Deciding $\mathcal{ALBO}$ with Tableau

Renate A. Schmidt[1] and Dmitry Tishkovsky[1]

School of Computer Science, The University of Manchester
{renate.schmidt,dmitry.tishkovsky}@manchester.ac.uk

Abstract. This paper presents a tableau approach for deciding description logics outside the scope of OWL DL and current state-of-the-art tableau-based description logic systems. In particular, we define a sound and complete tableau calculus for the description logic $\mathcal{ALBO}$ and show that it provides a basis for decision procedures for this logic and numerous other description logics. $\mathcal{ALBO}$ is the extension of $\mathcal{ALC}$ with the Boolean role operators, inverse of roles, domain and range restriction operators and it includes full support for objects (nominals). $\mathcal{ALBO}$ is a very expressive description logic which is NExpTime complete and subsumes Boolean modal logic and the two-variable fragment of first-order logic. An important novelty is the use of a versatile, unrestricted blocking rule as a replacement for standard loop checking mechanisms implemented in description logic systems. Our decision procedure is implemented in the MetTeL system.

1 Introduction

The description logic $\mathcal{ALBO}$ is an extension of the description logic $\mathcal{ALB}$ introduced in [7] with singleton concepts, called nominals in modal logic. $\mathcal{ALB}$ is the extension of $\mathcal{ALC}$, in which concepts and roles form a Boolean algebra, and additional operators include inverse of roles and a domain restriction operator. $\mathcal{ALBO}$ extends $\mathcal{ALC}$ by union of roles, negation of roles, inverse of roles, and domain as well as range restriction. In addition, it provides full support for ABox objects and singleton concepts.

None of the current state-of-the-art tableau-based description logic systems are able to handle $\mathcal{ALBO}$. Because $\mathcal{ALBO}$ allows full negation of roles, it is out of the scope of OWL DL and most description logic systems including Fact++, KAON2, Pellet, and RacerPro. A tableau decision procedure for the description logic $\mathcal{ALCQIb}$ which allows for Boolean combinations of 'safe' occurrences of negated roles is described in [14]. Safeness essentially implies a 'guardedness' property which is violated for unsafe occurrences of role negation. Description logics with full, i.e. safe and unsafe, role negation can be decided however by translation to first-order logic and first-order resolution theorem provers such as MSpass, Spass and Vampire. The paper [7] shows that the logic $\mathcal{ALB}$ can be decided by translation to first-order logic and ordered resolution. This result is extended in [3] to $\mathcal{ALB}$ with positive occurrences of composition of roles. $\mathcal{ALBO}$ can be embedded into the two-variable fragment

of first-order logic with equality which can be decided with first-order resolution methods [2]. This means that $\mathcal{ALBO}$ is decidable and can be decided using first-order resolution methods.

$\mathcal{ALBO}$ is a very expressive description logic. It subsumes the Boolean modal logic [4, 5] and tense, hybrid versions of Boolean modal logic with the @ operator and nominals. $\mathcal{ALBO}$ can also be shown to subsume the two-variable fragment of first-order logic (without equality) [8]. The following constructs and statements can be handled in $\mathcal{ALBO}$.

- Role negation, the universal role, the sufficiency or window operator, Peirce sum, domain restriction, cross product, and cylindrification.
- Role inclusion axioms and role equivalence axioms in the language of $\mathcal{ALBO}$.
- Role assertions in the language of $\mathcal{ALBO}$.
- Boolean combinations of both concept and role inclusion and equivalence axioms.
- Boolean combinations of concept and role assertions, including negated role assertions.
- Disjoint roles, symmetric roles and serial roles.[1]

Since $\mathcal{ALBO}$ subsumes Boolean modal logic it follows from [10] that the satisfiability problem in $\mathcal{ALBO}$ is NExpTime-hard. In [6] it is shown that the two variable first-order fragment with equality is NExpTime-complete. It follows therefore that the computational complexity of $\mathcal{ALBO}$-satisfiability is NExpTime-complete. This follows also from a (slight extension of a) result in [14].

In this paper we present a tableau approach which decides the description logic $\mathcal{ALBO}$. The tableau calculi we define for $\mathcal{ALBO}$ are ground semantic tableau calculi which work on ground labelled expressions. In contrast to the tableau calculi for description logics with role operators presented in [3, 11–13] (including $\mathcal{ALC}(\sqcup, \sqcap, ^{-1})$, $\mathcal{ALC}(\sqcup, \sqcap, ^{-1}, \restriction)$, and Peirce logic, or the equivalent modal versions, where $\restriction$ denotes the domain restriction operator), our tableau calculi operate only on ground labelled concept expressions. As a consequence the calculi can be implemented as extensions of existing tableau-based description logic systems which can handle singleton concepts.

In order to limit the number of objects in the tableau we need a mechanism for detecting periodicity in the underlying interpretations (models). *Standard loop checking* mechanisms are based on comparing sets of (labelled or unlabelled) concept expressions such as subset blocking or equality blocking. Instead of using the standard loop checking mechanisms our procedure uses a new inference rule, the *unrestricted blocking* rule, and equality reasoning. Our approach has the following advantages over standard loop checking.

- It is conceptually simple and easy to implement.
- It is universal and does not depend on the notion of a type.

[1] It is not difficult to extend our method and results to include full equality handling including reflexive roles, identity and diversity roles, and the test operator.

– It is versatile and enables more controlled model construction in a tableau procedure. For instance, it can be used to construct small models for a satisfiable concept, e.g. domain minimal models.
– It generalises to other logics, including full first-order logic.
– It can be simulated in first-order logic provers.

