A tableau calculus for a nonmonotonic extension of $\mathcal{EL}^{\perp}$

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. We introduce a tableau calculus for a nonmonotonic extension of low complexity Description Logic $\mathcal{EL}^{\perp}$ that can be used to reason about typicality and defeasible properties. The calculus deals with Left Local knowledge bases in the logic $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ recently introduced in [8] . The calculus performs a two-phase computation to check whether a query is minimally entailed from the initial knowledge base. It is sound, complete and terminating. Furthermore, it is a decision procedure for Left Local $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ knowledge bases, whose complexity matches the known results for the logic, namely that entailment is in Π^p_2.

1 Introduction

Nonmonotonic extensions of Description Logics (DLs) have been actively investigated since the early 90s, [14, 4, 2, 3, 7, 11, 10, 9, 6]. The reason is that DLs are used to represent classes and their properties, so that a nonmonotonic mechanism is wished to express defeasible inheritance of prototypical properties. A simple but powerful nonmonotonic extension of DL is proposed in [11, 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* [12]. In $\mathcal{ALC} + \mathbf{T}$ [11], 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 student 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 logics $\mathcal{ALC} + \mathbf{T}$ and $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ [10] are monotonic. As a consequence, unless a knowledge base (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 models semantics is proposed. The resulting logic, called $\mathcal{ALC} + \mathbf{T}_{min}$, supports typicality assumptions; 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 models semantics to a standard DL, such as $\mathcal{ALC}$, leads to a very high complexity (namely query entailment in the resulting logic is in CO-NEXP$^{\text{NP}}$ [9]). This fact has motivated the study of nonmonotonic extensions of low complexity DLs [3] such as $\mathcal{EL}^{\perp}$ of the $\mathcal{EL}$ family [1] which are nonetheless well-suited for encoding large KBs. In the same vein, we consider here the extension of the low complexity logic $\mathcal{EL}^{\perp}$ with the typicality operator based on the minimal models semantics introduced in [9]. But the restriction to $\mathcal{EL}^{\bot}$ does not suffice: as recently shown, deciding entailment in the resulting logic $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ is unfortunately EXPTIME-hard [8]. This result is analogous to the one for *circumscribed* $\mathcal{EL}^{\bot}$ KBs [3]. However, it has been shown in [8] that the complexity drops to Π_2^p for the fragment of *Left Local* $\mathcal{EL}^{\bot}\mathbf{T}_{min}$ KBs. Similar fragments have been previously studied for *circumscribed* $\mathcal{EL}^{\bot}$ KBs [3] obtaining the same complexity. To the best of our knowledge, however, no deduction calculi for these fragments with circumscription are known at present.

In this paper, we concentrate on the *Left Local* fragment of $\mathcal{EL}^{\bot}\mathbf{T}_{min}$. This fragment is determined by restricting the extistential quantification on concepts appearing on the *left* side of a concept inclusion: only existentially quantified concepts of the form $\exists R.\top$ are allowed. For this fragment, we propose a tableau calculus for deciding minimal entailment in Π_2^p. It is a two-phase calculus: 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. It comes as a surprise that the modification of the existential rule is sufficient to match the optimal complexity, so that the calculus provide in itself a constructive proof of the upper bound of this fragment (obtained in [8] by a non-constructive semantical argument).

2 The typicality operator T, the Logic $\mathcal{EL}^{\bot}\mathrm{T}_{min}$ and its Left Local fragment

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 $\mathcal{EL}^{\perp}$, C^I is defined in the usual way. For the $\mathbf{T}$ operator: $(\mathbf{T}(C))^I = Min_<(C^I)$.*

Definition 3 (Model satisfying a Knowledge Base). *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. $\mathcal{M}$ satisfies a KB (TBox,ABox), if it satisfies both its TBox and its ABox, where:*

- *$\mathcal{M}$ satisfies an inclusion $C \sqsubseteq D$ if $C^I \subseteq D^I$. $\mathcal{M}$ satisfies TBox if it satisfies all its inclusions.*
- *$\mathcal{M}$ satisfies $C(a)$ if $a^I \in C^I$ and aRb if $(a^I, b^I) \in R^I$. $\mathcal{M}$ satisfies ABox if it satisfies all its formulas.*

The operator $\mathbf{T}$ [11] is characterized by a set of postulates that are essentially a reformulation of KLM [12] 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:

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

We immediately get that $x \in (\mathbf{T}(C))^I$ iff $x \in (C \sqcap \Box\neg C)^I$. From now on, we consider $\mathbf{T}(C)$ as an abbreviation for $C \sqcap \Box\neg C$.

