Fachbereich Informatik der Universität Hamburg

Vogt-Kölln-Str. 30 ◊ D-22527 Hamburg / Germany

University of Hamburg - Computer Science Department

Mitteilung Nr. 301/01 • Memo No. 301/01

Decidable and Undecidable Extensions of $\mathcal{ALC}$ with Composition-Based Role Inclusion Axioms

Michael Wessel

Arbeitsbereich KOGS

FBI-HH-M-301/01

December 2000

Decidable and Undecidable Extensions of $\mathcal{ALC}$ with Composition-Based Role Inclusion Axioms

Michael Wessel
University of Hamburg, Computer Science Department,
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

Abstract

This paper continues our investigation on the extension of the standard description logic $\mathcal{ALC}$ with role axioms of the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$. We consider the concept satisfiability problem of $\mathcal{ALC}$ w.r.t. a set of role axioms of the proposed form. A set of these role axioms is called a role box. The original motivation for this kind of role axioms comes from foreseen applications in the field of qualitative spatial reasoning with description logics. In this paper, we define the logics $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, $\mathcal{ALC}_{\mathcal{RA}}$, and $\mathcal{ALC}_{\mathcal{RASG}}$. Basically, both $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ and $\mathcal{ALC}_{\mathcal{RA}}$ allow arbitrary role boxes containing axioms of the general form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$. In contrast to $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, $\mathcal{ALC}_{\mathcal{RA}}$ requires additionally that *all* roles have to be interpreted as disjoint. This requirement is also originally motivated by qualitative spatial reasoning applications with $\mathcal{ALC}_{\mathcal{RA}}$. Recently it turned out that $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ is undecidable. In fact, already $\mathcal{ALU}_{\mathcal{RA}^\ominus}$ with role boxes containing axioms of the form $R \circ S \sqsubseteq T$ is undecidable. A very similar result has also been obtained independently in a branch of normal multimodal logics, called *grammar logics*. Since role disjointness is a very severe restriction it is currently still unknown whether $\mathcal{ALC}_{\mathcal{RA}}$ might be decidable or not. In this paper we go back one step before $\mathcal{ALC}_{\mathcal{RA}}$ and discuss a common fragment of $\mathcal{ALC}_{\mathcal{RA}}$ and $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, called $\mathcal{ALC}_{\mathcal{RASG}}$. Like in $\mathcal{ALC}_{\mathcal{RA}}$, $\mathcal{ALC}_{\mathcal{RASG}}$ requires role disjointness, but the set of *admissible* role boxes is further pruned. It turned out that *associativity of role boxes* is an important requirement – exploiting associativity we were able to show the decidability and EXPTIME-completeness of $\mathcal{ALC}_{\mathcal{RASG}}$. Surprisingly, satisfiability of $\mathcal{ALC}_{\mathcal{RASG}}$-concepts w.r.t. admissible role boxes can be reduced to concept satisfiability w.r.t. general TBoxes in $\mathcal{ALC}$. We also show that the further augmentation of $\mathcal{ALC}_{\mathcal{RASG}}$ with unqualified number restrictions, yielding the language $\mathcal{ALCN}_{\mathcal{RASG}}$, is again undecidable.

1 Introduction and Motivation

Since the famous "Subsumption in KL-ONE is undecidable"-result by Schmidt-Schauß ([14]) who identified the so-called *role value maps* as a primary source of undecidability in the seminal description-logic system KL-ONE, role inclusion expressions containing the composition operator have not regained much attention in the DL community again (please refer to [2] for a more gentle and self-contained introduction of description logics). A *role value map* is a concept-expression of the form

$$R_1 \circ R_2 \circ \cdots \circ R_n \sqsubseteq S_1 \circ S_2 \circ \cdots \circ S_m,$$

whose extension (assuming the usual extensional, Tarski-style semantics) is given by the set

$$\{\, x \in \Delta^{\mathcal{I}} \mid \forall y : \langle x, y \rangle \in R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \Rightarrow \langle x, y \rangle \in S_1^{\mathcal{I}} \circ \ldots \circ S_m^{\mathcal{I}} \,\}.$$

Even though it was always clear that role value maps are expressive and elegant modeling constructs, only a very few restricted variants have been examined. For example, Molitor (see [9]) has shown that a certain fragment of $\mathcal{ALC}$ with role value maps of the special form $R_1 \circ R_2 \circ \cdots \circ R_n \sqsubseteq S_1 \circ S_2 \circ \cdots \circ S_n$ – note that the same number of roles appear on both sides of the inclusion – is decidable, by exploiting a certain "level structure" in the models. In our opinion, there is still a gap in the description logic theory. As pointed out by Sattler in [12], these restricted variants of role value maps still need to be investigated, since they might be decidable and very useful for knowledge representation purposes.