The unrestricted blocking rule corresponds to an unrestricted version of the first-order blocking rule invented by [1], simply called the *blocking rule*. The blocking rule is constrained to objects l and l' such that the object l' is a successor of the object l. I.e. in the common branch of l and l' the object l' is obtained from l as a result of a sequence of applications of the existential restriction rule. In this form the rule can be used to simulate standard blocking mechanisms.

The structure of the paper is as follows. The syntax and semantics of $\mathcal{ALBO}$ is defined in Section 2. In Section 3 we prove that $\mathcal{ALBO}$ has the finite model property. The constructions used in the proof are also used in Section 4, where we define a tableau calculus for $\mathcal{ALBO}$ and prove that it is sound and complete without the unrestricted blocking rule. Section 5 introduces the (unrestricted) blocking mechanism and proves soundness, completeness and termination of the extended tableau calculus. This allows us to define general decision procedures for $\mathcal{ALBO}$ and its sublogics which is discussed in Section 6. We conclude with Section 7. Due to lack of space some proofs are omitted or only sketched.

2 Syntax and semantics of $\mathcal{ALBO}$

The syntax of $\mathcal{ALBO}$ is defined over the signature $\sigma = (\mathsf{O}, \mathsf{C}, \mathsf{R})$ of three disjoint alphabets: $\mathsf{O} = \{\ell_0, \ell_1, \ldots\}$ the alphabet of object symbols, $\mathsf{C} = \{p_0, p_1, \ldots\}$ the alphabet of concept symbols, and $\mathsf{R} = \{r_0, r_1, \ldots\}$ the alphabet of role symbols. The logical connectives are: $\neg$, $\sqcup$, $\exists$, $^{-1}$ (role *inverse*), $\restriction$ (*domain restriction*), $\downharpoonright$ (*range restriction*). *Concept expressions* (or *concepts*) and *role expressions* (or *roles*) are defined as follows:

$$C \stackrel{\text{def}}{=} p \mid \{\ell\} \mid \neg C \mid C \sqcup D \mid \exists R.C,$$
$$R \stackrel{\text{def}}{=} r \mid R^{-1} \mid \neg R \mid R \sqcup S \mid R \restriction C \mid R \downharpoonright C.$$

p ranges over the set C, ℓ ranges over O, and r ranges over R. The $\sqcap$ connective on concepts and roles is defined as usual in terms of $\neg$ and $\sqcup$, and the top and bottom concepts are defined by $\top \stackrel{\text{def}}{=} p \sqcup \neg p$ and $\bot \stackrel{\text{def}}{=} p \sqcap \neg p$, respectively, for some concept name p. The universal restriction operator $\forall$ is a dual to the existential restriction operator $\exists$, specified by $\forall R.C \stackrel{\text{def}}{=} \neg \exists R. \neg C$.

Next, we define the semantics of $\mathcal{ALBO}$. A *model* (or an *interpretation*) $\mathcal{I}$ of $\mathcal{ALBO}$ is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, p_0^{\mathcal{I}}, \ldots, \ell_0^{\mathcal{I}}, \ldots, r_0^{\mathcal{I}}, \ldots)$, where $\Delta^{\mathcal{I}}$ is a non-empty set, $p_i^{\mathcal{I}}$ is a subset of $\Delta^{\mathcal{I}}$, $\ell_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, and $r_0^{\mathcal{I}}$ is a binary relation over $\Delta^{\mathcal{I}}$. The semantics of concepts and roles in the model $\mathcal{I}$, i.e. $C^{\mathcal{I}}$ and $R^{\mathcal{I}}$, is specified in Figure 1. A *TBox* (respectively *RBox*), is a (finite) set of inclusion statements $C \sqsubseteq D$ (respectively $R \sqsubseteq S$) which are interpreted in any model $\mathcal{I}$ as subset

$$
\begin{array}{rlrl}
\{\ell\}^{\mathcal{I}} & \stackrel{\text{def}}{=} \{\ell^{\mathcal{I}}\}, & (R^{-1})^{\mathcal{I}} & \stackrel{\text{def}}{=} (R^{\mathcal{I}})^{-1} = \{(x,y) \mid (y,x) \in R^{\mathcal{I}}\}, \\
(\neg C)^{\mathcal{I}} & \stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, & (\neg R)^{\mathcal{I}} & \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} & \stackrel{\text{def}}{=} C^{\mathcal{I}} \cup D^{\mathcal{I}}, & (R \sqcup S)^{\mathcal{I}} & \stackrel{\text{def}}{=} R^{\mathcal{I}} \cup S^{\mathcal{I}}, \\
\multicolumn{4}{r}{(R \restriction C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{(x,y) \mid x \in C^{\mathcal{I}} \text{ and } (x,y) \in R^{\mathcal{I}}\},} \\
\multicolumn{4}{r}{(R \downharpoonright C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{(x,y) \mid y \in C^{\mathcal{I}} \text{ and } (x,y) \in R^{\mathcal{I}}\},} \\
\multicolumn{4}{r}{(\exists R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \mid \exists y \in C^{\mathcal{I}}\ (x,y) \in R^{\mathcal{I}}\}.}
\end{array}
$$

Fig. 1. Definition of $\cdot^{\mathcal{I}}$

relationships, namely $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (respectively $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$). An *ABox* is a (finite) set of statements of the form $\ell : C$ or $(\ell, \ell') : R$, called concept assertions or role assertions. A *knowledge base* is a tuple (T, R, A) of a TBox T, an RBox R, and an ABox A.

