A tableaux calculus for $\mathcal{ALC} + T_{\min R}$

Laura Giordano *  Valentina Gliozzi †  Adam Jalal ‡  
Nicola Olivetti §  Gian Luca Pozzato ¶

June 12, 2013

Abstract
In this report we introduce a tableau calculus for deciding query entailment in  
preferential Description Logic $\mathcal{ALC} + T_{\min R}$ that allows to give a complexity  
upper bound for the logic, namely that query entailment is in $\text{CO-NEXP}^{\text{NP}}$.

1 Introduction

Non-monotonic extensions of Description Logics (DLs) have been actively investigated  
in the last few years [BH95a, BH95b, BLW09, DLN+98, DNR02, ELST04, Str93,  
CS10]. In this work, we focus on preferential Description Logics, obtained by (i) adding  
a typicality operator $T$ to standard DLs and (ii) based on a minimal model  
semantics in order to perform nonmonotonic inferences. In particular, here we focus on  
the logic $\mathcal{ALC} + T_{\min R}$: in this logic, the semantics underlying the typicality operator  
$T$ is strongly related to Kraus, Lehmann and Magidor’s rational logic $R$ [LM92].

We provide a decision procedure for checking minimal entailment in $\mathcal{ALC} + T_{\min R}$.  
Our decision procedure has the form of tableau calculus, with a two-step tableau con-  
struction. 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 indeed represents a minimal model. Termination is en-  
sured by means of a standard blocking mechanism. Our procedure can be used to  
determine constructively an upper bound of the complexity of $\mathcal{ALC} + T_{\min R}$. Namely  
we obtain that checking query entailment for $\mathcal{ALC} + T_{\min R}$ is in $\text{CO-NEXP}^{\text{NP}}$.

2 The logic $\mathcal{ALC} + T_{\min R}$

Given an alphabet of concept names $C$, of role names $R$, and of individual constants  
$O$, the language $\mathcal{L}$ of the logic $\mathcal{ALC} + T_{R}$ is defined by distinguishing concepts and

---

*DISIT - Università del Piemonte Orientale - Alessandria, Italy - laura.giordano@unipmn.it
† Dip. Informatica - Università di Torino - Italy - valentina.gliozzi@unito.it
‡ BE/CO/DA CERN, Geneva, Switzerland adam.jalal@cern.ch
§ Aix-Marseille Univ. - CNRS, LSIS UMR 7296 - France - nicola.olivetti@univ-amu.fr
¶ Dip. Informatica - Università di Torino - Italy - gianluca.pozzato@unito.it
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 \( \mathbb{T}(C) \) are extended concepts of \( \mathcal{L} \);
  - boolean combinations of extended concepts are extended concepts of \( \mathcal{L} \).

A knowledge base is a pair \((\text{TBox}, \text{ABox})\). TBox contains subsumptions \( C \sqsubseteq D \), where \( C \in \mathcal{L} \) is an extended concept of the form either \( C' \) or \( \mathbb{T}(C') \), 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 \( \mathbb{T} \), we extend the definition of a model used in “standard” terminological logic \( \mathcal{ALC} \):

**Definition 1 (Semantics of \( \mathbb{T} \) with selection function).** A model is any structure

\[ \langle \Delta, I, f_\mathbb{T} \rangle \]

where:

- \( \Delta \) 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:
  - \( \top^I = \Delta \)
  - \( \bot^I = \emptyset \)
  - \( (\neg C)^I = \Delta \setminus C^I \)
  - \( (C \sqcap D)^I = C^I \cap D^I \)
  - \( (C \sqcup D)^I = C^I \cup D^I \)
  - \( (\forall R.C)^I = \{ x \in \Delta \mid \forall y. (x, y) \in R^I \rightarrow y \in C^I \} \)
  - \( (\exists R.C)^I = \{ x \in \Delta \mid \exists y. (x, y) \in R^I \text{ and } y \in C^I \} \)
  - \( (\mathbb{T}(C))^I = f_\mathbb{T}(C^I) \)

- Given \( S \subseteq \Delta \), \( f_\mathbb{T} \) is a function \( f_\mathbb{T} : \text{Pow}(\Delta) \rightarrow \text{Pow}(\Delta) \) satisfying the following properties:
  \[ (f_\mathbb{T} - 1) \quad f_\mathbb{T}(S) \subseteq S \]
  \[ (f_\mathbb{T} - 2) \quad \text{if } S \neq \emptyset, \text{ then also } f_\mathbb{T}(S) \neq \emptyset \]
  \[ (f_\mathbb{T} - 3) \quad \text{if } f_\mathbb{T}(S) \subseteq R, \text{ then } f_\mathbb{T}(S) = f_\mathbb{T}(S \cap R) \]
  \[ (f_\mathbb{T} - 4) \quad f_\mathbb{T}(\bigcup S_i) \subseteq \bigcup f_\mathbb{T}(S_i) \]
  \[ (f_\mathbb{T} - 5) \quad \bigcap f_\mathbb{T}(S_i) \subseteq f_\mathbb{T}(\bigcup S_i) \]
  \[ (f_\mathbb{T} - \text{R}) \quad \text{if } f_\mathbb{T}(S) \cap R \neq \emptyset, \text{ then } f_\mathbb{T}(S \cap R) \subseteq f_\mathbb{T}(S) \]
Intuitively, given the extension of some concept \( C \), the selection function \( f_T \) selects the typical instances of \( C \). \((f_T - 1)\) requests that typical elements of \( S \) belong to \( S \). \((f_T - 2)\) requests that if there are elements in \( S \), then there are also typical such elements. The following properties constrain the behavior of \( f_T \) with respect to \( \cap \) and \( \cup \) in such a way that they do not entail monotonicity. According to \((f_T - 3)\), 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). \((f_T - 4)\) corresponds to one direction of the equivalence \( f_T(\bigcup S_i) = \bigcup f_T(S_i) \), so that it does not entail monotonicity. Similar considerations apply to the equation \( f_T(\bigcap S_i) = \bigcap f_T(S_i) \), of which only the inclusion \( \bigcap f_T(S_i) \subseteq f_T(\bigcap S_i) \) holds. \((f_T - 5)\) is a further constraint on the behavior of \( f_T \) with respect to arbitrary unions and intersections; it would be derivable if \( f_T \) were monotonic. \((f_T - R)\) is the property corresponding to the rational monotonicity in [LM92]: this property is not assumed to hold in the logic \( ALC + T_{min} \) introduced in [GGOP13].

In [GGOP10b], 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, transitive and modular relation among individuals and that the typical members of a concept \( C \) (i.e., those selected by \( f_T(C^I) \)) are the minimal elements of \( C \) with respect to this relation. A relation is modular if, for all \( x, y, z \), if \( x < y \), then either \( x < z \) or \( z < y \). 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 \( \Delta \) 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 [Nut80] \(^1\): 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, transitive and modular relation \( < \) over a domain \( \Delta \), called preference relation, for all \( S \subseteq \Delta \), we define

\[
\text{Min}_{<}(S) = \{ x \in S \mid \exists y \in S \text{ s.t. } y < x \}
\]

We say that \( < \) satisfies the Smoothness Condition if for all \( S \subseteq \Delta \), for all \( x \in S \), either \( x \in \text{Min}_{<}(S) \) or \( \exists y \in \text{Min}_{<}(S) \) such that \( y < x \).

The following representation theorem is proved in [GGOP10b]:

**Theorem 3.** Given any model \( \langle \Delta, I, f_T \rangle \), \( f_T \) satisfies postulates \((f_T - 1)\) to \((f_T - 5)\) and \((f_T - R)\) above iff there exists an irreflexive, transitive and modular relation \( < \) on \( \Delta \), satisfying the Smoothness Condition, such that for all \( S \subseteq \Delta \), \( f_T(S) = \text{Min}_{<}(S) \).

\(^1\)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.
Having the above Representation Theorem, from now on, we will refer to the following semantics:

**Definition 4** (Semantics of $\mathcal{ALC} + \mathcal{T}_R$). A model $\mathcal{M}$ of $\mathcal{ALC} + \mathcal{T}_R$ is any structure $\langle \Delta, I, < \rangle$ where:

- $\Delta$ is the domain;
- $<$ is an irreflexive, transitive and modular relation over $\Delta$ satisfying the Smoothness Condition (Definition 2);
- $I$ 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 1 with the exception of $(T(C))^I$, which is defined as $(T(C))^I = \text{Min}_<(C^I)$.

Let us now introduce the notion of satisfiability of an $\mathcal{ALC} + \mathcal{T}_R$ 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.

**Definition 5** (Model satisfying a Knowledge Base). Consider a model $\mathcal{M}$, as defined in Definition 4. We extend $I$ so that it assigns to each individual constant $a \in \mathcal{O}$ an element $a^I \in \Delta$, and $I$ satisfies the unique name assumption. Given a KB $(\text{TBox, ABox})$, we say that:

- $\mathcal{M}$ satisfies TBox iff for all inclusions $C \subseteq D$ 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 $\mathcal{ALC} + \mathcal{T}_R$ if it holds in all models satisfying KB. In this case we write $\text{KB} \models_{\mathcal{ALC} + \mathcal{T}_R} F$.

