Nonmonotonic Extensions of Low Complexity DLs: Complexity Results and Proof Methods

Laura Giordano[1], Valentina Gliozzi[2], Nicola Olivetti[3], and Gian Luca Pozzato[2]

[1] Dip. di Informatica - U. Piemonte O. - Alessandria - Italy - laura@mfn.unipmn.it
[2] Dip. Informatica - Univ. di Torino - Italy {gliozzi,pozzato}@di.unito.it
[3] LSIS-UMR CNRS 6168 - Marseille - France - nicola.olivetti@univ-cezanne.fr

Abstract. In this paper we propose nonmonotonic extensions of low complexity Description Logics $\mathcal{EL}^{\bot}$ and *DL-Lite*$_{core}$ for reasoning about typicality and defeasible properties. The resulting logics are called $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ and *DL-Lite*$_c\mathbf{T}_{min}$. We summarize complexity results for such extensions recently studied. Entailment in *DL-Lite*$_c\mathbf{T}_{min}$ is in Π^p_2, whereas entailment in $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ is EXPTIME-hard. However, considering the known fragment of Left Local $\mathcal{EL}^{\bot}\mathbf{T}_{min}$, we have that the complexity of entailment drops to Π^p_2. Furthermore, we present tableau calculi for $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ (focusing on Left Local knowledge bases) and *DL-Lite*$_c\mathbf{T}_{min}$. The calculi perform a two-phase computation in order to check whether a query is minimally entailed from the initial knowledge base. The calculi are sound, complete and terminating. Furthermore, they represent decision procedures for Left Local $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ knowledge bases and *DL-Lite*$_c\mathbf{T}_{min}$ knowledge bases, whose complexities match the above mentioned results.

1 Introduction

The family of description logics (DLs) is one of the most important formalisms of knowledge representation. They have a well-defined semantics based on first-order logic and offer a good trade-off between expressivity and complexity. DLs have been successfully implemented by a range of systems and they are at the base of languages for the semantic web such as OWL. A DL knowledge base (KB) comprises two components: the TBox, containing the definition of concepts (and possibly roles), and a specification of 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 arises.

Nonmonotonic extensions of Description Logics (DLs) have been actively investigated since the early 90s, [15, 4, 2, 3, 7, 12, 10, 9, 6]. A simple but powerful nonmonotonic extension of DLs is proposed in [12, 10, 9]: in this approach "typical" or "normal" properties can be directly specified by means of a "typicality" operator $\mathbf{T}$ enriching the underlying DL; the typicality operator $\mathbf{T}$ is essentially characterised by the core properties of nonmonotonic reasoning axiomatized by *preferential logic* [13]. In $\mathcal{ALC} + \mathbf{T}$ [12], one can consistently express defeasible inclusions and exceptions such as: typical students do not pay taxes, but working students do typically pay taxes, but working students having children normally do not: $\mathbf{T}(Student) \sqsubseteq \neg TaxPayer$; $\mathbf{T}(Student \sqcap Worker) \sqsubseteq TaxPayer$; $\mathbf{T}(Student \sqcap Worker \sqcap \exists HasChild.\top) \sqsubseteq \neg TaxPayer$. Although the operator $\mathbf{T}$ is nonmonotonic in itself, the logic $\mathcal{ALC} + \mathbf{T}$, as well as the logic $\mathcal{EL}^{+^\bot}\mathbf{T}$ [10] extending $\mathcal{EL}^\bot$, is monotonic. As a consequence, unless a KB contains explicit assumptions about typicality of individuals (e.g. that john is a typical student), there is no way of inferring defeasible properties of them (e.g. that john does not pay taxes). In [9], a non monotonic extension of $\mathcal{ALC}+\mathbf{T}$ based on a minimal model semantics is proposed. The resulting logic, called $\mathcal{ALC}+\mathbf{T}_{min}$, supports typicality assumptions, so that if one knows that john is a student, one can nonmonotonically assume that he is also a *typical* student and therefore that he does not pay taxes. As an example, for a TBox specified by the inclusions above, in $\mathcal{ALC}+\mathbf{T}_{min}$ the following inference holds: TBox $\cup \{Student(john)\} \models_{\mathcal{ALC}+\mathbf{T}_{min}} \neg TaxPayer(john)$.

Similarly to other nonmonotonic DLs, adding the typicality operator with its minimal-model semantics to a standard DL, such as $\mathcal{ALC}$, leads to a very high complexity (namely query entailment in $\mathcal{ALC}+\mathbf{T}_{min}$ is in co-NEXP$^{\text{NP}}$ [9]). This fact has motivated the study of nonmonotonic extensions of low complexity DLs such as *DL-Lite*$_{core}$ [5] and $\mathcal{EL}^\bot$ of the $\mathcal{EL}$ family [1] which are nonetheless well-suited for encoding large knowledge bases (KBs).

In this paper, we hence consider the extensions of the low complexity logics *DL-Lite*$_{core}$ and $\mathcal{EL}^\bot$ with the typicality operator based on the minimal model semantics introduced in [9]. We summarize complexity upper bounds for the resulting logics $\mathcal{EL}^\bot\mathbf{T}_{min}$ and *DL-Lite*$_c\mathbf{T}_{min}$ studied in [11]. For $\mathcal{EL}^\bot$, it turns out that its extension $\mathcal{EL}^\bot\mathbf{T}_{min}$ is unfortunately EXPTIME-hard. This result is analogous to the one for *circumscribed* $\mathcal{EL}^\bot$ KBs [3]. However, the complexity decreases to Π^p_2 for the fragment of *Left Local* $\mathcal{EL}^\bot$ KBs, corresponding to the homonymous fragment in [3]. The same complexity upper bound is obtained for *DL-Lite*$_c\mathbf{T}_{min}$.

We also present tableau calculi for *DL-Lite*$_c\mathbf{T}_{min}$ as well as for the Left Local fragment of $\mathcal{EL}^\bot\mathbf{T}_{min}$ for deciding minimal entailment in Π^p_2. Our calculi perform a two-phase computation: in the first phase, candidate models (complete open branches) falsifying the given query are generated, in the second phase the minimality of candidate models is checked by means of an auxiliary tableau construction. The latter tries to build a model which is “more preferred” than the candidate one: if it fails (being closed) the candidate model is minimal, otherwise it is not. Both tableaux constructions comprise some non-standard rules for existential quantification in order to constrain the domain (and its size) of the model being constructed. The second phase makes use in addition of special closure conditions to prevent the generation of non-preferred models. The calculi are very simple and do not require any blocking machinery in order to achieve termination. It comes as a surprise that the modification of the existential rule is sufficient to match the Π^p_2 complexity.

2 The typicality operator $\mathbf{T}$ and the Logic $\mathcal{EL}^\bot\mathbf{T}_{min}$

Before describing $\mathcal{EL}^\bot\mathbf{T}_{min}$, let us briefly recall the underlying monotonic logic $\mathcal{EL}^{+^\bot}\mathbf{T}$ [10], obtained by adding to $\mathcal{EL}^\bot$ the typicality operator $\mathbf{T}$. The intuitive idea is that $\mathbf{T}(C)$ selects the *typical* instances of a concept C. In $\mathcal{EL}^{+^\bot}\mathbf{T}$ we can therefore distinguish between the properties that hold for all instances of concept C ($C \sqsubseteq D$), and those that only hold for the normal or typical instances of C ($\mathbf{T}(C) \sqsubseteq D$).

Formally, the $\mathcal{EL}^{+^\bot}\mathbf{T}$ language is defined as follows.

Definition 1. *We consider an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individuals $\mathcal{O}$. Given $A \in \mathcal{C}$ and $R \in \mathcal{R}$, we define*

$$C := A \mid \top \mid \bot \mid C \sqcap C \qquad C_R := C \mid C_R \sqcap C_R \mid \exists R.C \qquad C_L := C_R \mid \mathbf{T}(C)$$

A KB is a pair (TBox, ABox). TBox contains a finite set of general concept inclusions (or subsumptions) $C_L \sqsubseteq C_R$. ABox contains assertions of the form $C_L(a)$ and $R(a,b)$, where $a, b \in \mathcal{O}$.