The main limit of $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ 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 $\mathbf{T}(C \sqcap D) \sqsubseteq E$), the logic $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ is monotonic: what is inferred from 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 $\mathcal{EL}^{+^{\perp}}\mathbf{T}$ by restricting entailment to a class of minimal (or preferred) models. We call the new logic $\mathcal{EL}^{\perp}\mathbf{T}_{min}$. 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 4 (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 5 (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 implict individuals *as well, without any ad-hoc mechanism.*

In [8] (Theorem 3.1), it has been proven that entailment in $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ 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 6 (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 [8] (Theorem 3.12), it has been proven that the problem of deciding whether KB $\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} F$ is in Π^p_2. In this paper, we focus our attention to Left Local KBs.

3 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 $\mathcal{EL}^{\perp}\mathbf{T}_{min}$.

The calculus $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{min}$ performs a two-phase computation in order to check whether a query F is minimally entailed from the initial KB. 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 4. The whole procedure $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{min}$ is formally defined at the end of this section (Definition 12).

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:

Definition 7 (Negation of a query). *Given a query F, we define its negation $\neg F$:*
- *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 $\mathcal{EL}^\perp\mathbf{T}_{min}$.

$\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{min}$ 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 $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{min}$, starting with $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{PH1}$.

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

A tableau of $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{PH1}$ 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 results of the Small Model Theorem in [8]. 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 $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{PH1}$, 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 $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{PH1}$ is significantly 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 1, 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 [8]). 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 $S^M_{u \to y} = \{y : \neg D, y : \Box\neg D \mid u : \Box\neg D \in S\}$ and, for $k = 1, 2, \ldots, n$, we define $\overline{S}^{\Box^{-k}}_{u \to y} = \{y : \neg\Box\neg C_j \sqcup C_j \mid u : \neg\Box\neg C_j \in \overline{S} \wedge j \neq k\}$. The strengthened rule $(\Box^-)$ is as follows:

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

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. As shown in the proof of Theorem 1, this rule is sound with respect to multilinear models. The advantage of this rule over the $(\Box^-)$ rule in the calculus $\mathcal{TAB}_{min}^{\mathcal{ALC}+\mathbf{T}}$ 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.

Definition 8 (Corresponding constraint system). *Given a knowledge base* KB=(*TBox*,*ABox*), *we define its* corresponding constraint system $\langle S \mid U \mid \emptyset\rangle$ *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 9 (Model satisfying a constraint system). *Let $\mathcal{M} = \langle \Delta, I, <\rangle$ be a model as defined 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:*

- $\mathcal{M} \models_\alpha x : C$ *iff* $\alpha(x) \in C^I$
- $\mathcal{M} \models_\alpha x \xrightarrow{R} y$ *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 $x \in D^I$.

Proposition 1. *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.*

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 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, as in [5], 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. Moreover, we define the satisfiability of a branch of a tableau:

Definition 10 (Satisfiability of a branch). *A branch* **B** *of a tableau of $\mathcal{TAB}^{\mathcal{EL}^\perp\mathbf{T}}_{PH1}$ is satisfiable by a model $\mathcal{M}$ if there is a mapping α from the labels in* **B** *to the domain of $\mathcal{M}$ such that for all constraint systems $\langle S \mid U \mid W \rangle$ on* **B**, *$\mathcal{M}$ satisfies under α (see Definition 9) every constraint in S and, for all $C \sqsubseteq D^L \in U$ and for all x occurring in S, we have that if $\alpha(x) \in C^I$ then $\alpha(x) \in D^I$.*