Notice that the meaning of $T$ can be split into two parts: for any $x$ of the domain $\Delta$, $x \in (T(C))^I$ 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 $T$ (for the purpose of the calculus that we will present in Section 4), 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 6. Given a model \( M \) as in Definition 4, we extend the definition of \( I \) with the following clause:

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

It is easy to observe that \( x \) is a typical instance of \( C \) if and only if it is an instance of \( C \) and \( □¬C \), that is to say:

**Proposition 7.** Given a model \( M \) as in Definition 4, given a concept \( C \) and an element \( x \in \Delta \), we have that

\[
x \in (T(C))^I \iff x \in (C \sqcap □¬C)^I
\]

Since we only use \( □ \) to capture the meaning of \( T \), in the following we will always use the modality \( □ \) followed by a negated concept, as in \( □¬C \).

The Smoothness condition, together with the transitivity of \( < \), ensures the following Lemma:

**Lemma 8.** Given an \( ALC + T \) model as in Definition 4, 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 \( z \in Min_<(C^I) \).

**Proof.** Since \( y \in C^I \), by the Smoothness Condition we have that either (i) \( y \in Min_<(C^I) \) or (ii) there is \( z < y \) such that \( z \in Min_<(C^I) \). In case (i) we are done. In case (ii), since \( < \) is transitive, we have also that \( z < x \) and we are done. \( \square \)

3 The logic \( ALC + T_{\text{min}} \)

As mentioned in the Introduction, the logic \( ALC + T \) presented in [GGOP10b] allows to reason about typicality. However, it is monotonic. In order to perform some useful nonmonotonic inferences, we restrict our attention to the minimal \( ALC + T \) models. As a difference with respect to \( ALC + T \), 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 \( L_T \) 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 \( L_T \) contains at least all concepts \( C \) such that \( T(C) \) occurs in the KB. Notice that in case \( L_T \) contains more concepts than those occurring in the scope of \( T \) 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 \( (T(C))^I = (C \sqcap □¬C)^I \): \( x \) is a typical instance of a concept \( C \ (x \in (T(C))^I) \) when it is an instance of \( C \) and there is no other instance of \( C \) preferred to \( x \), i.e. \( x \in (C \sqcap □¬C)^I \). By contraposition an instance of \( C \) is atypical if \( x \in (□¬C)^I \) therefore in order to minimize the atypical instances of \( C \), we minimize the instances of \( □¬C \). Notice that this is different from maximizing the instances.
of $T(C)$. 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 $T(C)$ would imply maximizing also the extension of $C$).

We define the set $\mathcal{M}_{L_T}^{-}$ of negated boxed formulas holding in a model, relative to the concepts in $L_T$:

**Definition 9.** Given a model $\mathcal{M} = \langle \Delta, I, < \rangle$ and a set of concepts $L_T$, we define

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

Let KB be a knowledge base and let $L_T$ be a set of concepts occurring in KB.

**Definition 10** (Preferred and minimal models). Given a model $\mathcal{M} = \langle \Delta_M, I_M, <_M \rangle$ of KB and a model $\mathcal{N} = \langle \Delta_N, I_N, <_N \rangle$ of KB, we say that $\mathcal{M}$ is preferred to $\mathcal{N}$ with respect to $L_T$, and we write $\mathcal{M} <_{L_T} \mathcal{N}$, if the following conditions hold:

- $\Delta_M = \Delta_N$
- $a^\mathcal{M} = a^\mathcal{N}$ for all individual constants $a \in O$
- $\mathcal{M}_{L_T}^{-} \subset \mathcal{N}_{L_T}^{-}$.

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

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 $a \in O$. We assume that, for all $T(C')$ occurring in $F$, $C' \in L_T$. 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 $\mathcal{ALC} + T_{min}^R$.

**Definition 11** (Minimal Entailment in $\mathcal{ALC} + T_{min}^R$). A query $F$ is minimally entailed from a knowledge base KB with respect to $L_T$ if it holds in all models of KB that are minimal with respect to $L_T$. We write $KB \models_{min}^{L_T} F$.

## 4 A Tableaux Calculus for $\mathcal{ALC} + T_{min}^R$

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 $\mathcal{TAB}^{ALC+T_{min}^R}$, which extends the calculus $T^{ALC+T}$ presented in [GGOP13], and allows to reason about minimal models.

$\mathcal{TAB}^{ALC+T_{min}^R}$ 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 $\mathcal{TAB}^{ALC+T_{min}^R}$, simply verifies whether $KB \cup \{ \neg F \}$ is satisfiable in an $\mathcal{ALC} + T_R$ model, building candidate models. In the second
phase another tableau calculus, called $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH2}$, 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, $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH2}$ tries to build a “smaller” model of KB, i.e. a model whose individuals satisfy less formulas $\neg \Box \rightarrow C$ than the corresponding candidate model. The whole procedure $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{min}$ is formally defined at the end of this section (Definition 39).

$\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{min}$ 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.

$\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{min}$ 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 12** (Constraint). A constraint (or labelled formula) is a syntactic entity of the form either $x \overset{R}{\rightarrow} y$ 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 $x \overset{R}{\rightarrow} y$ 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 $C^+$. As we will define in Definition 15, 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 $a \overset{R}{\rightarrow} b$.

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

### 4.1 The tableau calculus $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH1}$

Let us first define the basic notions of a tableau system in $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH1}$.

**Definition 13** (Tableau of $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH1}$). A tableau of $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH1}$ 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 $C \subseteq D^L$, representing subsumption relations $C \subseteq D$ 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 $\langle S_{i-1} \mid U_{i-1} \rangle$ by applying a rule of $\mathcal{TAB}^{\text{ACC}+\mathcal{T}_R}_{PH1}$ (see Figure 1), 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 (Clash)$_\top$ or (Clash)$_\bot$), otherwise it is open. A tableau is closed if all its branches are closed.

In the following, we will often refer to the height of a tableau: intuitively, the height of a tableau corresponds to the height of the tree of Definition 13. This is formally stated as follows:

---

$^2$As we will discuss later, this list is used in order to ensure the termination of the tableau calculus.
**Definition 14** (Height of a tableau). Given a tableau of $\text{TAB}^{\text{ALC+TR}}_{\text{PH1}}$ having $\langle S \mid U \rangle$ as a root, we define its height $h$ as follows:

- $h = 0$ if no rule is applied to $\langle S \mid U \rangle$;
- $h = 1 + \max\{h_1, h_2, \ldots, h_n\}$ if a rule $(R)$ is applied to $\langle S \mid U \rangle$ and $h_1, h_2, \ldots, h_n$ are the heights of the tableaux whose roots are the conclusions of $(R)$.

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 15** (Corresponding constraint system). Given a knowledge base $\text{KB}=(\text{TBox},\text{ABox})$, we define its corresponding constraint system $\langle S \mid U \rangle$ as follows:

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

**Definition 16** (Model satisfying a constraint system). Let $\mathcal{M} = \langle I, < \rangle$ be a model as defined in Definition 4. We define a function $\alpha$ which assigns to each variable of $\forall \alpha$ an element of $\Delta$, and assigns every individual constant $a \in \mathcal{O}$ to $a^I \in \Delta$. $\mathcal{M}$ satisfies a constraint $F$ under $\alpha$, 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 $\alpha$ such that $\mathcal{M}$ satisfies every constraint in $S$ under $\alpha$ and that, for all $C \subseteq D^\theta \in U$ 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 17.** $\text{KB}=(\text{TBox},\text{ABox})$ is satisfiable in an $\text{ALC} + \text{TR}$ model if and only if its corresponding constraint system $\langle S \mid U \rangle$ is satisfiable in the same model.

**Proof.** We show that a model $\mathcal{M}$ as in Definition 4 satisfies KB if and only if there is a function $\alpha$ such that $\langle S \mid U \rangle$ is satisfiable in $\mathcal{M}$ under $\alpha$. We simply define $\alpha$ as follows: $\alpha$ assigns each individual constant $a \in \mathcal{O}$ to $a^I \in \Delta$. Let us first consider the ABox and each formula belonging to it. By Definition 5, given $C(a) \in \text{ABox}$, we have that $\mathcal{M} \models C(a)$ iff $a^I \in C^I$. By Definition 15 of the corresponding constraint system, we have that $a : C \in S$; since $a$ is an individual constant occurring in the ABox, we have that $\alpha(a) = a^I$, thus $a^I \in C^I$ iff $\alpha(a) \in C^I$ and, by Definition 16, iff $\mathcal{M} \models_\alpha a : C$. In case $\mathcal{M} \models aRb$, we have that $a \xrightarrow{R} b \in S$. $\mathcal{M} \models_\alpha aRb$ iff $(a^I, b^I) \in R^I$ iff $(\alpha(a), \alpha(b)) \in R^I$ iff $\mathcal{M} \models_\alpha a \xrightarrow{R} b$. Concerning the TBox, $\mathcal{M} \models C \subseteq D$ iff, for each $x \in \Delta$, if $x \in C^I$ then $x \in D^I$, i.e., $\mathcal{M} \models C \subseteq D^\theta$. \hfill $\square$
To verify the satisfiability of $\text{KB} \cup \{\neg F\}$, we use $\text{TAB}_{PH1}^{\text{ALC+TR}}$ 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 $\text{TAB}_{PH1}^{\text{ALC+TR}}$ 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 [GGOP09a], given a node $\langle S \mid U \rangle$, for each subsumption $C \sqsubseteq D \in U$ 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 $\text{TAB}_{PH1}^{\text{ALC+TR}}$ we need some more definitions. First, as in [BDS93], 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 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 $\text{TAB}_{PH1}^{\text{ALC+TR}}$.

