal{M}$, as defined in Definition **??**. We extend I so that it assigns to each individual constant a of $\mathcal{O}$ an element $a^I \in \Delta$, and I satisfies the unique name assumption. Given a KB (TBox,ABox), we say that:

- $\mathcal{M}$ satisfies TBox iff for all inclusions $C \sqsubseteq D$ in TBox, $C^I \subseteq D^I$.
- $\mathcal{M}$ satisfies ABox iff: (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. Last, a query F is entailed by KB in $\mathcal{ALC}+\mathbf{T}$ if it holds in all models satisfying *KB*. In this case we write $KB \models_{\mathcal{ALC}+\mathbf{T}} F$.

Notice that the meaning of $\mathbf{T}$ can be split into two parts: for any 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 $y < x$. As already mentioned in the Introduction, in order to isolate the second part of the meaning of $\mathbf{T}$ (for the purpose of the calculus that we will present in Section **??**), 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 avoiding 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 5. Given a model $\mathcal{M}$ as in Definition **??**, we extend the definition of I with the following clause:

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

It is easy to observe that x is a typical instance of C if and only if it is an instance of C and $\Box\neg C$, that is to say:

Proposition 1. *Given a model $\mathcal{M}$ as in Definition **??**, given a concept C and an element $x \in \Delta$, we have that*

$$x \in (\mathbf{T}(C))^I \text{ iff } x \in (C \sqcap \Box\neg C)^I$$

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 $\Box\neg C$.
The Smoothness condition, together with the transitivity of $<$, ensures the following Lemma:

Lemma 1. *Given an $\mathcal{ALC}+\mathbf{T}$ model as in Definition **??**, an extended concept C, and an element $x \in \Delta$, if there exists $y < x$ such that $y \in C^I$, then either $y \in Min_<(C^I)$ or there is $z < x$ such that $z \in Min_<(C^I)$.*

Last, we state a theorem which will be used in the following:

Theorem 2 (Finite model property of $\mathcal{ALC}+\mathbf{T}$)**.** *The logic $\mathcal{ALC}+\mathbf{T}$ has the finite model property.*

3.2. The logic $\mathcal{ALC}+\mathbf{T}_{\text{min}}$

As mentioned in the Introduction, the logic $\mathcal{ALC}+\mathbf{T}$ presented in [**?**] allows 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, like *Department member*, *Temporary Department Member* and *Temporary Department member having restaurant tickets*, have a different status with respect to *Have lunch at a restaurant.* This can be consistently expressed by including in a knowledge base the three formulas:

$\mathbf{T}(\mathit{DepartmentMember}) \sqsubseteq \mathit{LunchAtRestaurant}$
$\mathbf{T}(\mathit{DepartmentMember} \sqcap \mathit{TemporaryResearcher}) \sqsubseteq \neg\mathit{LunchAtRestaurant}$
$\mathbf{T}(\mathit{DepartmentMember} \sqcap \mathit{TemporaryResearcher} \sqcap \exists \mathit{Owns}.\mathit{RestaurantTicket}) \sqsubseteq$
$\mathit{LunchAtRestaurant}$

Assume that *greg* is an instance of the concept $\mathit{DepartmentMember} \sqcap \mathit{TemporaryResearcher} \sqcap \exists \mathit{Owns}.\mathit{RestaurantTicket}$. What can we conclude about *greg*? We have already mentioned that if the ABox explicitly points out that *greg* is a *typical* instance of the concept, and it contains the assertion that:

$(*)$ $\mathbf{T}(\mathit{DepartmentMember} \sqcap \mathit{TemporaryResearcher} \sqcap \exists \mathit{Owns}.\mathit{RestaurantTicket})(\mathit{greg})$,

then, in $\mathcal{ALC}+\mathbf{T}$, we can conclude that

$$\mathit{LunchAtRestaurant}(\mathit{greg}).$$

However, if (*) is replaced by the weaker

$(**)$ $(\mathit{DepartmentMember} \sqcap \mathit{TemporaryResearcher} \sqcap \exists \mathit{Owns}.\mathit{RestaurantTicket})(\mathit{greg})$,