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

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

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

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

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

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

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

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

$$\frac{\langle S, u : \neg\Box\neg C_1, \neg\Box\neg C_2, \ldots, u : \neg\Box\neg C_n \mid U \mid W \rangle}{\langle S, x : C_k, x : \Box\neg C_k, S^M_{u \to x}, \overline{S}^{\Box^{-k}}_{u \to x} \mid U \mid W \rangle \quad \langle S, y_1 : C_k, y_1 : \Box\neg C_k, S^M_{u \to y_1}, \overline{S}^{\Box^{-k}}_{u \to y_1} \mid U \mid W \rangle \cdots \langle S, y_m : C_k, y_m : \Box\neg C_k, S^M_{u \to y_m}, \overline{S}^{\Box^{-k}}_{u \to y_m} \mid U \mid W \rangle}(\Box^-)$$
$$x \text{ new}$$
$$\text{if } y_1, \ldots, y_m \text{ are all the labels occurring in } S, y_1 \neq u, \ldots, y_m \neq u$$
$$k = 1, 2, \ldots, n$$

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

dynamic rules only if no static rule is applicable. The calculus so obtained is sound and complete with respect to the semantics in Definition 9.

Fact 1 *The only negated existential formulas that can occur in the tableau are (i) either general existential formulas $x : \neg\exists R.C$ that derive from the negation of the query; (ii) or $y : \neg\exists R.\top$, that can occur at any point of the branch and that derive from (Unfold) applied to a subsumption $\exists R.\top \sqsubseteq D$.*

Theorem 1 (Soundness of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$). *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 branch, which is satisfiable (via an injective assignment from labels to domain elements) in a minimal model of KB.*

Proof. (Sketch) If KB $\not\models_{\mathcal{EL}^{\perp}\mathbf{T}_{min}} F$, then there is a minimal model of KB that satisfies $\neg F$. By Definition 3, each individual occurring in KB is assigned to a different domain element (unique name assumption). It can be shown that this also holds for the constraint system corresponding to KB, which is therefore satisfiable by a minimal model of KB via an injective mapping from labels to domain elements. We show that each rule of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ preserves satisfiability by an injective mapping in a minimal model of KB. From this, we conclude that the branch is open, since no instance of the clash axioms would be satisfiable by an injective mapping in a minimal model of KB.

As an example of how we show that rules preserve satisfiability in a minimal model through an injective mapping, we consider here rules $(\exists^+)_1$, $(\exists^+)_2$, and $(\Box^-)$.

For $(\exists^+)_1$, assume there is a minimal model $\mathcal{M} = \langle \Delta, I, < \rangle$ of KB that satisfies the branch obtained before the application of the rule and premise of the rule (under an injective assignment α). In this model, $\alpha(u) \in (\exists R.C)^I$, i.e. there is $z \in \Delta$ with $(\alpha(u), z) \in R^I$ and $z \in C^I$. If the branch contains a label y_i such that $\alpha(y_i) = z$, then the consequence of the rule in which $u \xrightarrow{R} y_i, y_i : C$ appear is satisfiable by the same model under the same assignment. Otherwise, consider the branch containing $u \xrightarrow{R} x_C, x_C : C$. x_C is new, hence we can extend α so that $\alpha(x_C) = z$. α so extended is obviously still injective, and $\mathcal{M}$ satisfies this branch.