The *top role* ∇ and the *empty role* $\triangle$ are definable in $\mathcal{ALBO}$ as $r \sqcup \neg r$ and $r \sqcap \neg r$, respectively, for some role symbol r. As a consequence any concept assertion $\ell : C$ can be expressed as a concept expression as follows: $\ell : C \stackrel{\text{def}}{=} \exists \nabla.(\{\ell\} \sqcap C)$. It is clear that $(\ell : C)^{\mathcal{I}} = \Delta^{\mathcal{I}}$ iff $\ell^{\mathcal{I}} \in C^{\mathcal{I}}$ in every model $\mathcal{I}$. In $\mathcal{ALBO}$ a role assertion $(\ell, \ell') : R$ can also be expressed as a concept assertion and concept expression, namely $(\ell, \ell') : R \stackrel{\text{def}}{=} \ell : \exists R.\{\ell'\}$. Moreover, concept and role inclusion axioms and are definable as concept expressions, too. We let $C \sqsubseteq D \stackrel{\text{def}}{=} \forall \nabla.(\neg C \sqcup D)$ and $R \sqsubseteq S \stackrel{\text{def}}{=} \forall \nabla. \exists \neg(\neg R \sqcup S).\bot$, respectively. Thus, Boolean combinations of inclusion and assertion statements of concepts and roles are also expressible in $\mathcal{ALBO}$ as the corresponding Boolean combinations of the concepts which represent these statements. As usual, concept satisfiability in $\mathcal{ALBO}$ with respect to any knowledge base can be reduced to concept satisfiability with respect to a knowledge base where all TBox, RBox, and ABox are empty. Without loss of generality we therefore focus on the problem of concept satisfiability in $\mathcal{ALBO}$.

3 Finite model property

Let $\prec$ be the smallest transitive ordering on the set of all $\mathcal{ALBO}$ expressions (concepts and roles) satisfying:

(s1) $C \prec \neg C$,
(s2) $C \prec C \sqcup D$,
(s3) $D \prec C \sqcup D$,
(s4) $\neg C \prec \neg(C \sqcup D)$,
(s5) $\neg D \prec \neg(C \sqcup D)$,
(s6) $C \prec \exists R.C$,
(s7) $R \prec \exists R.C$,
(s8) $\neg C \prec \neg \exists R.C$,
(s9) $R \prec \neg \exists R.C$,
(s10) $R \prec R \sqcup S$,
(s11) $S \prec R \sqcup S$,
(s12) $R \prec R^{-1}$,
(s13) $\neg C \prec R \restriction C$,
(s14) $R \prec R \restriction C$,
(s15) $\neg C \prec R \downharpoonright C$,
(s16) $R \prec R \downharpoonright C$,
(s17) $R \prec \neg R$.

It is easy to see that $\prec$ is a well-founded ordering. Let $\preceq$ be the reflexive closure of $\prec$.

Let $\mathcal{I}$ be any $\mathcal{ALBO}$ model and C be a concept. A *C-type* $\tau^C(x)$ of an element x of the model $\mathcal{I}$ is defined by $\tau^C(x) \stackrel{\text{def}}{=} \{D \mid D \preceq C \text{ and } x \in D^{\mathcal{I}}\}$. Let $\sim$ be an equivalence relation on $\Delta^{\mathcal{I}}$ such that $x \sim y$ implies $\tau^C(x) = \tau^C(y)$. Let $\|x\| \stackrel{\text{def}}{=} \{y \in \Delta^{\mathcal{I}} \mid x \sim y\}$.

Given a model $\mathcal{I}$ we define the *filtrated* model $\overline{\mathcal{I}}$ (through $\sim$), as follows. Let $\Delta^{\overline{\mathcal{I}}} \stackrel{\text{def}}{=} \{\|x\| \mid x \in \Delta^{\mathcal{I}}\}$. For every $r \in \mathsf{R}$ let $r^{\overline{\mathcal{I}}} \stackrel{\text{def}}{=} \{(\|x\|, \|y\|) \mid (x, y) \in r^{\mathcal{I}}\}$. For every $p \in \mathsf{C}$ let $p^{\overline{\mathcal{I}}} \stackrel{\text{def}}{=} \{\|x\| \mid x \in p^{\mathcal{I}}\}$ and for every $\ell \in \mathsf{O}$ let $\ell^{\overline{\mathcal{I}}} \stackrel{\text{def}}{=} \|\ell^{\mathcal{I}}\| = \{\ell^{\mathcal{I}}\}$.

The following lemma can be proved by induction on the ordering $\prec$.

Lemma 1 (Filtration Lemma).

(1) $x \in D^{\mathcal{I}}$ iff $\|x\| \in D^{\overline{\mathcal{I}}}$ for any $D \preceq C$ and $x \in \Delta^{\mathcal{I}}$,
(2) $(x, y) \in R^{\mathcal{I}}$ iff $(\|x\|, \|y\|) \in R^{\overline{\mathcal{I}}}$ for every $R \prec C$ and $x, y \in \Delta^{\mathcal{I}}$.

A corollary of this lemma is the finite model property.

Theorem 1 (Finite Model Property). *$\mathcal{ALBO}$ has the finite model property, i.e. if a concept C is satisfiable then it has a finite model.*

Proof. Let $\mathcal{I}$ be a model for a concept C and $\sim$ is an equivalence relation on $\Delta^{\mathcal{I}}$ defined by $x \sim y \stackrel{\text{def}}{\Longleftrightarrow} \tau^C(x) = \tau^C(y)$. Then, by the Filtration Lemma, the model $\overline{\mathcal{I}}$ filtrated through $\sim$ is also a model for C. Moreover the domain of $\overline{\mathcal{I}}$ is finite because the number of C-types in every model is finite.