**Definition 18.** 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 $S^M_{x \rightarrow y}$, 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 19.** Given a tableau node $\langle S \mid U \rangle$ and two labels $x$ and $y$, we define

$$S^M_{x \rightarrow y} = \{y : \neg C, y : \Box \neg C \mid x : \Box \neg C \in S\}.$$  

The rules of $\text{TAB}_{PH1}^{\text{ALC+TR}}$ 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 rule $(\prec)$ captures the modularity of the preference relation. In particular, if the current constraint system contains $x \prec y$ and another label $z$ occurs in it, then the calculus introduces a branch on the two plausible situations of the preference relation: by its modularity, either $x \prec z$ or $z \prec y$ have to be added to the current constraint system. In order to ensure termination, the rule $(\prec)$ is applicable only in case both $x \prec z$ and $z \prec y$ do not belong to the current constraint system.
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 \((\exists -)\), \((\forall +)\), \((\Box +)\) and \((cut)\) this is needed in order to have a complete calculus. Rules for \(<, \cap, \cup, \neg\), and \(T\) also copy their principal formulas in their conclusions for uniformity sake.

In order to ensure the completeness of the calculus, the rules of \(\mathcal{T}AB^{\mathcal{ALC}+\mathcal{Tn}}_{PH1}\) 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 \(y < x\);

2. apply \((<)\) and dynamic rules only if no static rule is applicable.

The calculus so obtained is sound and complete with respect to the semantics in Definition 16. In order to prove this, we first define the notion of regular node:

**Definition 20 (Regular node).** A node \(\langle S \mid U \rangle\) of \(\mathcal{T}AB^{\mathcal{ALC}+\mathcal{Tn}}_{PH1}\) is regular if and only if the following conditions hold:

- if \(x: \Box - C \in S\), then \(C \in \mathcal{L}_T\);
• if \( x : \neg \Box \neg C \in S \), then \( C \in L_T \).

We can show that:

**Lemma 21.** Given an \( \mathcal{ALC} + T_R \) KB, its corresponding constraint system \( \langle S | U \rangle \), and a set of concepts \( L_T \), the nodes of every tableau of \( \mathcal{TAB}^{\mathcal{ALC}+T_R}_1 \) having \( \langle S | U \rangle \) as a root are regular nodes.

**Proof.** Considering each rule of \( \mathcal{TAB}^{\mathcal{ALC}+T_R}_1 \), we can show that if the premise is a regular node, then the conclusions are also regular nodes. The rules introducing boxed formulas are \( \langle \rangle \), \( (T^+) \), \( (T^-) \), \( (cut) \), and \( (\Box^-) \). \( (T^+) \) and \( (T^-) \) introduce \( \neg \Box \neg C \) in their conclusions when applied to some formula \( \neg T(C) \): we conclude that the conclusions are regular nodes, since \( C \in L_T \) by definition of \( L_T \) (it contains at least all concepts in the scope of the T operator). By definition of the rule, \( (cut) \) introduces \( \neg \Box \neg C \) in its conclusions by taking \( C \in L_T \), and we are done. Concerning \( (\Box^-) \), suppose an application to a regular node \( \langle S,x : \neg \Box \neg C \rangle \). Each conclusion has the form \( \langle S,x : \neg \Box \neg D \rangle \), and we conclude as follows: \( C \in L_T \), otherwise the premise would not be regular; if \( y : \Box \neg D \in S_x \rightarrow y \), then \( x : \Box \neg D \in S \) and \( D \in L_T \), otherwise the premise would not be regular. Last, the rule \( \langle \rangle \) introduces \( y : \Box \neg D \in S_x \rightarrow y \): exactly as in the case of \( (\Box^-) \), the node is regular otherwise the premise would not, against the inductive hypothesis.

From now on, by Lemma 21, we restrict our concern to regular nodes.

Furthermore, we introduce the notions of witness and of blocked label:

**Definition 22** (Witness and Blocked label). *Given a constraint system \( \langle S | U \rangle \) 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 \( \langle S | U \rangle \) if \( y \) has witness \( x \).*

By the strategy on the application of the rules described above and by Definition 22, we can prove the following Lemma:

**Lemma 23.** *In any constraint system \( \langle S | U \rangle \), if \( x \) is blocked, then it has exactly one witness.*

**Proof.** The property immediately follows from the definition of a witness (Definition 22). \( \square \)

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 \( \langle S,x : \exists R.C | U \rangle \) 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 $\exists z \prec x$ such that $z \equiv S.x \equiv R.C x$. The same blocking machinery is used to control the application of ($\Box^-$), which can be applied only if $\exists z \prec x$ such that $z \equiv S.x \equiv R.C x$.

We also need the following definitions:

**Definition 24 (Satisfiability of a branch).** A branch $B$ of a tableau of $\mathcal{T}_{AB}^{\text{ALC} + \text{TR}}$ is satisfiable w.r.t. $\mathcal{ALC} + \mathcal{TR}$ by a model $M$ if there is a mapping $\alpha$ from the labels in $B$ to the domain of $M$ such that for all constraint systems $\langle S \mid U \rangle$ on $B$, $M$ satisfies under $\alpha$ (see Definition 16) every constraint in $S$ and, for all $C \subseteq D^L \in U$ and for all $x$ occurring in $S$, we have that if $\alpha(x) \in C^I$ then $\alpha(x) \in D^I$.

**Definition 25 (Saturated Branch).** A branch $B=\langle S_0 \mid U_0 \rangle, \langle S_1 \mid U_1 \rangle, \ldots, \langle S_i \mid U_i \rangle, \ldots$ is saturated if the following conditions hold:

1. for all $C \subseteq D^L$ and for all labels $x$ occurring in $B$, either $x : \neg C$ or $x : D$ belong to $B$;
2. if $x : \top(C)$ occurs in $B$, then $x : C$ and $x : \Box \neg C$ occur in $B$;
3. if $x : \neg \top(C)$ occurs in $B$, then either $x : \neg C$ or $x : \neg \Box \neg C$ occur in $B$;
4. if $x : \Box \neg C$ and $y < x$ occur in $B$, also $y : \neg C$ and $y : \Box \neg C$ occur in $B$;
5. if $x : \neg \Box \neg C$ occurs in $B$, then either there is $y$ such that $y < x, y : C, y : \Box \neg C$, and $S_{y \rightarrow y}^M$ occur in $B$ or $x$ is blocked by a witness $w$, and $y < w, y : C, y : \Box \neg C$, and $S_{y \rightarrow w}^M$ occur in $B$;
6. if $x : \exists R.C$ occurs in $B$, then either there is $y$ such that $x \overset{R}{\rightarrow} y$ and $y : C$ occur in $B$ or $x$ is blocked by a witness $w$, and $w \overset{R}{\rightarrow} y$ and $y : C$ occur in $B$;
7. if $x : \forall R.C$ and $x \overset{R}{\rightarrow} y$ occur in $B$, also $y : C$ occurs in $B$;
8. for $x : \neg \forall R.C$ and for $x : \neg \exists R.C$ the condition of saturation is defined symmetrically to points 6 and 7, respectively;
9. for the boolean rules the condition of saturation is defined in the usual way. For instance, if $x : C \cap D$ occurs in $B$, so $x : C$ and $x : D$ occur in $B$;
10. for all $C \in \mathcal{L}_x$ and for all labels $x$ occurring in $B$, either $x : \neg \Box \neg C$ occurs in $B$ or $x : \Box \neg C$ occurs in $B$;
11. for all labels $x, y, z$ occurring in $B$, if $x < y$ occurs in $B$, then either $x < z$ occurs in $B$ or $z < y$ occurs in $B$. 

12
By following the strategy on the order of application of the rules outlined above and by Lemma 23, we can prove that any open branch can be expanded into an open saturated branch. However, it is worth noticing that, as a difference with the tableau calculus for $ALC + T_R$ presented in [GGOP09a], as well as the one for the standard DL $ALC$ introduced in [BDS93], the strategy on the order of application of the rules of $TAB^{ALC+T_R}$ does not ensure that the labels are considered one at a time, following the order $\prec$. Indeed, the rules $(\exists^+)$ and $(\forall^-)$ reconsider labels already introduced in the branch in their conclusions. When $(\forall^-)$ is applied to a formula $x : \neg \square \neg C$, a branching is introduced on the choice of the label used in the conclusion. In the leftmost conclusion, a “new” label $y$ is used to add $y : C, y : \square \neg C, S^M_{x \rightarrow y}$. In all the other conclusions, a label $v_i$ already present in the branch is chosen. Therefore, rules of $TAB^{ALC+T_R}$ are furthermore applied to formulas labelled with the “older” label $v_i$. One may conjecture that this could lead to an incomplete calculus, in particular that condition 4 of saturation above could be not fulfilled. We show in the proof of Proposition 26 below that this does not happen. Intuitively, it could be the case that above could be not fulfilled. We show in the proof of Proposition 26 below that this could lead to an incomplete calculus, in particular that condition 4 of saturation above could be not fulfilled. We show in the proof of Proposition 26 below that this does not happen. Intuitively, it could be the case that $v_i : \square \neg C$ is introduced by $(\forall^-)$, however a previous application of the same rule to $v_i : \neg \square \neg D$, introducing a new label $u < v_i$, causes the loss of the propagation of the concepts $\neg C$ and $\square \neg C$ ($u : \neg C$ and $u : \square \neg C$ should belong to a saturated branch). However, this cannot happen due to the order on the application of the rules and, in particular, by virtue of the rule $(cut)$: since $(cut)$ is a static rule, it has been already applied by using label $v_i$ before taking into account labels “younger” than $v_i$. By Lemma 21, $C \in L_T$, therefore either $v_i : \neg \square \neg C$ or $v_i : \square \neg C$ have been already introduced: in the former case, the branch is closed, otherwise $u : \neg C$ and $u : \square \neg C$ have been also introduced by the application of $(\forall^-)$ introducing $u$ in the computation of $S^M_{v_i \rightarrow u}$, ensuring the saturation of the branch.