For $(\exists^+)_2$ the reasoning is a bit more tricky. Suppose the portion of the branch already obtained and the premise of the rule, $\langle S, u : \exists R.C \mid U \mid W \rangle$, are satisfiable by a minimal model $\mathcal{M}$ of KB under an assignment α. In $\mathcal{M}$, $(\alpha(u), z) \in R^I$ and $z \in C^I$, for some z. We reason by cases. (A) If the branch contains a label y_i such that $\alpha(y_i) = z$, then the consequence of the rule in which $u \xrightarrow{R} y_i, y_i : C$ appear is satisfiable by the same model under the same assignment. (B) Otherwise, we show that the conclusion containing $u \xrightarrow{R} x_C$ is satisfiable by a minimal model $\mathcal{M}'$ of KB, that also satisfies the previous portion of the branch. We distinguish two cases. (i) In the branch there is no occurrence of $\neg\exists R.D$ (for any D), then $\mathcal{M}'$ is obtained from $\mathcal{M}$ by simply adding $(\alpha(u), \alpha(x_C))$ to R^I. α has not been modified and therefore remains injective. Furthermore, since this addition does not modify the valuation function, and by this fact it does not modify the boxed formulas holding in the model, $\mathcal{M}'$ is still a minimal model of KB. Furthermore, $\mathcal{M}'$ satisfies the same branch formulas as $\mathcal{M}$, as well as $\langle S, u : \exists R.C \mid U \mid W \rangle$. (ii) In the branch there is an occurrence of $u : \neg\exists R.D$. Then by Fact 1, this is the only negated formula different from $\exists R.C$ occurring in the branch, and hence u is the starting label. Since $\mathcal{M}$ satisfies $u : \exists R.C$ under α, $(\alpha(u), z) \in R^I$, and $z \in C^I$. Consider now α', equal to α, apart from the fact that $\alpha'(x_C) = z$. α' is still injective. Indeed, the branch does not contain a label y_i such that $\alpha(y_i) = z$, otherwise we would be in case (A). It can be verified that $\mathcal{M}$ satisfies $\langle S, u : \exists R.C \mid U \mid W \rangle$, as well as the previous branch formulas, under α'. Indeed, by the strategy, the only formula labelled by x_C at the moment in which we apply $(\exists^+)_2$ is $x_C : C$ (we start considering x_C only after having applied all the rules to u, which is the starting label), and this is satisfied in $\mathcal{M}'$ under α'. All the other constraints that do not involve x_C remain satisfied even in α' that coincides with α on all other labels.

For $(\Box^-)$, we prove that if a node $\langle S, u : \neg\Box\neg C_1, \ldots, u : \neg\Box\neg C_n \mid U \mid W \rangle$ is satisfiable in a minimal multi-linear model $\mathcal{M} = \langle \mathcal{W}, <, V \rangle$ under a certain injective α then also one of the conclusions of the rule, namely $\langle S, x : C_k, x : \Box\neg C_k, S^M_{u \to x}, \overline{S}^{\Box^{-k}}_{u \to x} \mid U \mid W \rangle$ where x is a new label, is satisfiable in the same minimal model, under an extension of α. Let $\alpha(u) = u^I$, where $u^I \in \Delta$. There are $z_1 < u^I, \ldots, z_n < u^I$, such that $z_i \in Min_<(C_i)$, thus $z_i \in (C_i \sqcap \Box\neg C_i)^I$, for $i = 1, 2, \ldots, n$. Since $\mathcal{M}$ is a multi-linear model, the z_i, $i = 1, 2, \ldots, n$, whenever distinct, are totally ordered: we have that $z_i < u^I$, so that they must belong to the same component. Let

z_k be the maximum of z_i ($1 \leq i \leq n$), i.e. for each z_i, $i = 1, 2, \ldots, n$, we have either (i) $z_i = z_k$ or (ii) $z_i < z_k$. In case (i), we have that $z_k \in C_i^I$. In case (ii) we have that $z_k \in (\neg\Box\neg C_i)^I$. We have shown that for each $i \neq k$, $z_k \in C_i^I$ or $z_k \in (\neg\Box\neg C_i)^I$. If there is a label y on the branch such that $\alpha(y) = z_k$, then the conclusion $\langle S, y : C_k, y : \Box\neg C_k, S^M_{u \to y}, \overline{S}^{\Box^{-k}}_{u \to x} \mid U \mid W\rangle$ is satisfiable under an assignment α, which is injective. Indeed, we have that $\mathcal{M} \models_\alpha S^M_{u \to y}$ and $\mathcal{M} \models_\alpha \overline{S}^{\Box^{-k}}_{u \to y}$. Otherwise, since x does not occur in S, we extend α in a way such that $\alpha(x) = z_k$, and the conclusion $\langle S, x : C_k, x : \Box\neg C_k, S^M_{u \to x}, \overline{S}^{\Box^{-k}}_{u \to x} \mid U \mid W\rangle$ is satisfiable under the injective assignment α. ■

We can furthermore prove completeness of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$. The proof is quite straightforward and is omitted due to space limitations.