The semantics of $\mathcal{EL}^{+^\bot}\mathbf{T}$ [10] is defined by enriching ordinary models of $\mathcal{EL}^\bot$ by a *preference relation* $<$ on the domain, whose intuitive meaning is to compare the "typicality" of individuals: $x < y$, means that x is more typical than y. Typical members of a concept C, that is members of $\mathbf{T}(C)$, are the members x of C that are minimal with respect to this preference relation.

Definition 2 (Semantics of T). *A model $\mathcal{M}$ is any structure $\langle \Delta, <, I \rangle$ where Δ is the domain; $<$ is an irreflexive and transitive relation over Δ that satisfies the following* Smoothness Condition*: for all $S \subseteq \Delta$, for all $x \in S$, either $x \in Min_<(S)$ or $\exists y \in Min_<(S)$ such that $y < x$, where $Min_<(S) = \{u : u \in S$ and $\nexists z \in S$ s.t. $z < u\}$. Furthermore, $<$ is multilinear: if $u < z$ and $v < z$, then either $u = v$ or $u < v$ or $v < u$. I is the extension function that maps each concept C to $C^I \subseteq \Delta$, and each role r to $r^I \subseteq \Delta^I \times \Delta^I$. For concepts of $\mathcal{EL}^\bot$, C^I is defined in the usual way. For the $\mathbf{T}$ operator: $(\mathbf{T}(C))^I = Min_<(C^I)$.*

Given a model $\mathcal{M}$, I can be extended so that it assigns to each individual a of $\mathcal{O}$ a distinct element a^I of the domain Δ. We say that $\mathcal{M}$ satisfies an inclusion $C \sqsubseteq D$ if $C^I \subseteq D^I$, and that $\mathcal{M}$ satisfies $C(a)$ if $a^I \in C^I$ and aRb if $(a^I, b^I) \in R^I$. Moreover, $\mathcal{M}$ satisfies TBox if it satisfies all its inclusions, and $\mathcal{M}$ satisfies ABox if it satisfies all its formulas. $\mathcal{M}$ satisfies a KB (TBox,ABox), if it satisfies both its TBox and its ABox.

The operator $\mathbf{T}$ [12] is characterized by a set of postulates that are essentially a reformulation of the KLM [13] axioms of *preferential logic* **P**. $\mathbf{T}$ has therefore all the "core" properties of nonmonotonic reasoning as it is axiomatised by **P**. The semantics of the typicality operator can be specified by modal logic. The interpretation 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 $y < x$. Condition (ii) can be represented by means of an additional modality $\Box$, whose semantics is given by the preference relation $<$ interpreted as an accessibility relation. Observe that by the Smoothness Condition, $\Box$ has the properties of Gödel-Löb modal logic of provability G. The interpretation of $\Box$ in $\mathcal{M}$ is as follows: $(\Box C)^I = \{x \in \Delta \mid$ for every $y \in \Delta$, if $y < x$ then $y \in C^I\}$. We immediately get that $x \in (\mathbf{T}(C))^I$ if and only if $x \in (C \sqcap \Box\neg C)^I$. From now on, we consider $\mathbf{T}(C)$ as an abbreviation for $C \sqcap \Box\neg C$.

As mentioned in the Introduction, the main limit of $\mathcal{EL}^{+^\bot}\mathbf{T}$ is that it is *monotonic*. Even if the typicality operator $\mathbf{T}$ itself is nonmonotonic (i.e. $\mathbf{T}(C) \sqsubseteq E$ does not imply $\mathbf{T}(C \sqcap D) \sqsubseteq E$), what is inferred from an $\mathcal{EL}^{+^\bot}\mathbf{T}$ KB can still be inferred from any KB' with KB $\subseteq$ KB'. In order to perform nonmonotonic inferences, as done in [9], we strengthen the semantics of $\mathcal{EL}^{+^\bot}\mathbf{T}$ by restricting entailment to a class of minimal (or preferred) models. We call the new logic $\mathcal{EL}^\bot\mathbf{T}_{min}$. Intuitively, the idea is to restrict our consideration to models that *minimize the non typical instances of a concept*.

Given a KB, we consider a finite set $\mathcal{L}_{\mathbf{T}}$ of concepts: these are the concepts whose non typical instances we want to minimize. We assume that the set $\mathcal{L}_{\mathbf{T}}$ contains at least all concepts C such that $\mathbf{T}(C)$ occurs in the KB or in the query F, where a *query* F is either an assertion $C(a)$ or an inclusion relation $C \sqsubseteq D$. As we have just said, $x \in C^I$ is typical if $x \in (\Box\neg C)^I$. Minimizing the non typical instances of C therefore means to minimize the objects not satisfying $\Box\neg C$ for $C \in \mathcal{L}_{\mathbf{T}}$. Hence, for a given model $\mathcal{M} = \langle \Delta, <, I \rangle$, we define:

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

Definition 3 (Preferred and minimal models). *Given a model $\mathcal{M} = \langle \Delta <, I \rangle$ of a knowledge base KB, and a model $\mathcal{M}' = \langle \Delta', <', I' \rangle$ of KB, we say that $\mathcal{M}$ is preferred to $\mathcal{M}'$ with respect to $\mathcal{L}_{\mathbf{T}}$, and we write $\mathcal{M} <_{\mathcal{L}_{\mathbf{T}}} \mathcal{M}'$, if (i) $\Delta = \Delta'$, (ii) $\mathcal{M}^{\Box^-}_{\mathcal{L}_{\mathbf{T}}} \subset \mathcal{M}'^{\Box^-}_{\mathcal{L}_{\mathbf{T}}}$, (iii) $a^I = a^{I'}$ for all $a \in \mathcal{O}$. $\mathcal{M}$ is a* minimal model *for KB (with respect to $\mathcal{L}_{\mathbf{T}}$) if it is a model of KB and there is no other model $\mathcal{M}'$ of KB such that $\mathcal{M}' <_{\mathcal{L}_{\mathbf{T}}} \mathcal{M}$.*

Definition 4 (Minimal Entailment in $\mathcal{EL}^{\perp}\mathbf{T}_{min}$). *A query F is minimally entailed in $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ by KB with respect to $\mathcal{L}_{\mathbf{T}}$ if F is satisfied in all models of KB that are minimal with respect to $\mathcal{L}_{\mathbf{T}}$. We write KB $\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} F$.*

Example 1. The KB of the Introduction can be reformulated as follows in $\mathcal{EL}^{+^{\perp}}\mathbf{T}$: $TaxPayer \sqcap NotTaxPayer \sqsubseteq \perp$; $Parent \sqsubseteq \exists HasChild.\top$; $\exists HasChild.\top \sqsubseteq Parent$; $\mathbf{T}(Student) \sqsubseteq NotTaxPayer$; $\mathbf{T}(Student \sqcap Worker) \sqsubseteq TaxPayer$; $\mathbf{T}(Student \sqcap Worker \sqcap Parent) \sqsubseteq NotTaxPayer$. Let $\mathcal{L}_{\mathbf{T}} = \{Student, Student \sqcap Worker, Student \sqcap Worker \sqcap Parent\}$. Then TBox $\cup \{Student(john)\} \models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} NotTaxPayer(john)$, since $john^I \in (Student \sqcap \Box\neg Student)^I$ for all minimal models $\mathcal{M} = \langle \Delta <, I \rangle$ of the KB. In contrast, by the nonmonotonic character of minimal entailment, TBox $\cup \{Student(john), Worker(john)\} \models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} TaxPayer(john)$. Last, notice that TBox $\cup \{\exists HasChild.(Student \sqcap Worker)(jack)\} \models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} \exists HasChild.TaxPayer(jack)$. The latter shows that minimal consequence applies to implicit individuals *as well, without any ad-hoc mechanism.*

Theorem 1 (Complexity for $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ KBs (Theorem 3.1 in [11])). *The problem of deciding whether KB $\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} \alpha$ is* EXPTIME*-hard.*

In order to lower the complexity of minimal entailment in $\mathcal{EL}^{\perp}\mathbf{T}_{min}$, we consider a syntactic restriction on the KB called Left Local KBs. This restriction is similar to the one introduced in [3] for circumscribed $\mathcal{EL}^{\perp}$ KBs.