4 Tableau calculus

Let T denote a tableau calculus and C a concept. We denote by $T(C)$ a finished tableau built using the rules of the calculus T starting with the concept C as input. I.e. we assume that all branches in the tableau are expanded and all applicable rules of T have been applied in $T(C)$. As usual we assume that all the rules of the calculus are applied *non-deterministically, to a tableau.* A branch of a tableau is *closed* if a contradiction has been derived in this branch, otherwise the branch is called *open*. The tableau $T(C)$ is *closed* if all its branches are closed and $T(C)$ is *open* otherwise. We say that T is *terminating* iff for every concept C either $T(C)$ is finite whenever $T(C)$ is closed or $T(C)$ has a finite open branch if $T(C)$ is open. T is *sound* iff C is unsatisfiable whenever $T(C)$ is closed for all concepts C. T is *complete* iff for any concept C, C is satisfiable (has a model) whenever $T(C)$ is open.

Let $T_{\mathcal{ALBO}}$ be the tableau calculus consisting of the rules listed in Table 1. Given an input concept C, preprocessing is performed which pushes the role inverse operators toward atomic concepts by exhaustively applying the following role equivalences from left to right.

$$(\neg R)^{-1} = \neg(R^{-1}), \qquad (R \sqcup S)^{-1} = R^{-1} \sqcup S^{-1},$$
$$(R \upharpoonright C)^{-1} = R^{-1} \upharpoonleft C, \qquad (R \upharpoonleft C)^{-1} = R^{-1} \upharpoonright C, \qquad (R^{-1})^{-1} = R.$$

$$(\bot)\colon \frac{\ell : C, \quad \ell : \neg C}{\bot} \qquad (\neg\neg)\colon \frac{\ell : \neg\neg C}{\ell : C}$$

$$(\neg\sqcup)\colon \frac{\ell : \neg(C \sqcup D)}{\ell : \neg C, \quad \ell : \neg D} \qquad (\sqcup)\colon \frac{\ell : (C \sqcup D)}{\ell : C \mid \ell : D}$$

$$(\mathsf{sym})\colon \frac{\ell : \{\ell'\}}{\ell' : \{\ell\}} \qquad (\mathsf{sym}\text{-})\colon \frac{\ell : \neg\{\ell'\}}{\ell' : \neg\{\ell\}} \qquad (\mathsf{mon})\colon \frac{\ell : \{\ell'\}, \quad \ell' : C}{\ell : C} \qquad (\mathsf{id})\colon \frac{\ell : C}{\ell : \{\ell\}}$$

$$(\exists)\colon \frac{\ell : \exists R.C}{\ell : \exists R.\{\ell'\}, \quad \ell' : C}(\ell' \text{ is new}) \qquad (\neg\exists)\colon \frac{\ell : \neg\exists R.C, \quad \ell : \exists R.\{\ell'\}}{\ell' : \neg C}$$

$$(\mathsf{bridge})\colon \frac{\ell : \exists R.\{\ell'\}, \quad \ell' : \{\ell''\}}{\ell : \exists R.\{\ell''\}}$$

$$(\exists\sqcup)\colon \frac{\ell : \exists(R \sqcup S).\{\ell'\}}{\ell : \exists R.\{\ell'\} \mid \ell : \exists S.\{\ell'\}} \qquad (\neg\exists\sqcup)\colon \frac{\ell : \neg\exists(R \sqcup S).C}{\ell : \neg\exists R.C, \quad \ell : \neg\exists S.C}$$

$$(\exists^{-1})\colon \frac{\ell : \exists R^{-1}.\{\ell'\}}{\ell' : \exists R.\{\ell\}} \qquad (\neg\exists^{-1})\colon \frac{\ell : \neg\exists R^{-1}.C, \quad \ell' : \exists R.\{\ell\}}{\ell' : \neg C}$$

$$(\exists\restriction)\colon \frac{\ell : \exists(R \restriction C).\{\ell'\}}{\ell : C, \quad \ell : \exists R.\{\ell'\}} \qquad (\neg\exists\restriction)\colon \frac{\ell : \neg\exists(R \restriction C).D}{\ell : \neg C \mid \ell : \neg\exists R.D}$$

$$(\exists\downharpoonright)\colon \frac{\ell : \exists(R \downharpoonright C).\{\ell'\}}{\ell' : C, \quad \ell : \exists R.\{\ell'\}} \qquad (\neg\exists\downharpoonright)\colon \frac{\ell : \neg\exists(R \downharpoonright C).D}{\ell : \neg\exists R.\neg(\neg C \sqcup \neg D)}$$

$$(\exists\neg)\colon \frac{\ell : \exists\neg R.\{\ell'\}}{\ell : \neg\exists R.\{\ell'\}} \qquad (\neg\exists\neg)\colon \frac{\ell : \neg\exists\neg R.C, \quad \ell' : D}{\ell : \exists R.\{\ell'\} \mid \ell : \exists\neg R.\{\ell'\}}$$

Table 1. Tableau calculus $T_{\mathcal{ALBO}}$ for $\mathcal{ALBO}$.

Next, the (preprocessed) input concept C is tagged with a fresh object name ℓ which does not occur in C. Then we build a complete tableau $T_{\mathcal{ALBO}}(C)$ as usual by applying the rules of $T_{\mathcal{ALBO}}$ to the concept assertion $\ell : C$. It is however important to note that $\ell : C$ and all labelled expressions and assertions really denote concept expressions.

Because every rule preserves the satisfiability of concept assertions, it is easy to see that the calculus $T_{\mathcal{ALBO}}$ is sound for $\mathcal{ALBO}$.