in which there is no information about the typicality of *greg*, in $\mathcal{ALC}+\mathbf{T}$ we can no longer draw this conclusion, and indeed we cannot make any inference about whether *greg* spends its lunch time at a restaurant or not. The limitation here lies in the fact that $\mathcal{ALC}+\mathbf{T}$ is *monotonic*, whereas we would like to make a non-monotonic inference. Indeed, we would like to non-monotonically assume, in the absence of information to the contrary, that *greg*

is a typical instance of the concept. In general, we would like to infer that individuals are typical instances of the concepts they belong to, if this is consistent with the KB.
As a difference with respect to $\mathcal{ALC} + \mathbf{T}$, $\mathcal{ALC} + \mathbf{T}_{min}$ is *non-monotonic*, and it allows to make this kind of inference. Indeed, in $\mathcal{ALC} + \mathbf{T}_{min}$ if (**) is all the information about *greg* present in the ABox, we can derive that *greg* is a typical instance of the concept, and from the inclusions above we conclude that *LunchAtRestaurant*(*greg*). We have already mentioned that we obtain this non-monotonic behaviour by restricting our attention to the minimal $\mathcal{ALC} + \mathbf{T}$ models. As a difference with respect to $\mathcal{ALC} + \mathbf{T}$, in order to determine what is entailed by a given knowledge base KB, we do not consider *all* models of KB but only the *minimal* ones. These are the models that minimize the number of atypical instances of concepts.
Given a KB, we consider a finite set $\mathcal{L}_{\mathbf{T}}$ of concepts occurring in the KB: these are the concepts for which we want to minimize the atypical instances. The minimization of the set of atypical instances will apply to individuals explicitly occurring in the ABox as well as to implicit individuals. We assume that the set $\mathcal{L}_{\mathbf{T}}$ contains at least all concepts C such that $\mathbf{T}(C)$ occurs in the KB. Notice that in case $\mathcal{L}_{\mathbf{T}}$ contains more concepts than those occurring in the scope of $\mathbf{T}$ in KB, the atypical instances of these concepts will be minimized but no extra properties will be inferred for the typical instances of the concepts, since the KB does not say anything about these instances.
We have seen that $(\mathbf{T}(C))^I = (C \sqcap \Box\neg C)^I$: x is a typical instance of a concept C ($x \in (\mathbf{T}(C))^I$) when it is an instance of C and there is no other instance of C preferred to x, i.e. $x \in (C \sqcap \Box\neg C)^I$. By contraposition an instance of C is atypical if $x \in (\neg\Box\neg C)^I$ therefore in order to minimize the atypical instances of C, we minimize the instances of $\neg\Box\neg C$. Notice that this is different from maximizing the instances of $\mathbf{T}(C)$. We have adopted this solution since it allows to maximize the set of typical instances of C without affecting the extension C^I of C (whereas maximizing the extension of $\mathbf{T}(C)$ would imply maximizing also the extension of C).
We define the set $\mathcal{M}^{\Box^-}_{\mathcal{L}_{\mathbf{T}}}$ of negated boxed formulas holding in a model, relative to the concepts in $\mathcal{L}_{\mathbf{T}}$:

Definition 6. Given a model $\mathcal{M} = \langle \Delta, I, < \rangle$ and a set of concepts $\mathcal{L}_{\mathbf{T}}$, we define

$$\mathcal{M}^{\Box^-}_{\mathcal{L}_{\mathbf{T}}} = \{(x, \neg\Box\neg C) \mid x \in (\neg\Box\neg C)^I, \text{ with } x \in \Delta, C \in \mathcal{L}_{\mathbf{T}}\}$$

Let KB be a knowledge base and let $\mathcal{L}_{\mathbf{T}}$ be a set of concepts occurring in KB.

Definition 7 (Preferred and minimal models). Given a model $\mathcal{M} = \langle \Delta_{\mathcal{M}}, I_{\mathcal{M}}, <_{\mathcal{M}} \rangle$ of KB and a model $\mathcal{N} = \langle \Delta_{\mathcal{N}}, I_{\mathcal{N}}, <_{\mathcal{N}} \rangle$ of KB, we say that $\mathcal{M}$ is preferred to $\mathcal{N}$ with respect to $\mathcal{L}_{\mathbf{T}}$, and we write $\mathcal{M} <_{\mathcal{L}_{\mathbf{T}}} \mathcal{N}$, if the following conditions hold:

- $\Delta_{\mathcal{M}} = \Delta_{\mathcal{N}}$
- $a^{I_{\mathcal{M}}} = a^{I_{\mathcal{N}}}$ for all individual constants $a \in \mathcal{O}$
- $\mathcal{M}^{\Box^-}_{\mathcal{L}_{\mathbf{T}}} \subset \mathcal{N}^{\Box^-}_{\mathcal{L}_{\mathbf{T}}}$.