Theorem 2 (Completeness of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$). *Given a constraint system $\langle S \mid U \mid W\rangle$, if it is unsatisfiable, then it has a closed tableau in $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$.*

Let us conclude this section by analyzing termination and complexity of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$. 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. As mentioned above, differently from the calculus for $\mathcal{ALC} + \mathbf{T}_{min}$ in [11, 9], the calculus $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ ensures termination *without* adopting the standard blocking machinery.

Concerning the first source of non-termination (point 1), the only rules copying their principal formulas in their conclusions are $(\exists^-)$ and (Unfold). However, the side conditions on the application of such rules avoid multiple applications on the same formula. Concerning the second source of non-termination (point 2), we can prove that only finitely-many labels are introduced on a branch. Intuitively, the $(\exists^+)_1$ rule introduces at most one new label x_C for each concept C belonging to the initial node. Moreover, thanks to the properties of $\Box$, no other additional machinery is required to ensure termination. Indeed, it can be shown that the interplay between rules $(\mathbf{T}^-)$ and $(\Box^-)$ does not generate branches containing infinitely-many labels. Intuitively, the application of $(\Box^-)$ to $x : \neg\Box\neg C, x : \neg\Box\neg C_1, \ldots, x : \neg\Box\neg C_k$ adds $y : \Box\neg C$ to the conclusion, so that $(\mathbf{T}^-)$ can no longer consistently introduce $y : \neg\Box\neg C$. 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 $\mathcal{L}_{\mathbf{T}}$.

Theorem 3 (Termination of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$). *Let $\langle S \mid U \mid \emptyset\rangle$ be the corresponding constraint system of a KB. Any tableau generated by $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ for $\langle S \mid U \mid \emptyset\rangle$ is finite.*

Let us conclude this section by estimating the complexity of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$. Let n be the size of the initial KB, i.e. the length of the string representing KB, and let $\langle S \mid U \mid \emptyset\rangle$ its corresponding constraint system. We assume that the size of F and $\mathcal{L}_{\mathbf{T}}$ is $O(n)$.

Theorem 4 (Complexity of Phase 1). *Given a KB and a query F, the problem of checking whether KB $\cup\{\neg F\}$ is satisfiable is in* NP.

Proof. (Sketch) 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 the 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 $\langle S, u : \neg\Box\neg C_1 \ldots, u : \neg\Box\neg C_n \mid U \mid W \rangle$. 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 $\neg\Box\neg C_1 \ldots \neg\Box\neg C_n$ 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 $\neg\Box\neg C_1 \ldots \neg\Box\neg C_{n-1}$ 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 that a node is an instance of a (Clash) axiom has at most complexity polynomial in n. ■

Notice that the above strategy is able to build branches of polynomial length thanks to the presence of the rule (cut). Indeed, the key point is that, when the rule $(\Box^-)$ building multilinear models is applied to a given label u, *all negated boxed formulas* $u : \neg\Box\neg C_k$ belong to current set of formulas. It could be the case that, after an application of $(\Box^-)$ by using u, the same label u is used in one of the conclusions of another application of $(\Box^-)$, say to some x_i. Therefore, the application of static rules could introduce $u : \neg\Box\neg C$, and a further application of $(\Box^-)$ could be needed. However, since (cut) is a static rule, and since $C \in \mathcal{L}_{\mathbf{T}}$ because $\neg\Box\neg C$ has been generated by unfolding some $\mathbf{T}(C) \sqsubseteq D$ in the TBox, either $u : \neg\Box\neg C$ or $u : \Box\neg C$ have already been introduced in the branch *before* the second application of $(\Box^-)$, which is a dynamic rule.

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

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

Definition 11. *Given an open branch* **B** *of a tableau built from* $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$*, we define:*
- $\mathcal{D}(\mathbf{B})$ *as the set of labels occurring on* **B***;*
- $\mathbf{B}^{\Box^-} = \{x : \neg\Box\neg C \mid x : \neg\Box\neg C$ *occurs in* $\mathbf{B}\}$.