We turn to proving completeness of the calculus. Suppose that a tableau $T_{\mathcal{ALBO}}(C)$ for the given concept C is open, i.e. it contains an open branch $\mathcal{B}$. We construct a model $\mathcal{I}$ for the satisfiability of C as follows. By definition, let $\ell \sim \ell' \overset{\text{def}}{\Longleftrightarrow} \ell : \{\ell'\} \in \mathcal{B}$. It is clear that the rules (sym), (mon), and (id) ensure that $\sim$ is an equivalence relation on objects. The equivalence class $\|\ell\|$ of a representative ℓ is defined as usual by: $\|\ell\| \overset{\text{def}}{=} \{\ell' \mid \ell \sim \ell'\}$. We set

$$\Delta^{\mathcal{I}} \overset{\text{def}}{=} \{\|\ell\| \mid \ell : \{\ell\} \in \mathcal{B}\}, \qquad r^{\mathcal{I}} \overset{\text{def}}{=} \{(\|\ell\|, \|\ell'\|) \mid \ell : \exists r.\{\ell'\} \in \mathcal{B}\},$$

$$p^{\mathcal{I}} \overset{\text{def}}{=} \{\|\ell\| \mid \ell : p \in \mathcal{B}\}, \qquad \ell^{\mathcal{I}} \overset{\text{def}}{=} \begin{cases} \|\ell\|, & \text{if } \ell : \{\ell\} \in \mathcal{B}, \\ \|\ell'\| & \text{for some } \|\ell'\| \in \Delta^{\mathcal{I}}, \text{ otherwise.} \end{cases}$$

It is easy to show that, using the rules (sym), (mon), and (id), the definition of $\mathcal{I}$ does not depend on representatives of the equivalence classes.

Lemma 2. *(1) If $\ell : D \in \mathcal{B}$ then $\|\ell\| \in D^{\mathcal{I}}$ for any concept D.*
(2) For every role R and every concept D
(2a) $\ell : \exists R.\{\ell'\} \in \mathcal{B}$ implies $(\|\ell\|, \|\ell'\|) \in R^{\mathcal{I}}$,
(2b) if $(\|\ell\|, \|\ell'\|) \in R^{\mathcal{I}}$ and $\ell : \neg\exists R.D \in \mathcal{B}$ then $\ell' : \neg D \in \mathcal{B}$.

Proof. We prove both properties simultaneously by induction on the ordering $\prec$. The induction hypothesis is: for an arbitrary $\mathcal{ALBO}$ expression E for every expression F such that $F \prec E$ if F is a concept then property (1) holds with $D = F$. Otherwise (i.e. if F is a role), property (2) holds with $R = F$. To prove property (1) we consider the following cases.

$D = p$, $D = \{\ell'\}$. These cases follow from the definitions of $p^{\mathcal{I}}$ and $\ell^{\mathcal{I}}$.

$D = \neg D_0$. If $\ell : \neg D_0 \in \mathcal{B}$ then $\ell : D_0 \notin \mathcal{B}$ because otherwise $\mathcal{B}$ would have been closed by the $(\bot)$ rule. We have the following subcases.

$D_0 = p$. We have $\ell : p \in \mathcal{B} \iff \|\ell\| \in p^{\mathcal{I}}$ by the definition of $p^{\mathcal{I}}$.

$D_0 = \{\ell'\}$. As the rules (sym-) and (id) have been applied in $\mathcal{B}$ it is clear that $\ell' : \{\ell'\}$ is in $\mathcal{B}$ and $\ell'^{\mathcal{I}} = \|\ell'\|$. Similarly, $\ell^{\mathcal{I}} = \|\ell\|$. Furthermore, because $\ell : \{\ell'\} \notin \mathcal{B}$ we have $\ell \not\sim \ell'$, i.e. $\ell \notin \|\ell'\| = \ell'^{\mathcal{I}}$. That is, $\|\ell\| \notin \{\|\ell'\|\} = \{\ell'\}^{\mathcal{I}}$.

$D_0 = \neg D_1$, $D_0 = D_1 \sqcup D_2$. The proofs of these cases are easy.

$D_0 = \exists R.D_1$. Let $\|\ell'\|$ be an arbitrary element of $\Delta^{\mathcal{I}}$ such that $(\|\ell\|, \|\ell'\|) \in R^{\mathcal{I}}$ (trivially, if there is no such element then there is nothing to prove). By the induction hypothesis the property (2b) holds for $R \prec D_0$. Thus, $\ell' : \neg D_1 \in \mathcal{B}$. The induction hypothesis for the property (1) gives us $\|\ell'\| \notin D_1^{\mathcal{I}}$. Finally, we obtain $\|\ell\| \in (\neg\exists R.D_1)^{\mathcal{I}}$ because ℓ' was chosen arbitrarily.

$D = D_0 \sqcup D_1$. The proof of this case is easy.

$D = \exists R.D_0$. If $\ell : \exists R.D_0 \in \mathcal{B}$ then $\ell' : D_0 \in \mathcal{B}$ and $\ell : \exists R.\{\ell'\} \in \mathcal{B}$ for some object ℓ' by the $(\exists)$ rule. By the induction hypothesis the properties (1) and (2a) hold for $D_0 \prec D$ and $R \prec D$ respectively. Hence, $\|\ell'\| \in D_0^{\mathcal{I}}$ and $(\|\ell\|, \|\ell'\|) \in R^{\mathcal{I}}$. That is, $\|\ell\| \in (\exists R.D_0)^{\mathcal{I}}$.

To prove property (2) we consider all cases corresponding to the possible forms of a role R.

$R = r$. This case easily follows from the definition of $r^{\mathcal{I}}$.