A model $\mathcal{M}$ is a *minimal model* for KB (with respect to $\mathcal{L}_{\mathbf{T}}$) if it is a model of KB and there is no a model $\mathcal{M}'$ of KB such that $\mathcal{M}' <_{\mathcal{L}_{\mathbf{T}}} \mathcal{M}$.

Given the notion of preferred and minimal models above, we introduce a notion of *minimal entailment*, that is to say we restrict our consideration to minimal models only. First of

all, we introduce the notion of *query*, which can be minimally entailed from a given KB. A query F is a formula of the form $C(a)$ where C is an extended concept and $a \in \mathcal{O}$. We assume that, for all $\mathbf{T}(C')$ occurring in F, $C' \in \mathcal{L}_{\mathbf{T}}$. Given a KB and a model $\mathcal{M} = \langle \Delta, I, < \rangle$ satisfying it, we say that a query $C(a)$ holds in $\mathcal{M}$ if $a^I \in C^I$.
Let us now define minimal entailment of a query in $\mathcal{ALC} + \mathbf{T}_{min}$. In Section **??** we will reduce the other standard reasoning tasks to minimal entailment.

Definition 8 (Minimal Entailment in $\mathcal{ALC} + \mathbf{T}_{min}$). A query F is minimally entailed from a knowledge base *KB* with respect to $\mathcal{L}_{\mathbf{T}}$ if it holds in all models of *KB* that are minimal with respect to $\mathcal{L}_{\mathbf{T}}$. We write $KB \models_{min}^{\mathcal{L}_{\mathbf{T}}} F$.

The non-monotonic character of $\mathcal{ALC} + \mathbf{T}_{min}$ also allows to deal with the following examples.

Example 1. Consider the following KB:

$$\text{KB } = \{\mathbf{T}(\mathit{Athlet}) \sqsubseteq \mathit{Confident}, \mathit{Athlet}(\mathit{john}), \mathit{Finnish}(\mathit{john})\}$$

and $\mathcal{L}_{\mathbf{T}} = \{\mathit{Athlet}, \mathit{Finnish}\}$. We have

$$\text{KB } \models_{min}^{\mathcal{L}_{\mathbf{T}}} \mathit{Confident}(\mathit{john})$$

Indeed, there is no minimal model of KB that contains a non typical instance of some concept (indeed in all minimal models of KB the relation $<$ is empty). Hence *john* is an instance of $\mathbf{T}(\mathit{Athlet})$ (it can be easily verified that any model in which *john* is not an instance of $\mathbf{T}(\mathit{Athlet})$ is not minimal). By KB, in all these models, *john* is an instance of *Confident*. Observe that *Confident*(*john*) is obtained, in spite of the presence of the irrelevant assertion *Finnish*(*john*).

Example 2. Consider now the knowledge base KB' obtained by adding to KB the formula $\mathbf{T}(\mathit{Athlet} \sqcap \mathit{Finnish}) \sqsubseteq \neg \mathit{Confident}$, that is to say:

$$\text{KB'} = \{\mathbf{T}(\mathit{Athlet}) \sqsubseteq \mathit{Confident}, \mathbf{T}(\mathit{Athlet} \sqcap \mathit{Finnish}) \sqsubseteq \neg\mathit{Confident}, \mathit{Athlet}(\mathit{john}), \mathit{Finnish}(\mathit{john})\}$$

and to $\mathcal{L}_{\mathbf{T}}$ concept $\mathit{Athlet} \sqcap \mathit{Finnish}$. From KB', *Confident*(*john*) is no longer derivable. Instead, we have that

$$\text{KB}' \models_{min}^{\mathcal{L}_{\mathbf{T}}} \neg\mathit{Confident}(\mathit{john}).$$

Indeed, by reasoning as above it can be shown that in all the minimal models of KB', *john* is an instance of $\mathbf{T}(\mathit{Athlet} \sqcap \mathit{Finnish})$, and it is no longer an instance of $\mathbf{T}(\mathit{Athlet})$. This example shows that, in case of conflict (here, *john* cannot be both a typical instance of *Athlet* and of $\mathit{Athlet} \sqcap \mathit{Finnish}$), 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 $\mathcal{L}_{\mathbf{T}}$. The following property holds.