**Proposition 26.** Any open branch $B$ can be expanded by applying the rules of $TAB^{ALC+T_R}$ into an open saturated branch.

**Proof.** As mentioned, let us analyze the case of condition 4. For the other conditions, the proof is standard and then left to the reader. Suppose that $x : \square \neg C$ and $y < x$ belong to $B$. The relation $y < x$ has been added to $B$ either by an application of the rule $(\forall^-)$ to a node $(S, x : \neg \square \neg D \mid U)$ or by the rule $(<)$. We show that $x : \square \neg C$ was already in $S$ before the application of $(<)$ or $(\forall^-)$. Let us first consider the application of $(\forall^-)$ to $x : \neg \square \neg D$. By the order on the application of the rules, if $(\forall^-)$ is going to be applied to introduce $y < x$, then all the static rules have already been applied to formulas labelled by $x$, including $(<)$ and the $(cut)$ rule. By Lemma 21, we have that $C \in L_T$, and $(cut)$ has been also applied to $C$ by using the label $x$. Therefore, $S$ contains either $x : \neg \square \neg C$ or $x : \square \neg C$: the former case cannot be, otherwise $B$ would become closed when $x : \square \neg C$ is introduced, then we are done. By the fact that $x : \square \neg C \in S$, we can conclude that $y : C, y : \square \neg C$ belong to $B$, since they are introduced by the application of $(\forall^-)$ to $(S, x : \neg \square \neg D \mid U)$. Let us now analyze the case in which $y < x$ has been introduced by an application of $(<)$ to either $(S, y < z \mid U)$ or $(S, z < x \mid U)$: as for $(\forall^-)$, when $(<)$ is applied, the labelled formula $x : \neg \square \neg C$ belongs to $B$, because $(cut)$ has been already applied by the order on the application of the rules: therefore, in both cases, we have that $\{y : \neg C, y : \square \neg C\} \subseteq S^M_{x \rightarrow y}$, and they are introduced in $B$ by the rule $(<)$ having
S^M_{x \rightarrow y}$ in its conclusions. \hfill \Box

In order to show the completeness of $\mathcal{TAX}_{PH1}^{ALC+TR}$, given an open, saturated branch $B$, we explicitly add to $B$ the relation $y < x$, if $x$ is blocked and $w$ is the witness of $x$ and $y < w$ occurs in $B$.

Before proving the completeness, we prove the following lemmas:

**Lemma 27.** In any tableau built by $\mathcal{TAX}_{PH1}^{ALC+TR}$, there is no open saturated branch $B$ containing an infinite descending chain of labels $\ldots x_2 < x_1 < x_0$.

**Proof.** The only way to obtain an infinite descending chain $\ldots x_2 < x_1 < x_0$ would be to have either (i) a loop or (ii) an infinite set of distinct labels. We can show that neither (i) nor (ii) can occur.

As far as (i) is concerned, suppose for a contradiction that there is a loop, that is to say there is an infinite descending chain $x < u < \ldots < y < x$. We distinguish three cases:

- the relation $x < u$ has been inserted in the branch by the rule $(\square -)$ in the leftmost conclusion of this rule: this cannot be the case, since in the leftmost conclusion of the rule $x$ is a new label;

- the relation $x < u$ has been inserted in the branch by the rule $(\square -)$ not in the leftmost conclusion, i.e. by using $x$ occurring in $B$, $x \neq u$: the relation $y < x$ has been introduced by an application of $(\square -)$, then there is $x : \neg \square \neg C$ in $B$ (the formula to which the $(\square -)$ rule is applied). Therefore, $x : \neg \square \neg C$ belongs to $B$, as well as $y : \square \neg C$ belongs to $B$. Moreover, $y_i : \square \neg C$ belongs to $B$, for all $y_i$, then also $u : \square \neg C$ belongs to $B$. When $(\square -)$ is applied to introduce $x < u$, the constraint $x : \square \neg C$ is also added to $B$, since $x : \square \neg C \in S^M_{u \rightarrow x}$, which contradicts the hypothesis that $B$ was open;

- the relation $x < u$ has been inserted in the branch because $u$ is blocked by some witness $w$, and $x < w$ occurs in $B$. Notice, however, that in this case: 1. $x < w$ has been introduced by $(\square -)$ applied to some $w : \neg \square \neg C$, hence, $x : \square \neg C$ occurs in $B$; 2. similarly to the previous case, it can be shown that also for all $y_i$ and for $u$, we have that $y_i : \square \neg C$ and $u : \square \neg C$ belong to $B$; 3. since $w$ is a witness of $u$, also $u : \neg \square \neg C$ occurs in the branch $B$, which contradicts the hypothesis that $B$ was open.

Concerning (ii), suppose there were an infinite descending chain $\ldots < x_i \ldots < x_0$. Each relation must be generated by a $\neg \square \neg C$ that has not yet been used in the chain,
either by an application of the rule \((\Box^-)\) to \(\neg \Box \neg C\) in \(x_{i-1}\), or by an application of the rule \((\Box^-)\) to \(\neg \Box \neg C\) in the witness \(w\) of \(x_{i-1}\). Indeed, if \(\neg \Box \neg C\) had been previously used in the chain, say in introducing \(x_i < x_{i-1}\), for each \(x_j\) such that \(x_j < \ldots < x_i\), we have that \(x_j : \Box \neg C\) is in \(B\), hence \(x_j : \neg \Box \neg C\) cannot be in \(B\), otherwise \(B\) would be closed, against the hypothesis. Notice however that, by Lemma 21, the only formulas \(\neg \Box \neg C\) that appear in the branch are such that \(C \in \mathcal{L}_T\). Since \(\mathcal{L}_T\) is finite, it follows that also the number of possible different \(\neg \Box \neg C\) is finite, and the infinite descending chain cannot be generated. \(\Box\)

Let us now show that all the rules of \(\mathcal{T AB}_{PH1}^{ALC+Tn}\) are invertible. In order to do this, we first show that weakening is admissible, namely:

**Lemma 28 (Admissibility of weakening).** Given a constraint \(F\) and a constraint system \(\langle S \mid U \rangle\), if \(\langle S \mid U \rangle\) has a closed tableau in \(\mathcal{T AB}_{PH1}^{ALC+Tn}\), then also \(\langle S, F \mid U \rangle\) has a closed tableau in \(\mathcal{T AB}_{PH1}^{ALC+Tn}\).

**Proof.** By induction on the height of the closed tableau for \(\langle S \mid U \rangle\), in the sense of Definition 14. For the base case, it is easy to observe that, if \(\langle S \mid U \rangle\) is a clash, then also \(\langle S, F \mid U \rangle\) is a clash. As an example, consider the case of \(\langle S', x : \bot \mid U \rangle\), which is an instance of (Clash)\_\_\_: obviously, also \(\langle S', x : \bot, F \mid U \rangle\) is an instance of (Clash)\_\_. For the inductive step, we analyze the first step in the tableau construction for \(\langle S \mid U \rangle\), by considering all the rules. We only show the most interesting cases of \((\Box^-)\) and \((\Box)\), the other cases are similar and left to the reader. Suppose that \((\Box^-)\) has been applied to \(\langle S', x : \neg \Box \neg C \mid U \rangle\), by generating the conclusion \(\langle S', x : \neg \Box \neg C, y < x, y : C, y : \neg C, S_x^{\rightarrow y} \mid U \rangle\), where \(y\) does not occur in \(S'\), as well as the conclusions \(\langle S', x : \neg \Box \neg C, v_i < x, v_i : \neg \Box \neg C, S_{x-v_i} \mid U \rangle\) for each \(v_i\) occurring in \(S'\). We can apply the inductive inductive hypothesis on each conclusion, to obtain a closed tableau for \(\langle S', x : \neg \Box \neg C, y < x, y : C, y : \neg C, S_x^{\rightarrow y}, F \mid U \rangle\) and for \(\langle S', x : \neg \Box \neg C, v_i < x, v_i : \neg C, S_{x-v_i}, F \mid U \rangle\), from which we can conclude by an application of \((\Box^-)\) to obtain a closed tableau also for \(\langle S', x : \neg \Box \neg C, F \mid U \rangle\). Notice that, in case \(F\) contains the label \(y\), we can replace \(y\) in the tableau with a new label \(y'\) wherever it occurs. For \((\Box)\), consider a tableau starting with an application of such a rule to \(\langle S \mid U', C \subseteq D^L\rangle\), whose conclusion is \(\langle S, x : \neg C \cup D \mid U', C \subseteq D^L, x \rangle\) (with \(x \notin L\)). By inductive hypothesis, we have a closed tableau for \(\langle S, x : \neg C \cup D, F \mid U', C \subseteq D^L, x \rangle\), from which we obtain a closed tableau for \(\langle S, F \mid U', C \subseteq D^L \rangle\) by an application of \((\Box)\).

Now we can easily prove that the rules of \(\mathcal{T AB}_{PH1}^{ALC+Tn}\) are invertible:

**Lemma 29.** Let \((R)\) be a rule of the calculus \(\mathcal{T AB}_{PH1}^{ALC+Tn}\), let \(\langle S \mid U \rangle\) be its premise and let \(\langle S_1 \mid U_1 \rangle, \langle S_2 \mid U_2 \rangle, \ldots, \langle S_n \mid U_n \rangle\) be its conclusions. If \(\langle S \mid U \rangle\) has a closed tableau in \(\mathcal{T AB}_{PH1}^{ALC+Tn}\), then also \(\langle S_1 \mid U_1 \rangle, \langle S_2 \mid U_2 \rangle, \ldots, \langle S_n \mid U_n \rangle\) have a closed tableau, i.e. the rules of \(\mathcal{T AB}_{PH1}^{ALC+Tn}\) are invertible.