Definition 5 (Left Local knowledge base). *A Left Local KB only contains subsumptions $C_L^{LL} \sqsubseteq C_R$, where C and C_R are as in Definition 1 and:*

$C_L^{LL} := C \mid C_L^{LL} \sqcap C_L^{LL} \mid \exists R.\top \mid \mathbf{T}(C)$

There is no restriction on the ABox.

Observe that the KB in the Example 1 is Left Local, as no concept of the form $\exists R.C$ with $C \neq \top$ occurs on the left hand side of inclusions. In [11] an upper bound for the complexity of $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ Left Local KBs is provided by a small model theorem. Intuitively, what allows us to keep the size of the small model polynomial is that we reuse the same world to verify the same existential concept throughout the model. This allows us to conclude that:

Theorem 2 (Complexity for $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ Left Local KBs (Theorem 3.12 in [11])). *If KB is Left Local, the problem of deciding whether KB $\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} \alpha$ is in Π^p_2.*

3 The Logic *DL-Lite*$_c\mathbf{T}_{min}$

In this section we present the extension of the logic *DL-Lite*$_{core}$ [5] with the $\mathbf{T}$ operator. We call the resulting logic *DL-Lite*$_c\mathbf{T}_{min}$. The language of *DL-Lite*$_c\mathbf{T}_{min}$ is defined as follows.

Definition 6. *We consider an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individuals $\mathcal{O}$. Given $A \in \mathcal{C}$ and $r \in \mathcal{R}$, we define*

$$C_L := A \mid \exists R.\top \mid \mathbf{T}(A) \qquad R := r \mid r^- \qquad C_R := A \mid \neg A \mid \exists R.\top \mid \neg\exists R.\top$$

A DL-Lite$_c\mathbf{T}_{min}$ KB is a pair (TBox, ABox). TBox contains a finite set of concept inclusions of the form $C_L \sqsubseteq C_R$. ABox contains assertions of the form $C(a)$ and $r(a, b)$, where C is a concept C_L or C_R, $r \in \mathcal{R}$, and $a, b \in \mathcal{O}$.

As for $\mathcal{EL}^{\perp}\mathbf{T}_{min}$, a model $\mathcal{M}$ for *DL-Lite*$_c\mathbf{T}_{min}$ is any structure $\langle \Delta, <, I \rangle$, defined as in Definition 2, where I is extended to take care of inverse roles: given $r \in \mathcal{R}$, $(r^-)^I = \{(a, b) \mid (b, a) \in r^I\}$.

In [11] it has been shown that a small model construction similar to the one for Left Local $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ KBs can be made also for *DL-Lite*$_c\mathbf{T}_{min}$. As a difference, in this case, we exploit the fact that, for each atomic role r, the same element of the domain can be used to satisfy all occurrences of the existential $\exists r.\top$. Also, the same element of the domain can be used to satisfy all occurrences of the existential $\exists r^-.\top$.

Theorem 3 (Complexity for *DL-Lite*$_c\mathbf{T}_{min}$ KBs (Theorem 4.6 in [11])). *The problem of deciding whether KB $\models_{\textit{DL-Lite}_c\mathbf{T}_{min}} \alpha$ is in Π^p_2.*

4 The Tableau Calculus for Left Local $\mathcal{EL}^{\perp}\mathbf{T}_{min}$

In this section we present a tableau calculus $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{min}$ for deciding whether a query F is minimally entailed from a Left Local knowledge base in the logic $\mathcal{EL}^{\perp}\mathbf{T}_{min}$. It performs a two-phase computation: in the first phase, a tableau calculus, called $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$, simply verifies whether KB $\cup \{\neg F\}$ is satisfiable in an $\mathcal{EL}^{\perp}\mathbf{T}$ model, building candidate models; in the second phase another tableau calculus, called $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$, checks whether the candidate models found in the first phase are *minimal* models of KB, i.e. for each open branch of the first phase, $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ tries to build a model of KB which is preferred to the candidate model w.r.t. Definition 3. The whole procedure $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{min}$ is formally defined at the end of this section (Definition 8).

As usual, $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{min}$ tries to build an open branch representing a minimal model satisfying KB $\cup \{\neg F\}$. The negation of a query $\neg F$ is defined as follows: if $F \equiv C(a)$, then $\neg F \equiv (\neg C)(a)$; if $F \equiv C \sqsubseteq D$, then $\neg F \equiv (C \sqcap \neg D)(x)$, where x does not occur in KB. Notice that we introduce the connective $\neg$ in a very "localized" way. This is very different from introducing the negation all over the knowledge base, and indeed it does not imply that we jump out of the language of $\mathcal{EL}^{\perp}\mathbf{T}_{min}$.

$\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ makes use of labels, which are denoted with $x, y, z, \ldots$. Labels represent either a variable or an individual of the ABox, that is to say an element of $\mathcal{O} \cup \mathcal{V}$. These labels occur in *constraints* (or *labelled* formulas), that can have the form $x \xrightarrow{R} y$ or $x : C$, where x, y are labels, R is a role and C is either a concept or the negation of a concept of $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ or has the form $\Box\neg D$ or $\neg\Box\neg D$, where D is a concept.

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

4.1 First Phase: the tableaux calculus $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$

A tableau of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is a tree whose nodes are tuples $\langle S \mid U \mid W\rangle$. 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, used in order to ensure the termination of the tableau calculus. W is a set of labels x_C used in order to build a "small" model, matching the construction of Theorem 3.11 in [11]. A branch is a sequence of nodes $\langle S_1 \mid U_1 \mid W_1\rangle, \langle S_2 \mid U_2 \mid W_2\rangle, \ldots, \langle S_n \mid U_n \mid W_n\rangle \ldots$, where each node $\langle S_i \mid U_i \mid W_i\rangle$ is obtained from its immediate predecessor $\langle S_{i-1} \mid U_{i-1} \mid W_{i-1}\rangle$ by applying a rule of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$, having $\langle S_{i-1} \mid U_{i-1} \mid W_{i-1}\rangle$ as the premise and $\langle S_i \mid U_i \mid W_i\rangle$ as one of its conclusions. A branch is closed if one of its nodes is an instance of a (Clash) axiom, otherwise it is open. A tableau is closed if all its branches are closed.