Proposition 2. *If* KB *has a model, then* KB *has a minimal model with respect to any* $\mathcal{L}_{\mathbf{T}}$.

The above fact is a consequence of the *finite model property* of the logic $\mathcal{ALC}+\mathbf{T}$ (Theorem **??**).

4. A tableaux calculus for $\mathcal{ALC} + \mathbf{T}_{min}$

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

$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ 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 $\cup \{\neg F\}$.

In the first phase, a tableau calculus, called $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, simply verifies whether KB $\cup \{\neg F\}$ is satisfiable in an $\mathcal{ALC} + \mathbf{T}$ model, building candidate models. In the second phase another tableau calculus, called $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$, checks whether the candidate models found in the first phase are *minimal* models of KB. To this purpose for each open branch of the first phase, $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ 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 formally defined at the end of this section (Definition **??**).

$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is based on the notion of a *constraint system.* We consider a set of *variables* drawn from a denumerable set $\mathcal{V}$. Variables are used to represent individuals not explicitly mentioned in the ABox, that is to say implicitly expressed by existential as well as universal restrictions.

$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ makes use of labels, which are denoted with $x, y, z, \ldots$. A label represents either a variable or an individual constant occurring in the ABox, that is to say an element of $\mathcal{O} \cup \mathcal{V}$.

Definition 9 (Constraint). A *constraint* (or *labelled* formula) is a syntactic entity of the form either $x \xrightarrow{R} y$ or $y < x$ or $x : C$, where x, y are labels, 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.

Intuitively, a constraint of the form $x \xrightarrow{R} y$ says that the individual represented by label x is related to the one denoted by y by means of role R; a constraint $y < x$ says that the individual denoted by y is "preferred" to the individual represented by x with respect to the relation $<$; a constraint $x : C$ says that the individual denoted by x is an instance of the concept C, i.e. it belongs to the extension C^I. As we will define in Definition **??**, the ABox of a 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 $a \xrightarrow{R} b$.

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

4.1. The tableau calculus $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$

Let us first define the basic notions of a tableau system in $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$:

Definition 10 (Tableau of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$). A tableau of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ is a tree whose nodes are constraint systems, i.e., pairs $\langle S \mid U\rangle$, where S is a set of constraints, 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[2]. 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 $\langle S_{i-1} \mid U_{i-1}\rangle$

[2] As we will discuss later, this list is used in order to ensure the termination of the tableau calculus.

by applying a rule of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ (see Figure **??**), 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* (either (Clash) or $(\text{Clash})_\top$ or $(\text{Clash})_\perp$), 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 11 (Corresponding constraint system). Given a knowledge base KB=(TBox,ABox), we define its *corresponding constraint system* $\langle S \mid U \rangle$ as follows:

- $S = \{a : C \mid C(a) \in \text{ABox}\} \cup \{a \xrightarrow{R} b \mid aRb \in \text{ABox}\}$
- $U = \{C \sqsubseteq D^{\emptyset} \mid C \sqsubseteq D \in \text{TBox}\}$

Definition 12 (Model satisfying a constraint system). Let $\mathcal{M} = \langle \Delta, I, < \rangle$ be a model as defined in Definition **??**. We define a function α which assigns to each variable of $\mathcal{V}$ an element of Δ, and assigns every individual constant $a \in \mathcal{O}$ to $a^I \in \Delta$. $\mathcal{M}$ satisfies a constraint F under α, written $\mathcal{M} \models_\alpha F$, as follows:

- $\mathcal{M} \models_\alpha x : C$ if and only if $\alpha(x) \in C^I$
- $\mathcal{M} \models_\alpha x \xrightarrow{R} y$ if and only if $(\alpha(x), \alpha(y)) \in R^I$
- $\mathcal{M} \models_\alpha y < x$ if and only if $\alpha(y) < \alpha(x)$

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 every constraint in S under α and that, for all $C \sqsubseteq D^L \in U$ and for all $x \in \Delta$, we have that if $x \in C^I$ then $x \in D^I$.