A tableau of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ 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

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

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

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

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

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

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

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

of the form $x : \neg\Box\neg C$, with $C \in \mathcal{L}_{\mathbf{T}}$. The basic idea of $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ is as follows. Given an open branch **B** built by $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH1}$ and corresponding to a model $\mathcal{M}^{\mathbf{B}}$ of KB $\cup \{\neg F\}$, $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ 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 **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 $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ closes if it is not possible to build a model smaller than $\mathcal{M}^B$, it remains open otherwise. Since by Definition 4 two models can be compared only if they have the same domain, $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ tries to build an open branch containing all the labels appearing on **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 $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ 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 $x : \neg\Box\neg C \notin K$, then $\mathcal{TAB}^{\mathcal{EL}^{\perp}\mathbf{T}}_{PH2}$ 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**.

Let us now analyze soundness and completeness of $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$. First, given a branch **B**, we associate with **B** a relation $<$ defined as follows: $y < u$ if y is the label chosen in the conclusion of the application of the rule $(\Box^-)$ to $u : \neg\Box\neg C$. We define a canonical model $\mathcal{M}^B$ for **B** as follows: $\mathcal{M}^B = \langle \Delta_B, <', I\rangle$ where: - $\Delta_B = \{x : x$ is a label appearing in **B**$\}$; - $<'$ is the transitive closure of relation $<$ associated with **B**; - I is an interpretation function such that for all atomic concepts A, $A^I = \{x$ such that $x : A$ occurs in **B**$\}$. I is then extended to all concepts C in the standard way, according to the semantics of the operators. For role names R, $R^I = \{(x, y) : x \xrightarrow{R} y$ occurs in **B**$\}$.

Lemma 1. *If a branch* **B** *is satisfiable by an injective mapping in a minimal model of KB, then the canonical model* $\mathcal{M}^B$ *for* **B** *is a minimal model of KB satisfying* **B**.

Theorem 5 (Soundness and completeness of $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$). *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.*

Proof. In order to show the soundness (if direction), we show that if the tableau in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-}\rangle$ is closed, then $\mathcal{M}^B$ (which by Theorem 2 is a model of $\mathbf{B}$) is a minimal model of KB that satisfies **B** (hence **B** is satisfiable by an injective mapping in a minimal model of KB). We show the contrapositive, that if $\mathcal{M}^{\mathbf{B}}$ was not minimal (i.e. if there was a model $\mathcal{M}$ of KB with same domain as $\mathcal{M}^{\mathbf{B}}$ but with $\mathcal{M}^{\Box^-} \subset \mathcal{M}^{B^{\Box^-}}$) then there would be an open branch in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ by showing that: (i)$\langle S' \mid U \mid \mathbf{B}^{\Box^-}\rangle$ would be satisfiable in $\mathcal{M}$, (ii) each rule of the calculus preserves the satisfiability in $\mathcal{M}$, and (iii) that no clash condition is satisfiable in such a model.

We now consider the completeness (only if direction). By hypothesis **B** is satisfiable by an injective mapping in a minimal model for KB. By Lemma 1, $\mathcal{M}^B$ is a minimal model of KB satisfying **B**. We want to show that the tableau in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-}\rangle$ 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 for **B'**, $\mathcal{M}^{B'}$, is still a model of KB which is preferred to $\mathcal{M}^B$. Indeed, the domain of $\mathcal{M}^B$ coincides with that of $\mathcal{M}^{B'}$ (which is $\mathcal{D}(\mathbf{B})$). Clearly, $\mathcal{M}_{\mathcal{L}_\mathbf{T}}^{B'^{\Box^-}} \subset \mathcal{M}_{\mathcal{L}_\mathbf{T}}^{B^{\Box^-}}$, since $\mathbf{B'}^{\Box^-} \subset \mathbf{B}^{\Box^-}$, otherwise by (Clash)$_\emptyset$ **B'** would be closed, and by (cut) for all $C \in \mathcal{L}_\mathbf{T}$, for all labels x, either $x : \Box\neg C \in$ **B'** or $x : \neg\Box\neg C \in$ **B'** . Hence, $\mathcal{M}^{B'}$ would be preferred to $\mathcal{M}^B$, against the minimality of $\mathcal{M}^B$. This contradiction forces us to conclude that there cannot be an open **B'** in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$, and that the tableau must be 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 6 (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 12. *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 7 (Soundness and completeness of $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$). *$\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is a sound and complete decision procedure for verifying if KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$.*