**Proof.** It can be easily observed that all the rules of \(\mathcal{T AB}_{PH1}^{ALC+Tn}\) copy their principal formulas in all their conclusions. Therefore, if we have a closed tableau for the premise of a given rule \((R)\), by weakening (Lemma 28 above) we have also a closed tableau for each of its conclusions, and we are done. \(\Box\)
By Lemma 29, we have that in $\mathcal{T}A\mathcal{B}^{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}}_{PH1}$ the order of application of the rules is not relevant. Hence, no backtracking is required in the tableau construction, and we can assume, without loss of generality, that a given constraint system $\langle S \mid U \rangle$ has a unique tableau.

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

**Theorem 30** (Soundness of $\mathcal{T}A\mathcal{B}^{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}}_{PH1}$). If the tableau for the constraint system corresponding to $KB \cup \{\neg F\}$ is closed then $KB \models_{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}} F$.

**Proof.** We first show that if the tableau for the constraint system corresponding to $KB \cup \{\neg F\}$ is closed, then (\text{*}) $KB \cup \{\neg F\}$ is unsatisfiable. By Proposition 17, $KB \cup \{\neg F\}$ is satisfiable if and only if its corresponding constraint system $\langle S \mid U \rangle$ is satisfiable in the same model. We proceed by induction on the height of the closed tableau for $\langle S \mid U \rangle$. For the base case, it is easy to observe that if $(S \mid U)$ is an instance of either (Clash) or (Clash)$_\bot$ or (Clash)$_\top$, then $KB \cup \{\neg F\}$ is unsatisfiable. For the inductive step, we consider each rule applied to the root $(S \mid U)$ of the closed tableau, and we show that $KB \cup \{\neg F\}$ is unsatisfiable assuming, by inductive hypothesis, that also the conclusions are unsatisfiable. We proceed by contraposition, that is to say, by considering each rule of $\mathcal{T}A\mathcal{B}^{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}}_{PH1}$, it can be shown that if the premise is satisfiable in an $\mathcal{A}\mathcal{L}\mathcal{C} + \mathcal{T}\mathcal{R}$ model, so is (at least) one of its conclusions. We only show the case of $(\leftarrow)$, the other cases are as for $\mathcal{T}A\mathcal{B}^{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}}_{PH1}$ in [GGOP13]. Suppose the premise $\langle S, x < y \mid U \rangle$ is satisfiable, i.e. there is a model $\mathcal{M} = \langle \Delta, I, \leftarrow \rangle$ and a function $\alpha$ such that $\mathcal{M} \models_\alpha F$ for each $F \in S$. Moreover, we have that $C^I \subseteq D^I$ for each $C \subseteq D^L \in U$. Finally, $\mathcal{M} \models_\alpha x < y$, i.e. $\alpha(x) < \alpha(y)$ in the model. Since $<$ is modular, given another label $z$, we have that either (i) $\alpha(z) < \alpha(y)$ or (ii) $\alpha(x) < \alpha(z)$. In case (i), since $\alpha(z) < \alpha(y)$, for all $y : \Box \neg D \in S$, we have that $\alpha(z) \in (\neg D)^I$ by Definition 6 and, since $<$ is transitive, $\alpha(z) \in (\neg D)^I$. Formulas $z : \neg D, z : \Box \neg D$ are those belonging to $S^M_{\neg z}$. We conclude that the leftmost conclusion of $(\leftarrow)$ is satisfiable in $\mathcal{M}$ via $\alpha$, since $\mathcal{M} \models_\alpha F$ for all $F \in S$, $\mathcal{M} \models_\alpha x < y, z < y$ and, furthermore, we have that $\mathcal{M} \models_\alpha S^M_{\neg z}$.

We can conclude by observing that, if $KB \not\models_{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}} F$, then $KB \cup \{\neg F\}$ is satisfiable in an $\mathcal{A}\mathcal{L}\mathcal{C} + \mathcal{T}\mathcal{R}$ model. Given (\text{*}), we conclude that $KB \not\models_{\mathcal{A}\mathcal{L}\mathcal{C}+\mathcal{T}\mathcal{R}} F$ by contraposition. $\square$

In the proof of the theorem and later in the paper we will use the notion of canonical model $\mathcal{M}^B$ built from an open branch $B$. The canonical model $\mathcal{M}^B = \langle \Delta_B, \leftarrow', I^B \rangle$ is defined as follows:

- $\Delta_B = \{x : x$ is a label appearing in $B\}$;
- we first define $\leftarrow'$ as follows: $\leftarrow' = \{y \leftarrow^* x : \text{either } y < x \text{ occurs in } B \text{ or } x \text{ is blocked and } w \text{ is the witness of } x \text{ (by Lemma 23 such } w \text{ exists) and } y < w \text{ occurs in } B\}$. We define $\leftarrow'$ as the transitive closure of relation $\leftarrow^*$;
- $I^B$ is an interpretation function such that for all atomic concepts $A$, $A^{I^B} = \{x : A$ occurs in $B\}$. $I^B$ is then extended to all concepts $C$ in the standard way, according to the semantics of the operators. For role names $R$,
Theorem 31 (Completeness of $\mathcal{T}AB^{|\cal{ALC}|+\cal{T}_R}$). If $KB \models_{\cal{ALC}+\cal{T}_R} F$, then the tableau for the constraint system corresponding to $KB \cup \{\neg F\}$ is closed.

Proof. We show the contrapositive, that if the tableau is open, then the starting constraint system $\langle \text{variables} \rangle$ is satisfied in an $\cal{ALC} + \cal{T}_R$ model, and by Proposition 17 $KB \cup \{\neg F\}$ is satisfiable in the same model, hence $KB \models_{\cal{ALC}+\cal{T}_R} F$. An open tableau contains an open branch that by Proposition 26 can be expanded into an open saturated branch. From such a branch, call it $B$, we define the canonical model $M^B = \langle \Delta_B, <', I^B \rangle$ as described above.

We can show that:

- $<'$ is irreflexive, transitive, modular, and satisfies the Smoothness Condition. Irreflexivity follows from the fact that the relation $<$ is either introduced by rule $\langle \neg \Box \rangle$ between a label $x$ already present in $B$ and either a new label or a label different from $x$, or it is explicitly added in case some $x : \neg \Box \neg C$ is on the branch and $x$ is blocked. In this case, suppose for a contradiction that $x < x$ is added, this means that $x$ is blocked by a witness $w$ and $x < w$, thus $w : \neg \Box \neg C$ belongs to $B$, as well as $x < w, x : C, x : \neg \Box C$ belong to $B$, but this contradicts the fact that $B$ is open (both $x : \neg \Box \neg C$ and $x : \Box \neg C$ occur). Modularity follows from the fact that $B$ is saturated (condition 11). Transitivity follows from definition of $<$. The Smoothness Condition follows from transitivity of $<'$ together with the finiteness of chains of $<$ deriving from Lemma 27.

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

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

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

- If $C$ is $\Box \neg D$ and $x : \Box \neg D$ occurs in $B$, then, by saturation, for all $y < x$, we have that also $y : \neg D$ occurs in $B$. By definition of $<'$, we have that $y < x$. By inductive hypothesis, $y \not\in D^{I^B}$ for all $y <' x$, and we are done. If $x : \neg \Box \neg D$ occurs in $B$, then by saturation and by definition of $<'$ there is $y$ s.t. $y <' x$, and $y : D$ and $y : \neg \Box \neg D$ occur in $B$. By inductive hypothesis, $y \in D^{I^B}$. It follows that $x \in (\neg \Box \neg D)^{I^B}$.
Proof. If \( C \) is \( \mathbf{T}(D) \) and \( x : \mathbf{T}(D) \) occurs in \( B \), by saturation, both \( x : D \) and \( x : \neg D \) occur in \( B \), hence by inductive hypothesis \( x \in D^{IB} \) and \( x \in (\neg D)^{IB} \), and by Proposition 7, \( x \in (\mathbf{T}(D))^{IB} \). If \( x : \neg \mathbf{T}(D) \) occurs in \( B \), then by saturation also either \( x : \neg D \) occurs in \( B \) or \( x : \neg \neg D \) occurs in \( B \). By inductive hypothesis either \( x \notin D^{IB} \) or \( x \notin (\neg D)^{IB} \). In both cases, we conclude that \( x \in (\neg \mathbf{T}(D))^{IB} \).

- for all \( C \subseteq D \in U \) and all labels \( x \), we want to show that either \( x \in (\neg C)^{IB} \) or \( x \in D^{IB} \), i.e., \( C^{IB} \subseteq D^{IB} \). By saturation, either \( x : \neg C \) occurs in \( B \) or \( x : D \) occurs in \( B \). The property follows by inductive hypothesis.

The above points allow us to conclude that \( M^B \) satisfies the starting constraint system.

The above theorem concerns satisfiability of \( \text{KB} \cup \{ \neg F \} \) in any \( \mathcal{ALC} + \mathbf{T}_R \) model (as in Definition 5). However in order to deal with entailment in \( \mathcal{ALC} + \mathbf{T}_{min R} \), we need something stronger: we need to restrict our attention to minimal models of \( \text{KB} \).