Let us now show that:

Proposition 3. *KB=(TBox,ABox) is satisfiable in an $\mathcal{ALC} + \mathbf{T}$ model if and only if its corresponding constraint system $\langle S \mid U \rangle$ is satisfiable in the same model.*

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 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 $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ 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 [**?**], 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 $x : \neg C \sqcup D$: we refer to this mechanism as *subsumption expansion*. As mentioned above, each subsumption $C \sqsubseteq D$ is equipped with a list L of labels in which the subsumption has been expanded in the current branch. This is needed to avoid multiple expansions of the same subsumption by using the same label, generating infinite branches.
Before introducing the rules of $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$ we need some more definitions. First, as in [**?**], 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. Moreover, we need to define the *equivalence* between two labels: intuitively, two labels x and y are equivalent if they label the same set of extended concepts. This notion is stated in the following definition, and it is used in order to apply the blocking machinery described in the following, based on the fact that equivalent labels represent the same element in the model built by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$.

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

$$\frac{\langle S, x : C \sqcap D \mid U \rangle}{\langle S, x : C \sqcap D, x : C, x : D \mid U \rangle}\ (\sqcap^+)$$
if $\{x : C, x : D\} \not\subseteq S$

$$\frac{\langle S, x : \neg(C \sqcap D) \mid U \rangle}{\langle S, x : \neg(C \sqcap D), x : \neg C \mid U \rangle \qquad \langle S, x : \neg(C \sqcap D), x : \neg D \mid U \rangle}\ (\sqcap^-)$$
if $x : \neg C \notin S$ and $x : \neg D \notin S$

$$\frac{\langle S, x : C \sqcup D \mid U \rangle}{\langle S, x : C \sqcup D, x : C \mid U \rangle \qquad \langle S, x : C \sqcup D, x : D \mid U \rangle}\ (\sqcup^+)$$
if $x : C \notin S$ and $x : D \notin S$

$$\frac{\langle S, x : \neg(C \sqcup D) \mid U \rangle}{\langle S, x : \neg(C \sqcup D), x : \neg C, x : \neg D \mid U \rangle}\ (\sqcup^-)$$
if $\{x : \neg C, x : \neg D\} \not\subseteq S$

$$\frac{\langle S, x : \neg\neg C \mid U \rangle}{\langle S, x : \neg\neg C, x : C \mid U \rangle}\ (\neg)$$
if $x : C \notin S$

$$\frac{\langle S, x : \mathbf{T}(C) \mid U \rangle}{\langle S, x : \mathbf{T}(C), x : C, x : \Box\neg C \mid U \rangle}\ (\mathbf{T}^+)$$
if $\{x : C, x : \Box\neg C\} \not\subseteq S$

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

$$\frac{\langle S \mid U \rangle}{\langle S, x : \Box\neg C \mid U \rangle \qquad \langle S, x : \neg\Box\neg C \mid U \rangle}\ (cut)$$
if $x : \neg\Box\neg C \notin S$ and $x : \Box\neg C \notin S$
$C \in \mathcal{L}_{\mathbf{T}}$
x occurs in S

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

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

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

$$\frac{\langle S, x : \neg\Box\neg C \mid U \rangle}{\langle S, x : \neg\Box\neg C, y < x, y : C, y : \Box\neg C, S^M_{x\to y} \mid U \rangle \ \langle S, x : \neg\Box\neg C, v_1 < x, v_1 : C, v_1 : \Box\neg C, S^M_{x\to v_1} \mid U \rangle \cdots \langle S, x : \neg\Box\neg C, v_n < x, v_n : C, v_n : \Box\neg C, S^M_{x\to v_n} \mid U \rangle}\ (\Box^-)$$
y new
if $\nexists z \prec x$ s.t. $z \equiv_{S, x:\neg\Box\neg C} x$ and $\nexists u$ s.t. $\{u < x, u : C, u : \Box\neg C, S^M_{x\to u}\} \subseteq S$
$\forall v_i$ occurring in $S, x \neq v_i$

Figure 1: The calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$. To save space, we omit the rules $(\forall^-)$ and $(\exists^-)$, dual to $(\exists^+)$ and $(\forall^+)$, respectively.