$R = S^{-1}$. For the property (2a) let $\ell : \exists S^{-1}.\{\ell'\} \in \mathcal{B}$. Then $\ell' : \exists S.\{\ell\} \in \mathcal{B}$ by the rule $(\exists^{-1})$. By the induction hypothesis for $S \prec R$ we have $(\|\ell'\|, \|\ell\|) \in S^{\mathcal{I}}$. Consequently, $(\|\ell\|, \|\ell'\|) \in (S^{-1})^{\mathcal{I}}$. For (2b) suppose that $(\|\ell\|, \|\ell'\|) \in (S^{-1})^{\mathcal{I}}$ and $\ell : \neg\exists S^{-1}.D \in \mathcal{B}$. As all the occurrences of the inverse operator have been pushed through other role connectives and double occurrences of $^{-1}$ have been removed[2] we can assume that $S = r$ for some role name r. Hence, $(\|\ell'\|, \|\ell\|) \in r^{\mathcal{I}}$ and, consequently, $\ell' : \exists r.\{\ell\} \in \mathcal{B}$ by the definition of $r^{\mathcal{I}}$. Finally, by the $(\neg\exists^{-1})$ rule, $\ell' : \neg D$ is in the branch $\mathcal{B}$.

$R = S_0 \sqcup S_1$, $R = S \downharpoonleft D$, $R = S \upharpoonleft D$. The proofs of these cases are easy.

[2] Removal of the double occurrences of the inverse operator in front of atoms is not essential for the proof but it simplifies the proof a bit.

$R = \neg S$. For (2a) suppose $\ell : \exists\neg S.\{\ell'\} \in \mathcal{B}$. Then $\ell : \neg\exists S.\{\ell'\} \in \mathcal{B}$ is obtained with the $(\exists\neg)$ rule. If $(\|\ell\|, \|\ell'\|) \notin (\neg S)^{\mathcal{I}}$ then $(\|\ell\|, \|\ell'\|) \in S^{\mathcal{I}}$ and by property (2b) which holds by the induction hypothesis for $S \prec R$ we have that $\ell' : \neg\{\ell'\}$ is in $\mathcal{B}$. This concept together with $\ell' : \{\ell'\}$ implies the branch is closed. We reach a contradiction, so $(\|\ell\|, \|\ell'\|) \in (\neg S)^{\mathcal{I}}$.
For property (2b) suppose that $(\|\ell\|, \|\ell'\|) \in (\neg S)^{\mathcal{I}}$ and $\ell : \neg\exists\neg S.D$ are in the branch $\mathcal{B}$. Then we have $(\|\ell\|, \|\ell'\|) \notin S^{\mathcal{I}}$ and, hence, by the contra-positive of property (2a) for $S \prec R$, $\ell : \exists S.\{\ell'\}$ is not in $\mathcal{B}$. Applying the $(\neg\exists\neg)$ rule to $\ell : \neg\exists\neg S.D$ we get $\ell : \exists\neg S.\{\ell'\} \in \mathcal{B}$. Therefore, by the $(\neg\exists)$ rule, $\ell' : \neg D$ is in the branch too.

A consequence of this lemma is completeness of the tableau calculus. Hence, we can state:

Theorem 2. *$T_{\mathcal{ALBO}}$ is a sound and complete tableau calculus for $\mathcal{ALBO}$.*

5 Blocking

The calculus $T_{\mathcal{ALBO}}$ is non-terminating for $\mathcal{ALBO}$. There are satisfiable concepts which result in an infinite $T_{\mathcal{ALBO}}$-tableau where all open branches are infinite. All the rules respect the well-founded ordering $\prec$ of expressions under labels, i.e. in every rule the main symbol of the concept above the line is strictly greater w.r.t. $\prec$ than the main symbol(s) of the expression(s) below the line of the rule. Furthermore, it is easy to see that only applications of the $(\exists)$ rule generate new symbols in the branch. Thus, the reason that a branch can be infinite is the unlimited application of the $(\exists)$ rule. As a consequence, the following lemma holds, where $\#^{\exists}(\mathcal{B})$ denotes the number of applications of the $(\exists)$ rule in a branch $\mathcal{B}$.

Lemma 3. *If $\#^{\exists}(\mathcal{B})$ is finite then $\mathcal{B}$ is finite.*

In order to avoid infinite derivations we restrict the application of the $(\exists)$ rule by the following *blocking* mechanism.

Let $<$ be an ordering on objects in the branch which is a linear extension of the order in which the objects are introduced during a derivation. I.e. let $\ell < \ell'$ whenever the first appearance of object ℓ' in the branch is strictly later than the first appearance of the object ℓ. We add the following rule, called the *unrestricted blocking* rule, to the calculus.

$$(\mathsf{ub}): \frac{\ell : C, \quad \ell' : D}{\ell : \{\ell'\} \mid \ell : \neg\{\ell'\}}$$

Moreover, we require the following conditions to hold.

(c1) Any rule is applied at most once to the same set of premises.
(c2) The $(\exists)$ rule is applied only to expressions of the form $\ell : \exists R.C$ when C is not a singleton, i.e. $C \neq \{\ell''\}$ for some object ℓ''.

(c3) If $\ell : \{\ell'\}$ appears in a branch and $\ell < \ell'$ then all further applications of the $(\exists)$ rule to expressions of the form $\ell' : \exists R.C$ are not performed within the branch.
(c4) In every open branch there is some node from which point onwards before any application of the $(\exists)$ rule all possible applications of the (ub) rule must have been performed.

We use the notation $T_{\mathcal{ALBO}}$ + (ub) for the extension of $T_{\mathcal{ALBO}}$ with this rule and this blocking mechanism.

Theorem 3. *$T_{\mathcal{ALBO}}$+(ub) is a sound and complete tableau calculus for $\mathcal{ALBO}$.*

Proof. The blocking requirements (c1)–(c4) are sound in the sense that they cannot cause an open branch to become closed. The (ub) rule is sound in the usual sense. Thus, we can safely add the blocking requirements and the blocking rule to a sound and complete tableau without endangering soundness or completeness. Hence, $T_{\mathcal{ALBO}}$ + (ub) is sound and complete.