The calculus $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is different in two respects from the calculus $\mathcal{ALC} + \mathbf{T}_{min}$ presented in [9]. First, the rule $(\exists^{+})$ is split in the following two rules:

$$\frac{\langle S, u : \exists R.C \mid U \mid W\rangle}{\langle S, u \xrightarrow{R} x_C, x_C : C \mid U \mid W \cup \{x_C\}\rangle \quad \langle S, u \xrightarrow{R} y_1, y_1 : C \mid U \mid W\rangle \ \cdots \ \langle S, u \xrightarrow{R} y_m, y_m : C \mid U \mid W\rangle}\,(\exists^{+})_1$$
$$\text{if } x_C \notin W \text{ and } y_1, \ldots, y_m \text{ are all the labels occurring in } S$$

$$\frac{\langle S, u : \exists R.C \mid U \mid W\rangle}{\langle S, u \xrightarrow{R} x_C \mid U \mid W\rangle \quad \langle S, u \xrightarrow{R} y_1, y_1 : C \mid U \mid W\rangle \cdots \langle S, u \xrightarrow{R} y_m, y_m : C \mid U \mid W\rangle}\,(\exists^{+})_2$$
$$\text{if } x_C \in W \text{ and } y_1, \ldots, y_m \text{ are all the labels occurring in } S$$

When the rule $(\exists^{+})_1$ is applied to a formula $u : \exists R.C$, it introduces a new label x_C only when the set W does not already contain x_C. Otherwise, since x_C has been already introduced in that branch, $u \xrightarrow{R} x_C$ is added to the conclusion of the rule rather than introducing a new label. As a consequence, in a given branch, $(\exists^{+})_1$ only introduces a new label x_C for each concept C occurring in the initial KB in some $\exists R.C$, and no blocking machinery is needed to ensure termination. As it will become clear in the proof of Theorem 4, this is possible since we are considering Left Local KBs, which have small models; in these models all existentials $\exists R.C$ occurring in KB are made true by reusing a single witness x_C (Theorem 3.12 in [11]). Notice also that the rules $(\exists^{+})_1$ and $(\exists^{+})_2$ introduce a branching on the choice of the label used to realize the existential restriction $u : \exists R.C$: just the leftmost conclusion of $(\exists^{+})_1$ introduces a new label (as mentioned, the x_C such that $x_C : C$ and $u \xrightarrow{R} x_C$ are added to the branch); in all the other branches, each one of the other labels y_i occurring in S may be chosen.

Second, in order to build multilinear models of Definition 2, the calculus adopts a strengthened version of the rule $(\Box^{-})$ used in $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ [9]. We write $\overline{S}$ as an abbreviation for $S, u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n$. Moreover, we define $S^M_{u \rightarrow y} = \{y : \neg D, y : \Box\neg D \mid u : \Box\neg D \in S\}$ and, for $k = 1, 2, \ldots, n$, we define $\overline{S}^{\Box^{-k}}_{u \rightarrow y} = \{y : \neg\Box\neg C_j \sqcup C_j \mid u : \neg\Box\neg C_j \in \overline{S} \wedge j \neq k\}$. The strengthened rule $(\Box^-)$ is as follows:

$$\frac{\langle S, u : \neg\Box\neg C_1, u : \neg\Box\neg C_2, \ldots, u : \neg\Box\neg C_n \mid U \mid W\rangle}{\langle S, x : C_k, x : \Box\neg C_k, S^M_{u \rightarrow x}, \overline{S}^{\Box^{-k}}_{u \rightarrow x} \mid U \mid W\rangle \quad \langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{u \rightarrow y_1}, \overline{S}^{\Box^{-k}}_{u \rightarrow y_1} \mid U \mid W\rangle \cdots \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{u \rightarrow y_m}, \overline{S}^{\Box^{-k}}_{u \rightarrow y_m} \mid U \mid W\rangle} \ (\Box^-)$$

for all $k = 1, 2, \ldots, n$, where $y_1, \ldots, y_m$ are all the labels occurring in S and x is new.

Rule $(\Box^-)$ contains: n branches, one for each $u : \neg\Box\neg C_k$ in $\overline{S}$; in each branch a *new* typical C_k individual x is introduced (i.e. $x : C_k$ and $x : \Box\neg C_k$ are added), and for all other $u : \neg\Box\neg C_j$, either $x : C_j$ holds or the formula $x : \neg\Box\neg C_j$ is recorded; - other $n \times m$ branches, where m is the number of labels occurring in S, one for each label y_i and for each $u : \neg\Box\neg C_k$ in $\overline{S}$; in these branches, a given y_i is chosen as a typical instance of C_k, that is to say $y_i : C_k$ and $y_i : \Box\neg C_k$ are added, and for all other $u : \neg\Box\neg C_j$, either $y_i : C_j$ holds or the formula $y_i : \neg\Box\neg C_j$ is recorded. This rule is sound with respect to multilinear models. The advantage of this rule over the $(\Box^-)$ rule in the calculus $\mathcal{TAB}^{\mathcal{ALC}+\mathbf{T}}_{min}$ is that all the negated box formulas labelled by u are treated in one step, introducing only a new label x in (some of) the conclusions. Notice that in order to keep $\overline{S}$ readable, we have used $\sqcup$. This is the reason why our calculi contain the rule for $\sqcup$, even if this constructor does not belong to $\mathcal{EL}^{\perp}\mathbf{T}_{min}$.

In order to check the satisfiability of a KB, we build its *corresponding constraint system* $\langle S \mid U \mid \emptyset\rangle$, and we check its satisfiability. Given *KB*=(TBox,ABox), its *corresponding constraint system* $\langle S \mid U \mid \emptyset\rangle$ is defined as follows: $S = \{a : C \mid C(a) \in ABox\} \cup \{a \xrightarrow{R} b \mid R(a, b) \in ABox\}$; $U = \{C \sqsubseteq D^{\emptyset} \mid C \sqsubseteq D \in TBox\}$.

Definition 7 (Model satisfying a constraint system). *Let $\mathcal{M} = \langle \Delta, I, < \rangle$ be a model as in Definition 2. We define a function α which assigns to each variable of $\mathcal{V}$ an element of Δ, and assigns every individual $a \in \mathcal{O}$ to $a^I \in \Delta$. $\mathcal{M}$ satisfies a constraint F under α, written $\mathcal{M} \models_\alpha F$, as follows: (i) $\mathcal{M} \models_\alpha x : C$ iff $\alpha(x) \in C^I$; (ii) $\mathcal{M} \models_\alpha x \xrightarrow{R} y$ iff $(\alpha(x), \alpha(y)) \in R^I$. A constraint system $\langle S \mid U \mid W\rangle$ 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 $x \in D^I$.*

Given a KB=(TBox,ABox), it is satisfiable if and only if its corresponding constraint system $\langle S \mid U \mid \emptyset\rangle$ is satisfiable. In order to verify the satisfiability of KB $\cup \{\neg F\}$, we use $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ to check the satisfiability of the constraint system $\langle S \mid U \mid \emptyset\rangle$ obtained by adding the constraint corresponding to $\neg F$ to S', where $\langle S' \mid U \mid \emptyset\rangle$ is the corresponding constraint system of KB. To this purpose, the rules of the calculus $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ are applied until either a contradiction is generated (Clash) or a model satisfying $\langle S \mid U \mid \emptyset\rangle$ can be obtained from the resulting constraint system.

Given a node $\langle S \mid U \mid W\rangle$, 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 *unfolding*. As mentioned above, each formula $C \sqsubseteq D$ is equipped with a list L of labels in which it has been unfolded in the current branch. This is needed to

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

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

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

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

$$\frac{\langle S, u : \exists R.C \mid U \mid W \rangle}{\langle S, u \xrightarrow{R} x_C, x_C : C \mid U \mid W \cup \{x_C\} \rangle \quad \langle S, u \xrightarrow{R} y_1, y_1 : C \mid U \mid W \rangle \cdots \langle S, u \xrightarrow{R} y_m, y_m : C \mid U \mid W \rangle}(\exists^+)_1$$

if $x_C \notin W$ and $y_1, \ldots, y_m$ are all the labels occurring in S

$$\frac{\langle S, u : \exists R.C \mid U \mid W \rangle}{\langle S, u \xrightarrow{R} x_C \mid U \mid W \rangle \quad \langle S, u \xrightarrow{R} y_1, y_1 : C \mid U \mid W \rangle \cdots \langle S, u \xrightarrow{R} y_m, y_m : C \mid U \mid W \rangle}(\exists^+)_2$$

if $x_C \in W$ and $y_1, \ldots, y_m$ are all the labels occurring in S

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

if $y : \neg C \notin S$

$$\frac{\langle S \mid U \mid W \rangle}{\langle S, x : \neg\Box\neg C \mid U \mid W \rangle \quad \langle S, x : \Box\neg C \mid U \mid W \rangle}(cut)$$

if $x : \neg\Box\neg C \notin S$ and $x : \Box\neg C \notin S$

x occurs in S $\quad C \in \mathcal{L}_{\mathbf{T}}$

$$\frac{\langle S, u : \neg\Box\neg C_1, u : \neg\Box\neg C_2, \ldots, u : \neg\Box\neg C_n \mid U \mid W \rangle}{\langle S, x : C_k, x : \Box\neg C_k, S^M_{u \to x}, \overline{S}^{\Box^{-k}}_{u \to x} \mid U \mid W \rangle \quad \langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{u \to y_1}, \overline{S}^{\Box^{-k}}_{u \to y_1} \mid U \mid W \rangle \cdots \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{u \to y_m}, \overline{S}^{\Box^{-k}}_{u \to y_m} \mid U \mid W \rangle}(\Box^-)$$

x new

if $y_1, \ldots, y_m$ are all the labels occurring in $S, y_1 \neq u, \ldots, y_m \neq u$

$k = 1, 2, \ldots, n$

Fig. 1. The calculus $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$.

avoid multiple unfolding of the same subsumption by using the same label, generating infinite branches.