Definition 13. 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.

$$\sigma(\langle S \mid U \rangle, x) = \sigma(\langle S \mid U \rangle, y).$$

Last, we define the set of formulas $S^M_{x\to y}$, that will be used in the rule $(\Box^-)$ when $y < x$, in order to introduce $y : \neg C$ and $y : \Box\neg C$ for each $x : \Box\neg C$ in the current branch:

Definition 14. Given a tableau node $\langle S \mid U \rangle$ and two labels x and y, 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 **??**. Rules $(\exists^+)$ and $(\Box^-)$ are called *dynamic* since they introduce a new variable in their conclusions. The other rules are called *static*. A brief explanation of the rules follows:

- (Clash), $\text{(Clash)}_\top$ and $\text{(Clash)}_\bot$ are used to detect *clashes*, i.e. unsatisfiable constraint systems;

- rules for $\sqcup$, $\sqcap$, $\neg$, and $\forall$ are similar to the corresponding ones in the tableau calculus for standard $\mathcal{ALC}$ [**?**]: as an example, the rule $(\sqcup^+)$ is applied to a constraint system of the form $\langle S, x : C \sqcup D \mid U\rangle$ in order to deal with the constraint $x : C \sqcup D$ introducing two branches in the tableau construction, to check the two conclusions obtained by adding the constraints $x : C$ and $x : D$, respectively. The side condition of the rules are the usual conditions needed to avoid multiple applications on the same principal formula: concerning the example of $(\sqcup^+)$, it can be applied only if $x : C \notin S$ and $x : D \notin S$;

- the rules $(\mathbf{T}^+)$ and $(\mathbf{T}^-)$ are used to "translate" formulas of the form $\mathbf{T}(C)$ in the corresponding modal interpretation: for $(\mathbf{T}^+)$, this corresponds to introduce $x : C \sqcap \Box\neg C$ to a constraint system containing $x : \mathbf{T}(C)$, whereas for $(\mathbf{T}^-)$ a branching is introduced to add either $x : \neg C$ or $x : \neg\Box\neg C$ in case $x : \neg\mathbf{T}(C)$ belongs to the constraint system;

- the rule $(\sqsubseteq)$ is used in order to check whether, for all x belonging to a branch, the inclusion relations of the TBox are satisfied: given a label x and an inclusion $C \sqsubseteq D^L \in U$, the branching introduced by the rule ensures that either $x : \neg C$ holds or that $x : D$ holds;

- the rule $(\Box^-)$, applied to a principal formula $x : \neg\Box\neg C$ (x is not a typical instance of the concept C, i.e. there exists an element z which is a typical instance of C and is more normal than x), introduces the constraints $z < x$, $z : C$ and $z : \Box\neg C$. A branching on the choice of the label z to use is introduced, since it can be either a "new" label y, not occurring in the branch, or one of the labels $v_1, v_2, \ldots, v_n$ already belonging to the branch. 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 $S^M_{x\rightarrow y}$ of $(\Box^-)$. $(\exists^+)$ deals with constraints of the form $x : \exists R.C$ in a similar way. The additional side conditions on $(\exists^+)$ and $(\Box^-)$ are introduced in order to ensure a terminating proof search, by implementing the standard *blocking* technique described below. Intuitively, they are applied to constraints $x : \exists R.C$ and $x : \neg\Box\neg C$, respectively, only if x *is not blocked*, i.e. if there is no label (*witness*) z, labelling the same concepts of x, such that the rule has been already applied to $z : \exists R.C$ (resp. $z : \neg\Box\neg C$). This is formally stated in Definition **??** below;

- the (*cut*) rule ensures that, given any concept $C \in \mathcal{L}_{\mathbf{T}}$, 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.

All the rules of the calculus copy their principal formulas, i.e. the formulas to which the rules are applied, in all their conclusions. As we will discuss later, for the rules $(\exists^+)$, $(\forall^-)$ and $(\Box^-)$ this is used in order to apply the blocking technique, whereas for the rules $(\exists^-)$, $(\forall^+)$, $(\sqsubseteq)$, and (*cut*) this is needed in order to have a complete calculus. Rules for $\sqcap$, $\sqcup$, $\neg$, and $\mathbf{T}$ also copy their principal formulas in their conclusions for uniformity's sake.
In order to ensure the completeness of the calculus, 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 only if no static rule is applicable.

The calculus so obtained is sound and complete with respect to the semantics in Definition **??**.