Given \( \mathcal{L}_T \), the following theorem shows that \( \text{KB} \cup \{ \neg F \} \) is satisfiable in a minimal model of \( \text{KB} \) (i.e. \( \text{KB} \not\models^{\mathcal{L}_T} F \)) if and only if the open tableau for the constraint system corresponding to \( \text{KB} \cup \{ \neg F \} \) contains an open branch \( B \) such that \( M^B \) is a minimal model of \( \text{KB} \). \( M^B \) is the canonical model built from \( B \) as in the construction used in the proof of Theorem 31 above. With the following theorem, the problem of deciding whether \( \text{KB} \not\models^{\mathcal{L}_T} F \) amounts to deciding whether any possibly open branch \( B \) of \( \mathcal{T}_{AB^{\mathcal{ALC}+\mathbf{T}_R}} \) gives rise to a \( M^B \) which is a minimal model of \( \text{KB} \). As we will see this is the purpose of the second phase of the calculus \( \mathcal{T}_{AB^{\mathcal{ALC}+\mathbf{T}_R}} \) introduced in the next subsection.

**Theorem 32.** Given \( \mathcal{L}_T \), \( \text{KB} \models^{\mathcal{L}_T} F \) if and only if there is no open branch \( B \) in the tableau built by \( \mathcal{T}_{AB^{\mathcal{ALC}+\mathbf{T}_R}} \) for the constraint system corresponding to \( \text{KB} \cup \{ \neg F \} \) such that \( M^B \) is a minimal model of \( \text{KB} \).

**Proof.** If direction. We show the contrapositive by proving that if \( \text{KB} \not\models^{\mathcal{L}_T} F \), i.e. if \( \text{KB} \cup \{ \neg F \} \) is satisfiable in a minimal model of \( \text{KB} \) w.r.t. \( \mathcal{L}_T \), then there is an open branch \( B \) such that \( M^B \) is a minimal model of \( \text{KB} \). The proof comprises three steps:

(i) if \( \text{KB} \not\models^{\mathcal{L}_T} F \), then there is an open saturated branch \( B \) for the constraint system corresponding to \( \text{KB} \cup \{ \neg F \} \) which is satisfiable in a minimal model of \( \text{KB} \), call it \( M \);

(ii) The model \( M' \) obtained from \( M \) by restricting its domain to the elements denoted by the labels in \( B \), and by renaming the elements of the domain with the names of the labels in \( B \) is also a minimal model of \( \text{KB} \) that satisfies \( B \); (iii) The canonical model \( M^B \) for \( B \) is a minimal model of \( \text{KB} \).

For (i), we show that the starting constraint system is satisfiable by \( M \) by an injective mapping (by the unique name assumption in Definition 5), and that each rule preserves the property. For (ii), since \( B \) is saturated, by reasoning by induction on the complexity of the formulas, it can be proven that \( M' \) (whose extension function and \( < \) coincide with those in \( M \) when restricted to the domain in \( M' \)) satisfies \( \text{KB} \). Furthermore, we can prove that \( M' \) is a minimal model of \( \text{KB} \).
suppose it was not. Then, there would be $M''$, model of KB, with $M'' < M'$. Consider then $M'''$ built by adding to $M''$ all the elements in $M$ that are not in $M''$. The < relation and the extension function in $M'''$ are defined as in $M''$ for the elements already present in $M''$. For the other elements, no < is introduced, and the extension function is defined as for some fixed $a$ in $M''$ such that there is no $b < a$ in $M''$ (by the Smoothness Condition in $M''$ such $a$ exists). It can be shown that $M'''$ satisfies KB, and that $M''' < M$, which contradicts the minimality of $M$. It follows that also $M'$ must be minimal. We then obtain the $M'$ of (ii) by simply renaming its elements with the names of the labels of which they are images. (iii) consider that $M'$ has the same domain as $M^B$. Furthermore, by Definition 25, for all $C \in L_T$ and for all labels $x$ occurring in $B$, either $x : \neg \Box C$ occurs in $B$ or $x : \Box \neg C$ occurs in $B$. Therefore, in $M'$ and in $M^B$, we have that $x \in (\neg \Box C)^1$ just in case $x : \neg \Box C$ occurs in $B$.

Hence, $M^\exists \neg = M^B \exists \neg$, and from the minimality of $M'$ we conclude that $M^B$ is minimal too.

Only if direction. The contrapositive easily follows: since $M^B$ is a minimal model of KB in which $F$ does not hold, by Definition 11 we conclude that KB $\not\models^L_T F$. □

Let us conclude this section by analyzing termination and complexity of $\mathcal{TAB}_A^{\mathcal{L}C+\mathcal{T}R}$. In general, non-termination in labelled tableau calculi can be caused by two different reasons: 1. some rules copy their principal formula in the conclusion(s), and can thus be reapplied over the same formula without any control; 2. dynamic rules may generate infinitely many labels, creating infinite branches. Similarly to the calculus $\mathcal{TAB}_A^{\mathcal{L}C+\mathcal{T}R}$ for $\mathcal{A}L\mathcal{C} + \mathcal{T} \mathcal{M} \mathcal{I} \mathcal{N}$ [GGOP13], we adopt the standard loop-checking machinery known as blocking to ensure termination.

Concerning the first source of non-termination (point 1), as mentioned above, all the rules copy their principal formulas in their conclusions. However, the side conditions on the application of the rules avoid multiple applications on the same formula. Indeed, $(\exists \subseteq)$ can be applied to a constraint system $(S \upharpoonright U, C \subseteq D^L)$ by using the label $x$ only if it has not yet been applied to $x$ in the current branch (i.e., $x$ does not belong to $L$).

Concerning $(\forall^+)$, the rule can be applied to $(S, x : \forall R.C, x \rightarrow y \upharpoonright U)$ only if $y : C$ does not belong to $S$. When $y : C$ is introduced in the branch, the rule will not further apply to $x : \forall R.C$. Similarly for $(<)$, $(\exists^+)$, $(\Box \neg)$, and the rules for $\mathcal{T}$, $\neg$, $\sqcap$ and $\sqcup$.

Concerning the second source of non-termination (point 2), we can prove that we only need to adopt the standard loop-checking machinery known as blocking, which ensures that the rules $(\exists^+)$ and $(\Box \neg)$ do not introduce infinitely many labels on a branch. Thanks to the properties of $\Box$, no other additional machinery would be required to ensure termination. Indeed, it can be shown that the interplay between rules $(\mathcal{T} \neg)$ and $(\Box \neg)$ does not generate branches containing infinitely many labels.

It is also worth noticing that the (cut) rule does not affect termination, since it is applied only to the finitely many formulas belonging to $L_T$.

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

As mentioned, another possible source of infinite branches could be determined by the interplay between rules $(T^-)$ and $(\square^-)$. However, even if we had no blocking on $(\square^-)$ this could not occur, i.e., the interplay between these two rules does not generate branches containing infinitely many labels. Intuitively, the application of $(\square^-)$ to $x : \neg \square \neg C$ adds $y : \neg \square \neg C$ to the conclusion, so that $(T^-)$ can no longer consistently introduce $y : \neg \square \neg C$. This is due to the properties of $\Box$ (no infinite descending chains of $<$ are allowed). More in detail, if $(\square)$ is applied to $T(C) \sqsubseteq D$ by using $x$, an application of $(T^-)$ introduces a branch containing $x : \neg \square \neg C$; when a new label $y$ is generated by an application of $(\square^-)$ on $x : \neg \square \neg C$, we have that $y : \square \neg C$ is added to the current constraint system. If $(\square)$ and $(T^-)$ are also applied to $T(C) \sqsubseteq D$ on the new label $y$, then the conclusion where $y : \neg \square \neg C$ is introduced is closed, by the presence of $y : \square \neg C$. By this fact, we would not need to introduce any loop-checking machinery on the application of $(\square^-)$. A detailed proof of termination of the calculus without blocking on $(\square^-)$ can be found in [GGOP09b]. However, in this paper we have introduced blocking also on $(\square^-)$ for complexity reasons.

In order to prove that the calculus $\mathcal{T}_{AB}^{ACC+Tr}$ ensures termination in a rigorous way, we need the following Lemma:

**Lemma 33.** Given a constraint system $\langle S \mid U \rangle$, let $n_{\langle S \mid U \rangle}$ be the number of extended concepts appearing in $\langle S \mid U \rangle$, including also all the concepts appearing as a substring of another concept. In any set of labels in $S$ including more than $2n_{\langle S \mid U \rangle}$ labels there are at least two labels $x$ and $y$ s.t. $x \equiv_S y$, i.e. there are at most $2^{n_{\langle S \mid U \rangle}}$ non-blocked labels.

**Proof.** Since there are $n_{\langle S \mid U \rangle}$ extended concepts, given a label $x$ there cannot be more than $2^{n_{\langle S \mid U \rangle}}$ different sets of constraints $x : C$ in $S$. As a consequence, in $S$ there are at most $2^{n_{\langle S \mid U \rangle}}$ non-blocked labels. \qed

**Theorem 34** (Termination of $\mathcal{T}_{AB}^{ACC+Tr}$). Let $\langle S \mid U \rangle$ be a constraint system, then any tableau generated by $\mathcal{T}_{AB}^{ACC+Tr}$ is finite.

**Proof.** First, we prove that only a finite number of labels can be introduced in a tableau. The only rules introducing a new label are dynamic rules. However, these rules are applicable only to formulas whose label is not blocked. By Lemma 33, there are at most $2^{n_{\langle S \mid U \rangle}}$ non-blocked labels in $\langle S \mid U \rangle$. Dynamic rules can be further applied to those $2^{n_{\langle S \mid U \rangle}}$ non-blocked labels, therefore obtaining at most $m \times 2^{n_{\langle S \mid U \rangle}}$ labels, where $m$ is the maximum number of labels directly generated by an application of a
dynamic rule from a label in \( S \). When \( m \times 2^{\ell(|U|)} \) labels belong to the constraint system, dynamic rules cannot be further applied.