Proof. (Soundness) If KB $\not\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$, and KB $\cup \{\neg F\}$ is satisfiable by a minimal model of KB, then by Theorem 1, $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$ generates an open branch, which is satisfiable (via an injective assignment from labels to domain elements) in a minimal model of KB. By Theorem 5 for this branch the tableau in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ is closed. In this case, (i) and (ii) in Definition 12 do not hold, and the procedure correctly says that KB $\not\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$. (Completeness) Let KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$. For contraposition, let **B** be an open branch (if any) generated by $\mathcal{TAB}_{PH1}^{\mathcal{EL}^{\perp}\mathbf{T}}$. If this branch were satisfiable by an injective mapping in a minimal model of KB, then by Proposition 1, also KB $\cup \{\neg F\}$ would be, against the hypothesis that KB $\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$. Hence, **B** is not satisfiable by an injective mapping in a minimal model of KB, and by Theorem 5 the tableau in $\mathcal{TAB}_{PH2}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for $\langle S' \mid U \mid \mathbf{B}^{\Box^-} \rangle$ is open. ■

We can also prove that the complexity of $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ matches the known results for minimal entailment in Left Local $\mathcal{EL}^{\perp}\mathbf{T}_{min}$:

Theorem 8 (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.*

Proof. We first consider the complementary problem: KB $\not\models_{min}^{\mathcal{L}_{\mathbf{T}}} F$. This problem can be solved according to the procedure in Definition 12: 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. ■

4 Conclusions

In this work we have provided a two-phase tableau calculus $\mathcal{TAB}_{min}^{\mathcal{EL}^{\perp}\mathbf{T}}$ for checking minimal entailment in a nonmonotonic extension of the Left Local fragment of the logic $\mathcal{EL}^{\perp}\mathbf{T}_{min}$, a family of low complexity DLs $\mathcal{EL}^{\perp}$. The proposed calculus matches the known complexity results for such DL, namely that entailment is in Π_2^p [8]. 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 [14, 4, 2, 3, 7, 11, 10, 9, 6] and we refer to [11] for a survey. Concerning nonmonotonic extensions of low complexity DLs, the complexity of *circumscribed* fragments of the $\mathcal{EL}^{\perp}$ and DL-lite families have been studied in [3]. The contribution of this paper is to provide a calculus for the Left Local fragment of $\mathcal{EL}^{\perp}$ under minimal entailment. We expect that our tableau calculus can also be adapted to deal with the DL-lite$_c\mathbf{T}$ fragment, for which a Π_2^p upper bound has been proved in [8]. Recently, a fragment of $\mathcal{EL}^{\perp}$ for which the complexity of circumscribed KBs is polynomial has been identified in [13]. In future work, we shall investigate complexity of minimal entailment and proof methods for such a fragment extended with $\mathbf{T}$.

Acknowledgements. This work has been partially supported by the Project "MIUR PRIN08 LoDeN: Logiche Descrittive Nonmonotone: Complessitá e implementazioni".

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. *J. of Automated Reasoning (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. Description logics with circumscription. In *KR*, pages 400–410, 2006.
5. M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. Artif. Int. Research (JAIR)*, 1:109–138, 1993.
6. G. Casini and U. Straccia. Rational closure for defeasible description logics. In *JELIA*, pages 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. Reasoning about typicality in low complexity DLs: the logics $\mathcal{EL}^{\perp}\mathbf{T}_{min}$ and DL-lite$_c\mathbf{T}_{min}$. In *To appear in IJCAI 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*, 2009.
11. 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.
12. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
13. P.A.Bonatti, M. Faella, and L. Sauro. $\mathcal{EL}$ with default attributes and overriding. In *ISWC*, pages 64–79, 2010.
14. U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *IJCAI*, pages 676–681, 1993.