Before introducing the rules of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ we need some more definitions. First, 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. Furthermore, if x is the label occurring in the query F, then $x \prec y$ for all y occurring in the constraint system corresponding to the initial KB. The rules of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ are presented in Figure 1. Rules $(\exists^+_1)$ and $(\Box^-)$ are called *dynamic* since they can introduce a new variable in their conclusions. The other rules are called *static*. We do not need any extra rule for the positive occurrences of the $\Box$ operator, since these are taken into account by the computation of $S^M_{x \to y}$ of $(\Box^-)$. The (cut) rule ensures that, given any concept $C \in \mathcal{L}_{\mathbf{T}}$, an open branch built by $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ contains either $x : \Box\neg C$ or $x : \neg\Box\neg C$ for each label x: this is needed in order to allow $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ to check the minimality of the model corresponding to the open branch.

The rules of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ 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. In [8] it has been shown that the calculus is sound and complete with respect to the semantics in Definition 7 and it ensures termination:

Theorem 4 (Soundness and completeness of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ [8]). *If KB $\not\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} F$, then the tableau for the constraint system corresponding to KB $\cup \{\neg F\}$ contains an open saturated branch, which is satisfiable (via an injective assignment from labels to domain elements) in a minimal model of KB. Given a constraint system $\langle S \mid U \mid W\rangle$, if it is unsatisfiable, then it has a closed tableau in $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$.*

Theorem 5 (Termination of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ [8]). *Any tableau generated by $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S \mid U \mid \emptyset\rangle$ is finite.*

Let us conclude this section by estimating the complexity of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$. Let n be the size of the initial KB, i.e. the length of the string representing KB, and let $\langle S \mid U \mid \emptyset\rangle$ be its corresponding constraint system. We assume that the size of F and $\mathcal{L}_{\mathbf{T}}$ is $O(n)$. The calculus builds a tableau for $\langle S \mid U \mid \emptyset\rangle$ whose branches's size is $O(n)$. This immediately follows from the fact that dynamic rules $(\exists^+)_1$ and $(\Box^-)$ generate at most $O(n)$ labels in a branch. Indeed, the rule $(\exists^+)_1$ introduces a new label x_C for each concept C occurring in KB, then at most $O(n)$ labels. Concerning $(\Box^-)$, consider a branch generated by its application to a constraint system $\langle S, u : \neg\Box\neg C_1 \dots, u : \neg\Box\neg C_n \mid U \mid W\rangle$. In the worst case, a new label x_1 is introduced. Suppose also that the branch under consideration is the one containing $x_1 : C_1$ and $x_1 : \Box\neg C_1$. The $(\Box^-)$ rule can then be applied to formulas $u : \neg\Box\neg C_k$, introducing also a further new label x_2. However, by the presence of $x_1 : \Box\neg C_1$, the rule $(\Box^-)$ can no longer consistently introduce $x_2 : \neg\Box\neg C_1$, since $x_2 : \Box\neg C_1 \in S^M_{x_1 \rightarrow x_2}$. Therefore, $(\Box^-)$ is applied to $\neg\Box\neg C_1 \dots \neg\Box\neg C_n$ in u. This application generates (at most) one new world x_1 that labels (at most) $n-1$ negated boxed formulas. A further application of $(\Box^-)$ to $\neg\Box\neg C_1 \dots \neg\Box\neg C_{n-1}$ in x_1 generates (at most) one new world x_2 that labels (at most) $n-2$ negated boxed formulas, and so on. Overall, at most $O(n)$ new labels are introduced by $(\Box^-)$ in each branch. For each of these labels, static rules apply at most $O(n)$ times: (Unfold) is applied at most $O(n)$ times for each $C \sqsubseteq D \in U$, one for each label introduced in the branch. The rule (cut) is also applied at most $O(n)$ times for each label, since $\mathcal{L}_{\mathbf{T}}$ contains at most $O(n)$ formulas. As the number of different concepts in KB is at most $O(n)$, in all steps involving the application of boolean rules, there are at most $O(n)$ applications of these rules. Therefore, the length of the tableau branch built by the strategy is $O(n^2)$. Finally, we observe that all the nodes of the tableau contain a number of formulas which is polynomial in n, therefore to test whether a node is an instance of a (Clash) axiom has at most complexity polynomial in n.

Theorem 6 (Complexity of $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$). *Given a KB and a query F, the problem of checking whether KB $\cup \{\neg F\}$ in $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is satisfiable is in* NP.

4.2 The tableaux calculus $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$

Let us now introduce the calculus $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ which, for each open branch **B** built by $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$, verifies whether it represents a minimal model of the KB. Given an open

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

$$\langle S \mid U \mid \emptyset \rangle \ (\text{Clash})_{\emptyset} \qquad \underset{\text{if } x : \neg\Box\neg C \notin K}{\langle S, x : \neg\Box\neg C \mid U \mid K \rangle \ (\text{Clash})_{\Box^-}} \qquad \frac{\langle S \mid U, C \sqsubseteq D^L \mid K \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \mid K \rangle} \ (\text{Unfold}) \ \ x \in \mathcal{D}(\mathbf{B}) \text{ and } x \notin L$$

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

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

$$\frac{\langle S, u : \exists R.C \mid U \mid K \rangle}{\langle S, u \xrightarrow{R} y_1, y_1 : C \mid U \mid K \rangle \ \cdots \ \langle S, u \xrightarrow{R} y_m, y_m : C \mid U \mid K \rangle} \ (\exists^+) \quad \text{if } \mathcal{D}(\mathbf{B}) = \{y_1, \ldots, y_m\}$$

$$\frac{\langle S, u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n \mid U \mid K, u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n \rangle}{\langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{u \to y_1}, \overline{S}^{\Box^{-k}}_{u \to y_1} \mid U \mid K \rangle \ \cdots \ \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{u \to y_m}, \overline{S}^{\Box^{-k}}_{u \to y_m} \mid U \mid K \rangle} \ (\Box^-)$$
$$\text{if } \mathcal{D}(\mathbf{B}) = \{y_1, \ldots, y_m\} \text{ and } y_1 \neq u, \ldots, y_m \neq u$$

Fig. 2. The calculus $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$. To save space, we omit the rule $(\sqcup^+)$.

branch $\mathbf{B}$ of a tableau built from $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH1}$, let $\mathcal{D}(\mathbf{B})$ be the set of labels occurring on $\mathbf{B}$. Moreover, let $\mathbf{B}^{\Box^-}$ be the set of formulas $x : \neg\Box\neg C$ occurring in $\mathbf{B}$, that is to say $\mathbf{B}^{\Box^-} = \{x : \neg\Box\neg C \mid x : \neg\Box\neg C$ occurs in $\mathbf{B}\}$.