Second, we prove that, since only a finite number of labels are introduced in a tableau, static rules can be applied only a finite number of times. Let us consider all the rules:

- \((\forall^+):\) the rule is applied to a constraint system of the form \( \langle S, x : \forall R.C, x \xrightarrow{R} y \mid U \rangle \), to obtain a conclusion of the form \( \langle S, x : \forall R.C, x \xrightarrow{R} y, y \mid U \mid C \rangle \). However, the side condition on the application of the rule imposes that the rule is applied if \( y : C \not\in S \), therefore it is applied only once in a branch, for a given \( \forall R.C \) and for two labels \( x \) and \( y \). Since only a finite number of labels as well as a finite number of formulas \( \forall R.C \) are introduced in a tableau (for the formulas, only (sub-)formulas of the initial KB or (sub-)formulas of \( \mathcal{L}_T \) or (sub-)formulas of the query), we can conclude that the rule \((\forall^+)\) is applied only a finite number of times;

- rules \((\langle \rangle), (\cap^+), (\cap^-), (\cup^+), (\cup^-), (\neg), (T^+), (T^-), (\exists^-):\) the application of these rules is restricted exactly as \((\forall^+)\), then we can conclude as we have done in the previous case;

- \((\text{cut}):\) just observe that it is applied by introducing \( x : (\neg)\Box \neg C \) for all concepts \( C \in \mathcal{L}_T \): since \( \mathcal{L}_T \) is finite, and we have to consider a finite number of labels \( x \), this rule is applied only a finite number of times;

- \((\subseteq):\) we can reason analogously to what done for \((\text{cut})\), since \((\subseteq)\) is applied to a finite set of subsumption relations \( C \subseteq D \in U \) by using a finite number of labels.

Since \( \mathcal{T}_{AB}^{ALC+TR_{PH_1}} \) is sound and complete (Theorem 30 and Theorem 31), and since a KB is satisfiable in an \( ALC + TR \) model iff its corresponding constraint system is satisfiable in the same model (Proposition 17), from Theorem 34 above it follows that checking whether a given KB (TBox,ABox) is satisfiable is a decidable problem.

Furthermore, we can prove that, with the calculus \( \mathcal{T}_{AB}^{ALC+TR_{PH_1}} \) above, the satisfiability of a KB can be decided in nondeterministic exponential time in the size of the KB.

**Theorem 35 (Complexity).** Given a KB and a query \( F \), checking whether KB \( \cup \{\neg F\} \) is satisfiable in an \( ALC + TR \) model can be solved in nondeterministic exponential time.

**Proof.** In order to check whether KB \( \cup \{\neg F\} \) is satisfiable w.r.t \( ALC + TR \), we build its corresponding constraint system \( \langle S \mid U \rangle \) and we try to build a tableau having \( \langle S \mid U \rangle \) as a root by means of the rules of \( \mathcal{T}_{AB}^{ALC+TR_{PH_1}} \). We first show that the number of labels generated on a branch is at most exponential in the size of KB \( \cup \{\neg F\} \). Let \( n \) be the size of KB \( \cup \{\neg F\} \). Given a constraint system \( \langle S \mid U \rangle \), the number of extended concepts appearing in \( \langle S \mid U \rangle \), including also all the ones appearing as a subformula of
other concepts, is $O(n)$. We have already shown in Lemma 33 that, as there are at most $O(n)$ concepts, there are at most $O(2^n)$ variables labelling distinct sets of concepts. Hence, there are $O(2^n)$ non-blocked variables in $S$.

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

For a given label $x$, the concepts labelled by $x$ introduced in the branch (namely, all the possible subconcepts of the initial constraint system, as well as all boxed subconcepts) are $O(n)$. Hence, the labelled concepts introduced on the branch is $O(n)$ for each label, and the number of all labelled concepts on the branch is $O(n \times 2^n)$. Since no rule deletes the principal formula to which it is applied, a branch can contain at most an exponential number of applications of tableau rules.

The satisfiability of $KB \cup \{ \neg F \}$ can thus be solved by defining a procedure which nondeterministically generates an open branch of $\mathcal{TAB}^{ALC+T_R}_{PH_1}$ of exponential size (in the size of $KB \cup \{ \neg F \}$). The problem is in $\text{NEXPTIME}$.

### 4.2 The tableau calculus $\mathcal{TAB}^{ALC+T_R}_{PH_2}$

Let us now introduce the calculus $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ which, for each open branch $B$ built by $\mathcal{TAB}^{ALC+T_R}_{PH_1}$, verifies if $M^B$ is a minimal model of the KB.

**Definition 36.** Given an open branch $B$ of a tableau built from $\mathcal{TAB}^{ALC+T_R}_{PH_1}$, we define:

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

A tableau of $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ is a tree whose nodes are triples of the form $(S \mid U \mid K)$, where $(S \mid U)$ is a constraint system, whereas $K$ contains formulas of the form $x : \neg \square \neg C$, with $C \in \mathcal{L}_T$.

The basic idea of $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ is as follows. Given an open branch $B$ built by $\mathcal{TAB}^{ALC+T_R}_{PH_1}$ and corresponding to a model $M^B$ of $KB \cup \{ \neg F \}$, $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ checks whether $M^B$ is a minimal model of KB by trying to build a model of KB which is preferred to $M^B$. Starting from $(S \mid U \mid B^{\square \neg})$ where $(S \mid U)$ is the constraint system corresponding to the initial KB $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ tries to build an open branch containing all and only the labels appearing on $B$, i.e. those in $D(B)$, and containing less negated boxed formulas than $B$ does. To this aim, first the dynamic rules use labels in $D(B)$ instead of introducing new ones in their conclusions. Second the negated boxed formulas used in $B$ are stored in the additional set $K$ of a tableau node, initialized with $B^{\square \neg}$. A branch built by $\mathcal{TAB}^{ALC+T_R}_{PH_2}$ closes if it does not represent a model preferred to the candidate model $M^B$, and this happens if the branch contains
a contradiction (Clash) or it contains at least all the negated boxed formulas contained in B ((Clash)_{\neg} and (Clash)_{\Delta}).

More in detail the rules of $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}_{P_{H2}}$ are shown in Figure 2. The rule (⊕+) is applied to a constraint system containing a formula $x : \exists R.C$; it introduces $x \xrightarrow{R} y$ and $y : C$ where $y \in \mathcal{D}(B)$, instead of $y$ being a new label. The choice of the label $y$ introduces a branching in the tableau construction. The rule (□) is applied in the same way as in $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}$ to all the labels of $\mathcal{D}(B)$ (and not only to those appearing in the branch). The rule (□) is applied to a node $(S, x : \neg \Box C \mid U \mid K)$, when $x : \neg \Box C \in K$, i.e., when the formula $x : \neg \Box C$ also belongs to the open branch $B$. In this case, the rule introduces a branch on the choice of the individual $v_i \in \mathcal{D}(B)$ which is preferred to $x$ and is such that $C$ and $\Box C$ hold in $v_i$. In case a tableau node has the form $(S, x : \neg \Box C \mid U \mid K)$, and $x : \neg \Box C \notin \mathcal{B}^\neg$, then $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}_{P_{H2}}$ detects a clash, called (Clash)$_{\neg}$. This corresponds to the situation in which $x : \neg \Box C$ does not belong to $B$, while $S, x : \neg \Box C$ is satisfiable in a model $M$ only if $M$ contains $x : \neg \Box C$, and hence only if $M$ is not preferred to the model represented by $B$.

The calculus $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}_{P_{H2}}$ also contains the clash condition (Clash)$_{\Delta}$. Since each application of (□) removes the principal formula $x : \neg \Box C$ from the set $K$, when $K$ is empty all the negated boxed formulas occurring in $B$ also belong to the current branch. In this case, the model built by $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}_{P_{H2}}$ satisfies the same set of negated boxed formulas (for all individuals) as $B$ and, thus, it is not preferred to the one repre-

Figure 2: The calculus $\mathcal{TAB}^{A_{LC}+T\mathcal{R}}_{P_{H2}}$. To save space, we omit the rules $(\sqcup^+)$ and $(\sqcap^+)$.
We can now prove that:

**Theorem 37** (Soundness and completeness of $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$). Given a KB and a query $F$, let $\langle S \mid U \rangle$ be the corresponding constraint system of $KB \cup \{-F\}$ and $\langle S' \mid U \rangle$ be the corresponding constraint system of $KB$. Given an open saturated branch $B$ built by $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$ for $\langle S \mid U \rangle$, the canonical model $M^B$ built from $B$ is a minimal model of $KB$ iff the tableau in $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$ for $\langle S' \mid U \mid B'\rangle$ is closed.