Let $\mathcal{B}$ be the *leftmost* open branch with respect to the rule (ub) in the $T_{\mathcal{ALBO}}$ + (ub) tableau for a given concept C. Assume that $\mathcal{I}$ is a model constructed from $\mathcal{B}$ as in the completeness section above.

Lemma 4. *If $\tau^C(\|\ell\|) = \tau^C(\|\ell'\|)$ in $\mathcal{I}$ then $\ell : \{\ell'\} \in \mathcal{B}$.*

Proof. Suppose that $\tau^C(\|\ell\|) = \tau^C(\|\ell'\|)$. Therefore, $\ell : \{\ell'\}$ is consistent with C. By (c4) the rule (ub) has been applied to the objects ℓ and ℓ' in $\mathcal{B}$. As $\ell : \{\ell'\}$ is consistent with C, the left branch (containing $\ell : \{\ell'\}$) of this application of (ub) is open and, by the choice of the branch $\mathcal{B}$, coincides with $\mathcal{B}$.

Corollary 1. *The model $\overline{\mathcal{I}}$ obtained from $\mathcal{I}$ by filtration with respect to C is isomorphic to $\mathcal{I}$. In particular, $\Delta^{\mathcal{I}}$ is finite.*

For every $\|\ell\| \in \Delta^{\mathcal{I}}$, let $\#^{\exists}(\|\ell\|)$ denote the number of applications of the $(\exists)$ rule to concepts of the form $\ell' : \exists R.D$ with $\ell' \in \|\ell\|$.

Lemma 5. *$\#^{\exists}(\|\ell\|)$ is finite for every $\|\ell\| \in \Delta^{\mathcal{I}}$.*

Proof. Suppose not, i.e. $\#^{\exists}(\|\ell\|)$ is infinite. The number of concepts of the shape $\exists R.D$ under labels in the branch is finitely bounded. By requirements (c1) and (c2) there is a sequence of objects $\ell_0, \ell_1, \ldots$ such that every $\ell_i \in \|\ell\|$ and the $(\exists)$ rule has been applied to concepts $\ell_0 : \exists R.D, \ell_1 : \exists R.D, \ldots$ for some $\exists R.D \prec C$. However, such a situation is impossible because of requirements (c4) and (c3). Indeed, without loss of generality we can assume that $\ell < \ell_0 < \ell_1 < \cdots$. Then, by requirement (c4), starting from some node of $\mathcal{B}$, as soon as ℓ_i appears in $\mathcal{B}$, it is detected that $\ell_i \in \|\ell\|$ before any next application of the rule $(\exists)$ and, hence, ℓ_i is immediately blocked for any application of the rule $(\exists)$, by requirement (c3).

Lemma 6. *$\#^{\exists}(\mathcal{B})$ is finite.*

Proof. Clearly $\#^{\exists}(\mathcal{B}) \leq \max\{\#^{\exists}(\|\ell\|) \mid \|\ell\| \in \Delta^{\mathcal{I}}\} \times \mathsf{Card}(\Delta^{\mathcal{I}})$. The rest follows from Corollary 1 and Lemma 5.

Corollary 2. *If the leftmost branch with respect to the rule* (ub) *in a* $T_{\mathcal{ALBO}} +$ (ub) *tableau is open then the branch is finite.*

Theorem 4 (Termination). $T_{\mathcal{ALBO}} +$ (ub) *is a terminating tableau calculus for* $\mathcal{ALBO}$.

Proof. Termination of $T_{\mathcal{ALBO}} +$ (ub) follows from Corollary 2. Indeed, every closed branch of a $T_{\mathcal{ALBO}} +$ (ub)-tableau is trivially finite and by Corollary 2 the length of the leftmost open branch with respect to the rule (ub) is finite, too.

Notice that condition (c4) is essential for ensuring termination of a $T_{\mathcal{ALBO}} +$ (ub) derivation. Indeed, it easy to see that in absence of (c4) the $T_{\mathcal{ALBO}} +$ (ub) tableau for the concept $\neg(\exists(s \sqcup \neg s).\neg\exists r.p \sqcup \exists(s \sqcup \neg s).\neg\exists r.\neg p)$ does not terminate because new objects are generated more often than their equality check via the rule (ub) is performed in the tableau.

6 Decision procedures

When turning the presented calculus $T_{\mathcal{ALBO}} +$ (ub) into a deterministic decision procedure it is crucial that this is done in a *fair* way. A procedure is fair if, when if an inference is possible forever then it is performed eventually. In other words a deterministic tableau algorithm based on $T_{\mathcal{ALBO}} +$ (ub) may not defer the use of an applicable rule indefinitely. Note that understand fairness in a 'global' sense. That is, a tableau algorithm has to be fair not only to expressions in a particular branch but to expressions in all branches of a tableau. In another words, the algorithm is fair if it is fairly chooses a branch and expression(s) in it to apply a rule.