A tableau of $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ is a tree whose nodes are tuples of the form $\langle S \mid U \mid K \rangle$, where S and U are defined as in 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 $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ is as follows. Given an open branch $\mathbf{B}$ built by $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH1}$ and corresponding to a model $\mathcal{M}^{\mathbf{B}}$ of KB $\cup \{\neg F\}$, $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ checks whether $\mathcal{M}^{\mathbf{B}}$ is a minimal model of KB by trying to build a model of KB which is preferred to $\mathcal{M}^{\mathbf{B}}$. To this purpose, it keeps track (in K) of the negated box used in $\mathbf{B}$ ($\mathbf{B}^{\Box^-}$) in order to check whether it is possible to build a model of KB containing less negated box formulas. The tableau built by $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ closes if it is not possible to build a model smaller than $\mathcal{M}^{B}$, it remains open otherwise. Since by Definition 3 two models can be compared only if they have the same domain, $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ tries to build an open branch containing all the labels appearing on $\mathbf{B}$, i.e. those in $\mathcal{D}(\mathbf{B})$. To this aim, the dynamic rules use labels in $\mathcal{D}(\mathbf{B})$ instead of introducing new ones in their conclusions. The rules of $\mathcal{TAB}^{\mathcal{EL}^{\bot}\mathbf{T}}_{PH2}$ are shown in Fig. 2.

More in detail, the rule $(\exists^+)$ is applied to a constraint system containing a formula $x : \exists R.C$; it introduces $x \xrightarrow{R} y$ and $y : C$ where $y \in \mathcal{D}(\mathbf{B})$, instead of y being a new label. The choice of the label y introduces a branching in the tableau construction. The rule (Unfold) is applied 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 $\langle S, u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n \mid U \mid K \rangle$, when $\{u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n\} \subseteq K$, i.e. when the negated box formulas

$u : \neg\Box\neg C_i$ also belong to the open branch **B**. Even in this case, the rule introduces a branch on the choice of the individual $y_i \in \mathcal{D}(\mathbf{B})$ to be used in the conclusion. In case a tableau node has the form $\langle S, x : \neg\Box\neg C \mid U \mid K\rangle$, and $x : \neg\Box\neg C \not\in K$, then $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ detects a clash, called (Clash)$_{\Box^-}$: this corresponds to the situation where $x : \neg\Box\neg C$ does not belong to **B**, while the model corresponding to the branch being built contains $x : \neg\Box\neg C$, and hence is *not* preferred to the model represented by **B**.

The calculus $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ also contains the clash condition (Clash)$_{\emptyset}$. Since each application of $(\Box^-)$ removes the negated box formulas $x : \neg\Box\neg C_i$ 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 $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ satisfies the same set of $x : \neg\Box\neg C_i$ (for all individuals) as **B** and, thus, it is not preferred to the one represented by **B**.

Theorem 7 (Soundness and completeness of $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ [8]). *Given a* KB *and a query F, let $\langle S' \mid U \mid \emptyset\rangle$ be the corresponding constraint system of* KB, *and $\langle S \mid U \mid \emptyset\rangle$ the corresponding constraint system of* KB $\cup \{\neg F\}$. *An open branch* **B** *built by $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S \mid U \mid \emptyset\rangle$ is satisfiable by an injective mapping in a minimal model of* KB *iff the tableau in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-}\rangle$ is closed.*

$\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ always terminates. 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.

Theorem 8 (Termination of $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$). *Let $\langle S' \mid U \mid \mathbf{B}^{\Box^-}\rangle$ be a constraint system starting from an open branch* **B** *built by $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$, then any tableau generated by $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is finite.*

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

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

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

Theorem 9 (Soundness and completeness of $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ [8]). *$\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is a sound and complete decision procedure for verifying if KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$.*

The complexity of $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ matches the results of Theorem 2. Consider the complementary problem: KB $\not\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$. This problem can be solved according to the procedure in Definition 8: by nondeterministically generating an open branch of polynomial length in the size of KB in $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ (a model $\mathcal{M}^{\mathbf{B}}$ of KB $\cup \{\neg F\}$), and then by calling an NP oracle which verifies that $\mathcal{M}^{\mathbf{B}}$ is a minimal model of KB. In fact, the verification that $\mathcal{M}^{\mathbf{B}}$ is not a minimal model of the KB can be done by an NP algorithm which nondeterministically generates a branch in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ of polynomial size in the size of $\mathcal{M}^{\mathbf{B}}$ (and of KB), representing a model $\mathcal{M}^{B'}$ of KB preferred to $\mathcal{M}^{\mathbf{B}}$. Hence, the problem of verifying that KB $\not\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$ is in NP^{NP}, i.e. in Σ_2^p, and the problem of deciding whether KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$ is in CO-NP^{NP}, i.e. in Π_2^p.

Theorem 10 (Complexity of $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$). *The problem of deciding whether* KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}}$ F *by means of* $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ *is in* Π_2^p.

5 A Tableau Calculus for *DL-Lite*$_c\mathbf{T}_{min}$

In this section we present a tableau calculus $\mathcal{TAB}_{min}^{Lite_c\mathbf{T}}$ for deciding query entailment in the logic *DL-Lite*$_c\mathbf{T}_{min}$. The calculus is similar to the one for $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ in the previous section, however it contains a few significant differences. Let us analyze in detail the two components of $\mathcal{TAB}_{min}^{Lite_c\mathbf{T}}$.

5.1 First Phase: the tableaux calculus $\mathcal{TAB}_{PH1}^{Lite_c\mathbf{T}}$

The calculus $\mathcal{TAB}_{PH1}^{Lite_c\mathbf{T}}$ is significantly different in three respects from the calculus for $\mathcal{EL}^{\perp}\mathbf{T}_{min}$. We try to explain such differences in detail. First of all, given a set of constraints S and a role $r \in \mathcal{R}$, we define $r(S) = \{x \xrightarrow{r} y \mid x \xrightarrow{r} y \in S\}$.

1. The rule $(\exists^+)$ is split in the following two rules:

$$\frac{\langle S, x : \exists r.\top \mid U\rangle}{\langle S, x \xrightarrow{r} y \mid U\rangle \quad \langle S, x \xrightarrow{r} y_1 \mid U\rangle \cdots \langle S, x \xrightarrow{r} y_m \mid U\rangle}\ (\exists^+)_1^r \quad \begin{array}{r}\text{if } r(S)=\emptyset\\ y \text{ new}\\ \text{if } y_1,\dots,y_m \text{ are all the labels occurring in } S\end{array}$$

$$\frac{\langle S, x : \exists r.\top \mid U\rangle}{\langle S, x \xrightarrow{r} y_1 \mid U\rangle \ \cdots \ \langle S, x \xrightarrow{r} y_m \mid U\rangle}\ (\exists^+)_2^r \quad \begin{array}{r}\text{if } r(S)\neq\emptyset\\ \text{if } y_1,\dots,y_m \text{ are all the labels occurring in } S\end{array}$$

As in the calculus $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$, the split of the $(\exists^+)$ in the two rules above reflects the main idea of the construction of a small model at the base of Theorem 4.5 in [11]. Such small model theorem essentially shows that *DL-Lite*$_c\mathbf{T}_{min}$ KBs can have small models in which all existentials $\exists R.\top$ occurring in KB are made true in the model by reusing a single witness y. In the calculus we use the same idea: when the rule $(\exists^+)_1^r$ is applied to a formula $x : \exists r.\top$, it introduces a new label y and the constraint $x \xrightarrow{r} y$ only when there is no other previous constraint $u \xrightarrow{r} v$ in S, i.e. $r(S) = \emptyset$. Otherwise, rule $(\exists^+)_2^r$ is applied and it introduces $x \xrightarrow{r} y$. As a consequence, $(\exists^+)_2^r$ does not introduce any new label in the branch whereas $(\exists^+)_1^r$ only introduces a new label y for each role r occurring in the initial KB in some $\exists r.\top$ or $\exists r^-.\top$, and no blocking machinery is needed to ensure termination.