**Proof.** First, given an open branch $B$ built from $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$, by Theorem 31 and Proposition 17, $M^B = (\Delta_B, <', I^B)$ is a model of $KB$. In order to show the soundness (if direction), we show that if the tableau in $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$ for $\langle S' \mid U \mid B'\rangle$ is closed, then $M^B$ is a minimal model of $KB$. We show the contrapositive, that if $M^B$ was not minimal (i.e. if there was a model $M = (\Delta, <, I)$ of $KB$ s.t. $M <_{\mathcal{L}_T} M^B$ ) then there would be a branch in $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$ for $\langle S' \mid U \mid B'\rangle$ by showing that: (i) $\langle S' \mid U \rangle$ would be satisfiable in $M$ under the identity function $i$, (ii) each rule of the calculus preserves the satisfiability in $M$ under $i$, and (iii) no clash condition is satisfiable in such a model under $i$. (i) $M$ is a model of $KB$, and for all individual constants $a$ in the ABox $a^I = a$; by Proposition 17 it can be easily shown that also (i) holds. (ii) can be easily proven for all the rules. To save space, we only consider rules ($\square \neg$) and ($\exists^+$). The other rules are easy and then left to the reader. ($\square \neg$): suppose the premise $\langle S, x : \neg \square \neg C \mid U \mid K, x : \neg \square \neg C \rangle$ is satisfiable in $M$ under $i$, i.e. $x \in \neg \square \neg C^I$. Then there must be $v_i < x$ in $M$ with $v_i \in \Delta = \Delta_B = \mathcal{D}(B)$ such that $v_i \in C^I, v_i \in \square \neg C^I, v_i \in \neg C^I$, $v_i \in \square \neg C^I$ for all the formulas $v_i : \neg C^I, v_i : \square \neg C^I$ in $S^M_{x=v_i}$. It immediately follows that the conclusion of ($\square \neg$) containing $\langle S, v_i : C^I, v_i : \square \neg C^I, S^M_{x=v_i}, x : \neg \square \neg C \mid U \mid K \rangle$ is satisfiable in $M$ under $i$. ($\exists^+$): suppose the premise $\langle S, x : \exists R.C \mid U \mid K \rangle$ is satisfiable in $M$ under $i$, i.e. $x \in \exists R.C^I$. Then there is $v_i \in \Delta = \mathcal{D}(B)$ s.t. $(x, v_i) \in R^I$, and $v_i \in C^I$. The conclusion of the rule containing $\langle S, x \xrightarrow{R} v_i, v_i : C \mid U \mid K \rangle$ is therefore satisfiable in $M$ under $i$. (iii) clearly holds for (Clash), (Clash)$_\bot$ and (Clash)$_\top$. For (Clash)$_\beta$: if $K = \emptyset$, this means that rule $\square \neg$ (the only that removes formulas $x : \neg \square \neg C$ from $K$) has been applied to all $x : \neg \square \neg C$ in $B^\neg$, and all $x : \neg \square \neg C$ in $B^\neg$ are in $S$. However in this case the constraint system $\langle S \mid U \mid K \rangle$ in (Clash)$_\beta$ is not satisfiable in $M$ since by hypothesis $M^\square \neg \subset M^B^\square \neg$. Last (iii) holds for (Clash)$_\bot$, otherwise there would be a $\neg \square \neg C$ s.t. $M \models_i x : \neg \square \neg C$, i.e. $x \in (\neg \square \neg C)^I$ in $M$ but $x : \neg \square \neg C$ does not belong to $B^\neg$, i.e. $x \notin (\neg \square \neg C)^I$ in $M^B$, which contradicts that $M^\square \neg \subset M^B^\square \neg$.

We now consider the completeness (only if direction). By hypothesis, $M^B$ is a minimal model for $KB$. We want to show that the tableau in $\mathcal{TAB}^{\mathcal{ALC}^+T\mathcal{R}}_H$ for $\langle S' \mid U \mid B'\rangle$ is closed. For a contradiction, suppose that the tableau was open, with an open branch $B'$. It can be easily shown that the canonical model $M^{B'}$ built from $B'$ would be a model of $KB$ which is preferred to $M^B$. Indeed, the domain of $M^B$ coincides with that of $M^{B'}$ (which is $\mathcal{D}(B')$). Furthermore, $M^{B'}^\square \neg \subset M^B^\square \neg$, since the negated box formulas that hold in these canonical models are those that explicitly appear on the branch (by (cut) for all $C \in L_T$, for all labels $x$, either $x : \neg \square \neg C \in B'$ or $x : \neg \square \neg C \in B'$.)
B'), and $B'^\square^− \subset B^\square^−$, otherwise by (Clash) $B'$ would be closed. However, this contradicts the minimality of $M^B$. This contradiction forces us to conclude that there cannot be an open $B'$ in $\mathcal{T}AB^{ALC+TR}_{PH2}$, and that the tableau must be closed.

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

**Theorem 38** (Termination of $\mathcal{T}AB^{ALC+TR}_{PH2}$). Let $⟨S' \mid U \mid B^\square^−⟩$ be a constraint system starting from an open branch $B$ built by $\mathcal{T}AB^{ALC+TR}_{PH1}$, then any tableau generated by $\mathcal{T}AB^{ALC+TR}_{PH2}$ is finite.

**Proof.** Only a finite number of labels can occur on the tableau built by $\mathcal{T}AB^{ALC+TR}_{PH2}$, namely only those in $D(B)$ which is finite. Moreover, the side conditions on the application of the rules $(<), (\forall^+), (\subseteq), (\square^−)$, and (cut), copying their principal formulas in their conclusion(s), avoid the uncontrolled application of the same rules.

**Definition 39.** Let $KB$ be a knowledge base whose corresponding constraint system is $⟨S \mid U⟩$. Let $F$ be a query and let $S'$ be the set of constraints obtained by adding to $S$ the constraint corresponding to $¬F$. The calculus $\mathcal{T}AB^{ALC+TR}_{min}$ checks whether a query $F$ can be minimally entailed from a $KB$ by means of the following procedure:

- the calculus $\mathcal{T}AB^{ALC+TR}_{PH1}$ is applied to $⟨S' \mid U⟩$;
- if, for each branch $B$ built by $\mathcal{T}AB^{ALC+TR}_{PH1}$, either:
  - (i) $B$ is closed or
  - (ii) the tableau built by the calculus $\mathcal{T}AB^{ALC+TR}_{PH2}$ for $⟨S \mid U \mid B^\square^−⟩$ 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 40** (Soundness and completeness of $\mathcal{T}AB^{ALC+TR}_{min}$). $\mathcal{T}AB^{ALC+TR}_{min}$ is a sound and complete decision procedure for verifying if $KB \models_{\mathcal{L}^T_{min}} F$.

**Proof.** (Soundness) We show that if the procedure outputs YES, then $KB \models_{\mathcal{L}^T_{min}} F$ holds. If the procedure outputs YES, then for all branches generated by $\mathcal{T}AB^{ALC+TR}_{PH1}$ either they are closed or (ii) holds. If all branches generated by $\mathcal{T}AB^{ALC+TR}_{PH1}$ are closed then, by Theorem 30 we have that $KB \models_{ALC+TR} F$, then we conclude that $KB \models_{\mathcal{L}^T_{min}} F$. Consider now all open branches $B$ generated by $\mathcal{T}AB^{ALC+TR}_{PH1}$. Since the procedure outputs YES, then (ii) must hold for all $B$, i.e. the tableau built by $\mathcal{T}AB^{ALC+TR}_{PH2}$ for $⟨S \mid U \mid B^\square^−⟩$ is open. In this case, by Theorem 37, for all $B$, $M^B$ is not a minimal model of $KB$ and, by Theorem 32, $KB \models_{\mathcal{L}^T_{min}} F$ holds.

25
(Completeness) We show that if $\mathbf{KB} \models^T_{\mathcal{L}_{\text{min}}} F$ holds then the procedure outputs YES. First of all if all branches generated by $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{H_1}$ are closed, (i) holds for all branches and then the procedure outputs YES. Suppose now there are open branches $B$ generated by $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{H_1}$. Since $\mathbf{KB} \models^T_{\mathcal{L}_{\text{min}}} F$, by Theorem 32, $\mathcal{M}^B$ is not a minimal model of $\mathbf{KB}$ and by Theorem 37 the tableau in $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{H_2}$ for $\langle S' \mid U \mid B \square^- \rangle$ is open, hence (ii) holds and the procedure outputs YES.

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

**Theorem 41 (Complexity of $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}}^\mathcal{P}_{\mathcal{H}_1}$).** The problem of deciding whether $\mathbf{KB} \models^T_{\mathcal{L}_{\text{min}}} F$ is in $\text{CO-EXP}^\text{NP}$.

**Proof.** We first consider the complementary problem: $\mathbf{KB} \not\models^T_{\mathcal{L}_{\text{min}}} F$. This problem can be solved according to the procedure in Definition 39: by nondeterministically generating (NExp) an open branch of exponential length in the size of $\mathbf{KB}$ in $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{H_1}$ (a model $\mathcal{M}^B$ of $\mathbf{KB} \cup \{\neg F\}$), and then by calling an NP oracle which verifies that $\mathcal{M}^B$ is a minimal model of $\mathbf{KB}$. In fact, the verification that $\mathcal{M}^B$ is not a minimal model of the KB can be done by an NP algorithm which nondeterministically generates a branch in $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{H_2}$ (of polynomial size in the size of $\mathcal{M}^B$), representing a model $\mathcal{M}^{B'}$ of $\mathbf{KB}$ preferred to $\mathcal{M}^B$. Hence, the problem of verifying that $\mathbf{KB} \not\models^T_{\mathcal{L}_{\text{min}}} F$ is in NExp$^\text{NP}$, and the problem of deciding whether $\mathbf{KB} \models^T_{\mathcal{L}_{\text{min}}} F$ is in co-NExp$^\text{NP}$.}

By the above results, observe that if a formula is satisfiable in a minimal model, then there is a tableau in $\mathcal{T} \mathcal{A} \mathcal{B}^\mathcal{L}_{\mathcal{C}+\mathcal{T}_{\mathcal{R}}} P_{\mathcal{H}_1}$ containing a finite branch which is open in phase 1 and whose corresponding tableau in phase 2 is closed.

**References**