Theorem 5. *Any fair tableau procedure based on* $T_{\mathcal{ALBO}} +$ (ub) *is a decision procedure for* $\mathcal{ALBO}$ *and all its sublogics.*

Note that we do not assume that the branches are expanded in a depth-first left-to-right order. However it also follows from our results that:

Theorem 6. *Any fair tableau procedure based on* $T_{\mathcal{ALBO}} +$ (ub) *which uses a depth-first and left-to-right strategy, with respect to branch selection of the* (ub) *rule, is a decision procedure for* $\mathcal{ALBO}$ *and all its sublogics.*

To illustrate the importance of fairness we give an example. The concept

$$C \stackrel{\text{def}}{=} \neg\left(\exists(s \sqcup \neg s).\neg\exists r.p \sqcup \neg\exists t.\neg\exists r.p\right)$$

is not satisfiable. Figure 2 gives a depth-first left-to-right derivation which is unfair and does not terminate. Each line in the derivation is numbered on the left. The rule applied and the number of the premise(s) to which it was applied to

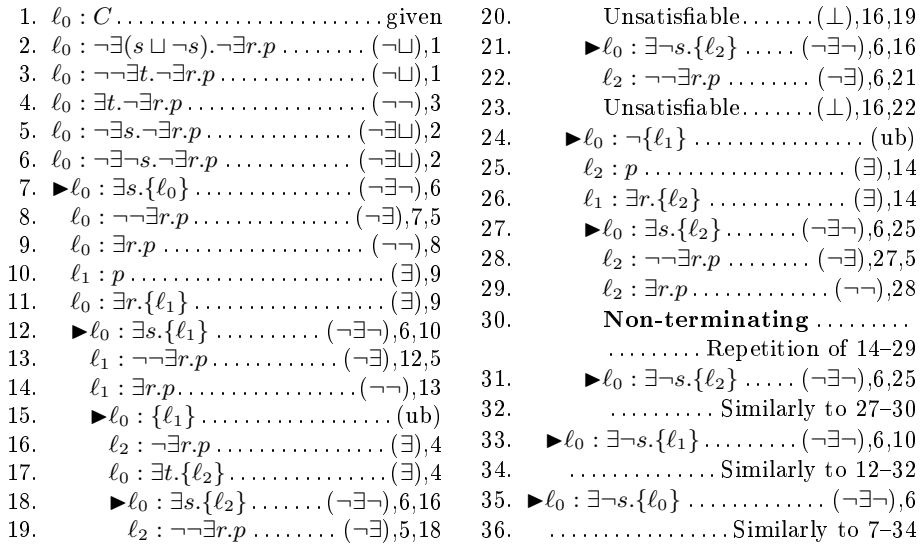

Fig. 2. An infinite, unfair derivation

produce the labelled concept expression (assertion) in each line is specified on the right. The black triangles denote branching points in the derivation. A branch expansion after a branching point is indicated by appropriate indentation. We observe that the derivation is infinite because the application of the $(\exists)$ rule to $\ell_0 : \exists t.\neg\exists r.p$ is deferred forever and, consequently, a contradiction is not found. The example illustrates the importance of fairness to completeness.

Without giving further details we observe that the calculi are compatible with standard optimisations such as backjumping, simplification, different strategies for branch selection and rule selection, etc provided that the fairness condition is not violated.

We have implemented the unrestricted blocking rule as a plug-in to the MetTeL tableau prover [9], and tested it on various description logics.

7 Conclusion

We have presented a new, general tableau approach for deciding description logics with complex role operators, including especially 'non-safe' occurrences of role negation. The tableau decision procedures found in the description logic literature, and implemented in existing tableau-based description logic systems, can handle a large class of description logics but cannot currently handle description logics with full role negation such as $\mathcal{ALB}$ or $\mathcal{ALBO}$. An important novelty of our approach is the use of a blocking mechanism based on the use of inference rules rather than standard loop checking mechanisms which are based on tests performed on sets of expressions or assertions which may need to be

tailored toward specific logics. Our techniques are versatile and are not limited to $\mathcal{ALBO}$ or its sublogics, but carry over to all description logics and also other logics including first-order logic. We are optimistic that the ideas of this paper can be taken a lot further.

References

1. P. Baumgartner and R. A. Schmidt. Blocking and other enhancements for bottom-up model generation methods. In *Proc. IJCAR 2006*, vol. 4130 of *LNAI*, pp. 125–139. Springer, 2006.
2. H. De Nivelle and I. Pratt-Hartmann. A resolution-based decision procedure for the two-variable fragment with equality. In *Proc. IJCAR 2001*, vol. 2083 of *LNAI*, pp. 211–225. Springer, 2001.
3. H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8(3):265–292, 2000.
4. G. Gargov and S. Passy. A note on Boolean modal logic. In *Mathematical Logic: Proc. 1988 Heyting Summerschool*, pp. 299–309. Plenum Press, 1990.
5. G. Gargov, S. Passy, and T. Tinchev. Modal environment for Boolean speculations. In *Mathematical Logic and its Applications: Proc. 1986 Gödel Conf.*, pp. 253–263. Plenum Press, 1987.
6. E. Grädel, P. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *Bull. Symbolic Logic*, 3:53–69, 1997.
7. U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. In *Automated Deduction in Classical and Non-Classical Logics*, vol. 1761 of *LNAI*, pp. 191–205. Springer, 2000.
8. U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *J. Relational Methods Comput. Sci.*, 1:251–276, 2004.
9. U. Hustadt, D. Tishkovsky, F. Wolter, and M. Zakharyaschev. Automated reasoning about metric and topology (System description). In *Proc. JELIA'06*, vol. 4160 of *LNAI*, pp. 490–493. Springer, 2006.
10. C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logics. In *Advances in Modal Logics, Vol. 3*. CSLI Publications, 2002.
11. R. A. Schmidt. Developing modal tableaux and resolution methods via first-order resolution. In *Advances in Modal Logic, Vol. 6*, pp. 1–26. College Publications, 2006.
12. R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics. In *Volume in memoriam of Harald Ganzinger*, LNCS. Springer, 2006. To appear.
13. R. A. Schmidt, E. Orlowska, and U. Hustadt. Two proof systems for Peirce algebras. In *Proc. RelMiCS 7*, vol. 3051 of *LNCS*, pp. 238–251. Springer, 2004.
14. S. Tobies. *Decidability Results and Practical Algorithms for Logics in Knowledge Representation*. Phd dissertation, RWTH Aachen, Germany, 2001.