Definition 15 (Witness and Blocked label). 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 **??**, we can prove the following Lemma:

Lemma 2. *In any constraint system $\langle S \mid U \rangle$, if x is blocked, then it has exactly one witness.*

As mentioned above, we apply a standard *blocking* technique to control the application of the rules $(\exists^+)$ and $(\Box^-)$, in order to ensure the termination of the calculus. Intuitively, we can apply $(\exists^+)$ to a constraint system of the form $\langle S, x : \exists R.C \mid U \rangle$ only if *x is not blocked*, i.e. it does not have any witness: indeed, in case x has a witness z, by the strategy on the application of the rules described above the rule $(\exists^+)$ has already been applied to some $z : \exists R.C$, and we do not need a further application to $x : \exists R.C$. This is ensured by the side condition on the application of $(\exists^+)$, namely if $\nexists z \prec x$ such that $z \equiv_{S,x:\exists R.C} x$. The same blocking machinery is used to control the application of $(\Box^-)$, which can be applied only if $\nexists z \prec x$ such that $z \equiv_{S,x:\neg\Box\neg C} x$.

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

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 $\mathcal{M}^{\mathbf{B}}$ is a minimal model of the KB.

Definition 16. Given an open branch $\mathbf{B}$ of a tableau built by $\mathcal{TAB}_{PH1}^{\mathcal{ALC}+\mathbf{T}}$, we define:

- $\mathcal{D}(\mathbf{B})$ as the set of labels occurring on $\mathbf{B}$;
- $\mathbf{B}^{\Box^-} = \{x : \neg\Box\neg C \mid x : \neg\Box\neg C \text{ occurs in } \mathbf{B}\}$.

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 $C \in \mathcal{L}_{\mathbf{T}}$.
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 $\mathcal{M}^{\mathbf{B}}$. Starting from $\langle S \mid U \mid \mathbf{B}^{\Box^-} \rangle$ where $\langle S \mid U \rangle$ is the constraint system corresponding to the initial KB $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ tries to build an open branch containing all and only the labels appearing on $\mathbf{B}$, i.e. those in $\mathcal{D}(\mathbf{B})$, and containing less negated boxed formulas than $\mathbf{B}$ does. To this aim, first the dynamic rules use labels in $\mathcal{D}(\mathbf{B})$ instead of introducing new ones in their conclusions. Second the negated boxed formulas used in $\mathbf{B}$ are stored in the additional set K of a tableau node, initialized with $\mathbf{B}^{\Box^-}$. A branch built by $\mathcal{TAB}_{PH2}^{\mathcal{ALC}+\mathbf{T}}$ closes if it does not represent a model preferred to the candidate model $\mathcal{M}^{\mathbf{B}}$, and this happens if the branch contains a contradiction (Clash) or it contains at least all the negated boxed formulas contained in $\mathbf{B}$ ((Clash)$_{\Box^-}$ and (Clash)$_{\emptyset}$).

$$\langle S, x : C, x : \neg C \mid U \mid K \rangle \ \text{(Clash)} \qquad \langle S, x : \bot \mid U \mid K \rangle \ \text{(Clash)}_{\bot} \qquad \langle S, x : \neg \top \mid U \mid K \rangle \ \text{(Clash)}_{\top} \qquad \langle S \mid U \mid \emptyset \rangle \ \text{(Clash)}_{\emptyset}$$

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

$$\frac{\langle S, x : \neg (C \sqcap D) \mid U \mid K \rangle}{\langle S, x : \neg C \mid U \mid K \rangle \qquad \langle S, x : \neg D \mid U \mid K \rangle} (\sqcap^-) \qquad \frac{\langle S, x : C \sqcap D \mid U \mid K \rangle}{\langle S, x : C, x : D \mid U \mid K \rangle} (\sqcap^+) \qquad \frac{\langle S, x : \neg\neg C \mid U \mid K \rangle}{\langle S, x : C \mid U \mid K \rangle} (\neg)$$