2. In order to keep into account inverse roles, two further rules for existential formulas are introduced:

$$\frac{\langle S, x : \exists r^-.\top \mid U\rangle}{\langle S, y \xrightarrow{r} x \mid U\rangle \ \langle S, y_1 \xrightarrow{r} x \mid U\rangle \ \cdots \ \langle S, y_m \xrightarrow{r} x \mid U\rangle}\ (\exists^+)_1^{r^-} \quad \begin{array}{r}\text{if } r(S)=\emptyset\\ y \text{ new}\\ \text{if } y_1,\dots,y_m \text{ are all the labels occurring in } S\end{array}$$

$$\frac{\langle S, x : \exists r^-.\top \mid U\rangle}{\langle S, y_1 \xrightarrow{r} x \mid U\rangle \ \cdots \ \langle S, y_m \xrightarrow{r} x \mid U\rangle}\ (\exists^+)_2^{r^-} \quad \begin{array}{r}\text{if } r(S)\neq\emptyset\\ \text{if } y_1,\dots,y_m \text{ are all the labels occurring in } S\end{array}$$

These rules work similarly to $(\exists^+)^r_1$ and $(\exists^+)^r_2$ in order to build a branch representing a small model: when the rule $(\exists^+)^{r^-}_1$ is applied to a formula $x : \exists r^-.\top$, it introduces a new label y and the constraint $y \xrightarrow{r} x$ only when there is no other constraint $u \xrightarrow{r} v$ in S. Otherwise, since a constraint $y \xrightarrow{r} u$ has been already introduced in that branch, $y \xrightarrow{r} x$ is added to the conclusion of the rule.

3. Negated existential formulas can occur in a branch, but only having the form (i) $x : \neg\exists r.\top$ or (ii) $x : \neg\exists r^-.\top$. (i) means that x has no relationships with other individuals via the role r, i.e. we need to detect a contradiction if both (i) and, for some y, $x \xrightarrow{r} y$ belong to the same branch, in order to mark the branch as closed. The clash condition $(\text{Clash})_r$ is added to the calculus $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$ in order to detect such a situation. Analogously, (ii) means that there is no y such that y is related to x by means of r, then $(\text{Clash})_{r^-}$ is introduced in order to close a branch containing both (ii) and, for some y, a constraint $y \xrightarrow{r} x$. These clash conditions are as follows:

$$\langle S, x \xrightarrow{r} y, x : \neg\exists r.\top \mid U\rangle \ (\text{Clash})_r \qquad \langle S, y \xrightarrow{r} x, x : \neg\exists r^-.\top \mid U\rangle \ (\text{Clash})_{r^-}$$

The rules of $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$ are presented in Figure 3. The calculus $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$ is sound, complete and terminating.

$$\langle S, x : C, x : \neg C \mid U\rangle \ (\text{Clash}) \qquad \langle S, x \xrightarrow{r} y, x : \neg\exists r.\top \mid U\rangle \ (\text{Clash})_r \qquad \langle S, y \xrightarrow{r} x, x : \neg\exists r^-.\top \mid U\rangle \ (\text{Clash})_{r^-}$$

$$\frac{\langle S, x : \exists r.\top \mid U\rangle}{\langle S, x \xrightarrow{r} y \mid U\rangle \ \langle S, x \xrightarrow{r} y_1 \mid U\rangle \cdots \langle S, x \xrightarrow{r} y_m \mid U\rangle} \ (\exists^+)^r_1 \quad \begin{array}{r}\text{if } r(S) = \emptyset \\ y \text{ new} \\ \text{if } y_1, \dots, y_m \text{ are all the labels occurring in } S\end{array}$$

$$\frac{\langle S, x : \exists r.\top \mid U\rangle}{\langle S, x \xrightarrow{r} y_1 \mid U\rangle \cdots \langle S, x \xrightarrow{r} y_m \mid U\rangle} \ (\exists^+)^r_2 \quad \begin{array}{r}\text{if } r(S) \neq \emptyset \\ \text{if } y_1, \dots, y_m \text{ are all the labels occurring in } S\end{array}$$

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

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

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

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

$$\frac{\langle S, x : \exists r^-.\top \mid U\rangle}{\langle S, y \xrightarrow{r} x \mid U\rangle \ \langle S, y_1 \xrightarrow{r} x \mid U\rangle \cdots \langle S, y_m \xrightarrow{r} x \mid U\rangle} \ (\exists^+)^{r^-}_1 \quad \begin{array}{r}\text{if } r(S) = \emptyset \\ y \text{ new} \\ \text{if } y_1, \dots, y_m \text{ are all the labels occurring in } S\end{array}$$

$$\frac{\langle S, x : \exists r^-.\top \mid U\rangle}{\langle S, y_1 \xrightarrow{r} x \mid U\rangle \cdots \langle S, y_m \xrightarrow{r} x \mid U\rangle} \ (\exists^+)^{r^-}_2 \quad \begin{array}{r}\text{if } r(S) \neq \emptyset \\ \text{if } y_1, \dots, y_m \text{ are all the labels occurring in } S\end{array}$$

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

$$\frac{\langle S, x : \neg\Box\neg C_1, \dots, \neg\Box\neg C_n \mid U\rangle}{\langle S, y : C_k, y : \Box\neg C_k, S^M_{x\to y}, \overline{S}^{\Box^{-k}}_{x\to y} \mid U\rangle \ \langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{x\to y_1}, \overline{S}^{\Box^{-k}}_{x\to y_1} \mid U\rangle \cdots \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{x\to y_m}, \overline{S}^{\Box^{-k}}_{x\to y_m} \mid U\rangle} \ (\Box^-)$$
$$\begin{array}{r}y \text{ new} \\ \text{if } y_1, \dots, y_m \text{ are all the labels occurring in } S, y_1 \neq x, \dots, y_m \neq x \\ \forall k = 1, 2, \dots, n\end{array}$$

Fig. 3. The calculus $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$.

Theorem 11 (Soundness and completeness of $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$). *If KB $\not\models_{\text{DL-Lite}_c\mathbf{T}_{min}}$ F, then the tableau for the constraint system corresponding to KB $\cup \{\neg F\}$ contains an open saturated branch, which is satisfiable (via an injective assignment from labels to domain elements) in a minimal model of KB. Given a constraint system $\langle S \mid U\rangle$, if it is unsatisfiable, then it has a closed tableau in $\mathcal{TAB}_{PH1}^{\text{Lite}_c\mathbf{T}}$.*

Theorem 12 (Termination of $\mathcal{TAB}_{PH1}^{\mathit{Lite}_c\mathbf{T}}$). *Any tableau generated by $\mathcal{TAB}_{PH1}^{\text{Lite}_c\mathbf{T}}$ for $\langle S \mid U\rangle$ is finite.*

Reasoning as we have done for $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$, we can show that:

$$\langle S, x : C, x : \neg C \mid U \mid K \rangle \; (\text{Clash}) \qquad \langle S, x \xrightarrow{r} y, x : \neg \exists r.\top \mid U \mid K \rangle \; (\text{Clash})_r \qquad \langle S, y \xrightarrow{r} x, x : \neg \exists r^-.\top \mid U \mid K \rangle \; (\text{Clash})_{r^-}$$

$$\langle S \mid U \mid \emptyset \rangle \; (\text{Clash})_\emptyset \qquad \begin{array}{r}\langle S, x : \neg\Box\neg C \mid U \mid K \rangle \; (\text{Clash})_{\Box^-} \\ \text{if } x : \neg\Box\neg C \not\in K\end{array}$$

$$\frac{\langle S \mid U \mid K \rangle}{\langle S, x : \Box\neg C \mid U \mid K \rangle \quad \langle S, x : \neg\Box\neg C \mid U \mid K \rangle} \; (cut) \qquad \begin{array}{ll}\text{if } x : \neg\Box\neg C \not\in S \text{ and } x : \Box\neg C \not\in S \\ x \in \mathcal{D}(\mathbf{B}) & C \in \mathcal{L}_{\mathbf{T}}\end{array}$$