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

$$\frac{\langle S \mid U, C \sqsubseteq D^L \mid K \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \mid K \rangle} (\sqsubseteq) \quad x \in \mathcal{D}(\mathbf{B}) \text{ and } x \notin L \qquad \frac{\langle S, x : \forall R.C, x \xrightarrow{R} y \mid U \mid K \rangle}{\langle S, x : \forall R.C, x \xrightarrow{R} y, y : C \mid U \mid K \rangle} (\forall^+) \quad \text{if } y : C \notin S$$

$$\frac{\langle S, x : \neg \Box \neg C \mid U \mid K, x : \neg \Box \neg C \rangle}{\langle S, v_1 : C, v_1 : \Box \neg C, S^M_{x \to v_1}, x : \neg \Box \neg C \mid U \mid K \rangle \quad \langle S, v_2 : C, v_2 : \Box \neg C, S^M_{x \to v_2}, x : \neg \Box \neg C \mid U \mid K \rangle \quad \cdots \quad \langle S, v_n : C, v_n : \Box \neg C, S^M_{x \to v_n}, x : \neg \Box \neg C \mid U \mid K \rangle} (\Box^-)$$
$$\text{if } \nexists u \text{ s.t. } \{u : C, u : \Box \neg C, S^M_{x \to u}\} \subseteq S \qquad \forall v_i \in \mathcal{D}(\mathbf{B}), x \neq v_i$$

$$\frac{\langle S \mid U \mid K \rangle}{\langle S, x : \Box \neg C \mid U \mid K \rangle \qquad \langle S, x : \neg \Box \neg C \mid U \mid K \rangle} (cut) \quad \text{if } x : \neg \Box \neg C \notin S \text{ and } x : \Box \neg C \notin S, \ C \in \mathcal{L}_{\mathbf{T}}, \ x \in \mathcal{D}(\mathbf{B})$$

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

Figure 2: The calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$. To save space, we omit the rules $(\sqcup^+)$ and $(\sqcup-)$.

More in detail, the rules of $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ are shown in Figure **??**. 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 $(\sqsubseteq)$ 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 $x : \neg\Box\neg C \notin \mathbf{B}^{\Box^-}$, then $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ detects a clash, called $(\text{Clash})_{\Box^-}$: this corresponds to the situation in which $x : \neg\Box\neg C$ does not belong to $\mathbf{B}$, while $S, x : \neg\neg C$ 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 $(\text{Clash})_{\emptyset}$. 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 $\mathbf{B}$.

$\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH2}$ always terminates. Intuitively, termination is ensured by the fact that dynamic rules make use of labels belonging to $\mathcal{D}(\mathbf{B})$, which is finite, rather than introducing “new” labels in the tableau.

Definition 17. 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 $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{min}$ checks whether a query F can be minimally entailed from a *KB* by means of the following procedure:

- the calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{PH1}$ 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) 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 the procedure says YES

  else the procedure says NO

The following theorem shows that the overall procedure is sound and complete.

Theorem 3 (Soundness and completeness of $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$)**.** *$\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ is a sound and complete decision procedure for verifying if $KB \models_{min}^{\mathcal{L}_\mathbf{T}} F$.*

We provide an upper bound on the complexity of the procedure for computing the minimal entailment $\mathrm{KB} \models_{min}^{\mathcal{L}_\mathbf{T}} F$:

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

5. Background

6. Logics

7. Implementation

7.1. Architecture

7.2. Conclusion

References

[] Buchheit, M., Donini, F. M., Schaerf, A., 1993. Decidable reasoning in terminological knowledge representation systems. Journal of Artificial Intelligence Research (JAIR) 1, 109–138.

[] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G. L., 2009. $\mathcal{ALC}+\mathbf{T}$: a preferential extension of Description Logics. Fundamenta Informaticae 96, 1–32.

[] Golbreich, C., Zhang, S., Bodenreider, O., Sep. 2006. The foundational model of anatomy in owl: Experience and perspectives. Web Semant. 4 (3), 181–195.
URL `http://dx.doi.org/10.1016/j.websem.2006.05.007`

[] Nute, D., 1980. Topics in conditional logic. Reidel, Dordrecht.

[] Sidhu, E. S., Dillon, T. S., Chang, E., Sidhu, B. S., 2005. Protein ontology development using owl. In: in Proceedings of the First OWL Experiences and Directions Workshop. p. 188.

[] Wolstencroft, K., Brass, A., Horrocks, I., Lord, P., Sattler, U., Turi, D., Stevens, R., 2005. A little semantic web goes a long way in biology. In: Proceedings of the 4th International Conference on The Semantic Web. ISWC'05. Springer-Verlag, Berlin, Heidelberg, pp. 786–800.
URL `http://dx.doi.org/10.1007/11574620_56`