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

$$\frac{\langle S, x : \neg\Box\neg C_1, \dots, x : \neg\Box\neg C_n \mid U \mid K, x : \neg\Box\neg C_1, \dots, x : \neg\Box\neg C_n \rangle}{\langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{x \to y_1}, \overline{S}^{\Box^{-k}}_{x \to y_1} \mid U \mid K \rangle \;\; \cdots \;\; \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{x \to y_m}, \overline{S}^{\Box^{-k}}_{x \to y_m} \mid U \mid K \rangle} \; (\Box^-)$$
$$\begin{array}{r}\text{if } \mathcal{D}(\mathbf{B}) = \{y_1, \dots, y_m\} \text{ and } y_1 \neq x, \dots, y_m \neq x \\ \forall\, k = 1, 2, \dots, n\end{array}$$

$$\frac{\langle S \mid U, C \sqsubseteq D^L \mid K \rangle}{\langle S, x : \neg C \sqcup D \mid U, C \sqsubseteq D^{L,x} \mid K \rangle} \; (\text{Unfold}) \quad x \in \mathcal{D}(\mathbf{B}) \text{ and } x \not\in L \qquad \frac{\langle S, x : C \sqcup D \mid U \mid K \rangle}{\langle S, x : C \mid U \mid K \rangle \;\; \langle S, x : D \mid U \mid K \rangle} \; (\sqcup^+)$$

$$\frac{\langle S, x : \exists r.\top \mid U \mid K \rangle}{\langle S, x \xrightarrow{r} y_1 \mid U \mid K \rangle \dots \langle S, x \xrightarrow{r} y_m \mid U \mid K \rangle} \; (\exists^+)^r \quad \text{if } \mathcal{D}(\mathbf{B}) = \{y_1, \dots, y_m\}$$

$$\frac{\langle S, x : \exists r^-.\top \mid U \mid K \rangle}{\langle S, y_1 \xrightarrow{r} x \mid U \mid K \rangle \dots \langle S, y_m \xrightarrow{r} x \mid U \mid K \rangle} \; (\exists^+)^{r^-} \quad \text{if } \mathcal{D}(\mathbf{B}) = \{y_1, \dots, y_m\}$$

Fig. 4. The calculus $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$.

Theorem 13 (Complexity of $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH1}$). *Given a KB and a query F, the problem of checking whether KB $\cup \{\neg F\}$ in $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{PH1}$ is satisfiable is in* NP.

5.2 The tableaux calculus $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$

Let us now introduce the calculus $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$. Exactly as for $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$, for each open saturated branch $\mathbf{B}$ built by $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH1}$, it verifies whether it represents a minimal model of the KB. The rules of $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$ are shown in Figure 4. The rules $(\exists^+)^r$ and $(\exists^+)^{r^-}$ introduce $x \xrightarrow{r} y$ and $y \xrightarrow{r} x$, respectively, where $y \in \mathcal{D}(\mathbf{B})$, instead of y being a new label.

Theorem 14 (Soundness and completeness of $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$). *Given a* KB *and a query F, let $\langle S' \mid U \rangle$ be the corresponding constraint system of* KB, *and $\langle S \mid U \rangle$ the corresponding constraint system of* KB $\cup \{\neg F\}$. *An open saturated branch* $\mathbf{B}$ *built by $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{PH1}$ for $\langle S \mid U \rangle$ is satisfiable by an injective mapping in a minimal model of* KB *iff the tableau in $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{PH2}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-} \rangle$ is closed.*

Theorem 15 (Termination of $\mathcal{TAB}^{Lite_c\mathbf{T}}_{PH2}$). *Let $\langle S' \mid U \mid \mathbf{B}^{\Box^-} \rangle$ be a constraint system starting from an open saturated branch* $\mathbf{B}$ *built by $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{PH1}$, then any tableau generated by $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{PH2}$ is finite.*

By reasoning exactly as done for $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{min}$, we prove that:

Theorem 16 (Complexity of $\mathcal{TAB}^{Lite_c\mathbf{T}}_{min}$). *The problem of deciding whether* KB $\models^{\mathcal{L}_{\mathbf{T}}}_{min}$ *F by means of $\mathcal{TAB}^{\mathrm{Lite}_c\mathbf{T}}_{min}$ is in Π^p_2.*

6 Conclusions

We have proposed a nonmonotonic extension of low complexity DLs $\mathcal{EL}^{\bot}$ and *DL-Lite*$_{core}$ for reasoning about typicality and defeasible properties. We have summarized complexity results recently studied for such extensions [11], namely that entailment is EXPTIME-hard for $\mathcal{EL}^{\bot}\mathbf{T}_{min}$, whereas it drops to Π^p_2 when considering the Left Local Fragment of $\mathcal{EL}^{\bot}\mathbf{T}_{min}$. The same Π^p_2 complexity has been found for *DL-Lite*$_c\mathbf{T}_{min}$. These results match the complexity upper bounds of the same fragments in circumscribed KBs [3]. We have also provided tableau calculi for checking minimal entailment in the Left Local fragment of $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ as well as in *DL-Lite*$_c\mathbf{T}_{min}$. The proposed calculi match the complexity results above. Of course, many optimizations are possible and we intend to study them in future work.

As mentioned in the Introduction, several nonmonotonic extensions of DLs have been proposed in the literature [15, 4, 2, 3, 7, 12, 10, 9, 6] and we refer to [12] for a survey. Concerning nonmonotonic extensions of low complexity DLs, the complexity of *circumscribed* fragments of the $\mathcal{EL}^{\bot}$ and *DL-lite* families have been studied in [3]. Recently, a fragment of $\mathcal{EL}^{\bot}$ for which the complexity of circumscribed KBs is polynomial has been identified in [14]. In future work, we shall investigate complexity of minimal entailment and proof methods for such a fragment extended with $\mathbf{T}$ and possibly the definition of a calculus for it.

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, pages 364–369, 2005.
2. F. Baader and B. Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *JAR*, 15(1):41–68, 1995.
3. P. Bonatti, M. Faella, and L. Sauro. Defeasible inclusions in low-complexity dls: Preliminary notes. In *IJCAI*, pages 696–701, 2009.
4. P. A. Bonatti, C. Lutz, and F. Wolter. DLs with circumscription. In *KR*, p. 400–410, 2006.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable Reasoning and Efficient Query Answering in DLs: The DL-Lite Family. *JAR*, 39(3):385429, 2007.
6. G. Casini and U. Straccia. Rational closure for defeasible DLs. In *JELIA*, p. 77–90, 2010.
7. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
8. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. A tableau calculus for a nonmonotonic extension of $\mathcal{EL}^{\bot}$. In *TABLEAUX*, pages 164–179, 2011.
9. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning About Typicality in Preferential Description Logics. In *JELIA*, pages 192–205, 2008.
10. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Prototypical reasoning with low complexity Description Logics: Preliminary results. In *LPNMR*, pages 430–436, 2009.
11. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning about typicality in low complexity DLs: the logics $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ and *DL-Lite*$_c\mathbf{T}_{min}$. In *IJCAI*, pages 894–899, 2011.
12. L. Giordano, V. Gliozzi, N. Olivetti, and G.L. Pozzato. $\mathcal{ALC}+\mathbf{T}_{min}$: a preferential extension of description logics. *Fundamenta Informaticae*, 96:1–32, 2009.
13. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
14. P.A.Bonatti, M. Faella, and L. Sauro. $\mathcal{EL}$ with default attributes and overriding. In *ISWC*, pages 64–79, 2010.
15. U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *IJCAI*, pages 676–681, 1993.