In this paper, we continue our research on $\mathcal{ALC}$ extended with *composition-based role inclusion axioms* of the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$ (see [16],[15]) and identify a decidable fragment. A role axiom $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$ is satisfied by an interpretation $\mathcal{I}$ if and only if $S^{\mathcal{I}} \circ T^{\mathcal{I}} \subseteq R_1^{\mathcal{I}} \cup R_2^{\mathcal{I}} \cup \cdots \cup R_n^{\mathcal{I}}$ holds. The most obvious consequence is the loss of the tree model property. Even worse, the logics we have considered don't have the finite model property. On one hand, our investigation can also be seen as an attempt towards contributing to the identification of decidable variants of role value maps. Our role axioms are not role value maps per se, since, for example, role value maps might appear *negated* if the considered base-description logic offers a full negation operator. But there is still an interesting correspondence with role value maps: if we restrict ourselves to *deterministic* composition-based role inclusion axioms (a composition-based role axiom is called deterministic iff it contains no disjunctions of roles on its right hand side) we might say, laxly speaking, that non-negated (positive) role value maps are "local" deterministic composition-based role inclusion axioms, or conversely, that deterministic composition-based role-axioms are "global" (positive) role value maps.[1]

On the other hand, the proposed kind of role axioms arises naturally when trying to address qualitative spatial reasoning tasks within description logic frameworks, which was the original motivation for considering these kinds of role axioms. Due to the problems encountered in our previous work (see [16], [15]), the primary objective of this paper is to get a better map of the borderline area between decidable and undecidable extensions of $\mathcal{ALC}$ with composition-based role inclusion axioms.

Certain well-known description logic constructs can be "emulated" with the help of role axioms having the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$. For example, a transitively closed role R corresponds to an axiom $R \circ R \sqsubseteq R$. This is the $\mathcal{R}^+$ "operator" of $\mathcal{ALC}_{\mathcal{R}^+}$ (see [10]). Simple role inclusion axioms of the form $R \sqsubseteq S$ (the $\mathcal{H}$ operator) can be emulated with $R \circ Id \sqsubseteq S$ in the presence of the *identity role* Id, having the fixed semantics of the identity relation. Also the concept satisfiability problem of the language $\mathcal{ALC}_{\mathcal{R}^\oplus}$, offering so-called *transitive orbits* (see [10]), can be reduced to concept satisfiability in $\mathcal{ALC}_{\mathcal{RA}}$ ($\mathcal{ALC}_{\mathcal{RA}^\ominus}$) w.r.t. a set of appropriate role axioms (see [15]). Since these are all well-known decidable description logics, it was always clear that certain "trivial" classes of role boxes could be found, such that the resulting logic would be decidable (see the discussion in [15]). However, it remained to show that there are any non-obvious classes of role boxes *not* corresponding to previously known decidable description logics for which this is also the case.

In this paper we identify the fragment $\mathcal{ALC}_{\mathcal{RASG}}$, offering *admissible* role boxes. Admissible role boxes (see below for an exact definition) are associative and contain only deterministic role axioms (no role axiom has disjunctions on its right hand side). The decidability result is not immediate, since $\mathcal{ALC}_{\mathcal{RASG}}$ does not have the finite model property. In previous work it turned out that $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ and even $\mathcal{ALU}_{\mathcal{RA}^\ominus}$ with role boxes solely containing axioms of the form $R \circ S \sqsubseteq T$ is undecidable. A very similar undecidability result has also been obtained independently in a branch of normal multimodal logics, called *grammar logics* (see below for a discussion). Since role disjointness is a very severe restriction it is currently still unknown whether $\mathcal{ALC}_{\mathcal{RA}}$ might be decidable or not. With $\mathcal{ALC}_{\mathcal{RASG}}$ we go back on step behind $\mathcal{ALC}_{\mathcal{RA}}$. Like in $\mathcal{ALC}_{\mathcal{RA}}$, $\mathcal{ALC}_{\mathcal{RASG}}$ requires role disjointness, but the set of *admissible* role boxes is further pruned, compared to $\mathcal{ALC}_{\mathcal{RA}}$. It turned out that *associativity of role boxes* is an important requirement – exploiting associativity we were able to show the decidability of $\mathcal{ALC}_{\mathcal{RASG}}$. We also show EXPTIME-completeness of $\mathcal{ALC}_{\mathcal{RASG}}$. Surprisingly,

[1] In the former case we have
$\{ x \in \Delta^{\mathcal{I}} \mid \forall y : \langle x,y \rangle \in R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \Rightarrow \langle x,y \rangle \in S_1^{\mathcal{I}} \circ \ldots \circ S_m^{\mathcal{I}} \} \subseteq \Delta^{\mathcal{I}}$, and in the latter
$\{ x \in \Delta^{\mathcal{I}} \mid \forall y : \langle x,y \rangle \in R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \Rightarrow \langle x,y \rangle \in S_1^{\mathcal{I}} \circ \ldots \circ S_m^{\mathcal{I}} \} = \Delta^{\mathcal{I}}$.

it turns out that satisfiability of $\mathcal{ALC}_{\mathcal{RASG}}$-concepts w.r.t. admissible role boxes can be reduced to concept satisfiability w.r.t. general TBoxes in $\mathcal{ALC}$. We like to mention that $\mathcal{ALC}_{\mathcal{RASG}}$, even though clearly a very restricted language, is at least as expressive as $\mathcal{ALC}_{\mathcal{R}^+}$ and $\mathcal{ALC}_{\mathcal{R}^\oplus}$ (see [10]). Therefore, the given reduction works also for $\mathcal{ALC}_{\mathcal{R}^+}$ and $\mathcal{ALC}_{\mathcal{R}^\oplus}$.

The structure of this paper is a follows: we will first define the syntax and semantics of the languages $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, $\mathcal{ALC}_{\mathcal{RA}}$, and $\mathcal{ALC}_{\mathcal{RASG}}$. Certain auxiliary definitions and propositions as well as already known undecidability results and facts from [15] are recalled. We will then briefly sketch the relationship to grammar logics. The next chapter, which contains a tableaux-based decision procedure for $\mathcal{ALC}_{\mathcal{RASG}}$ and the proof of EXPTIME-completeness, is the main contribution of this paper. We then show that a slight extension of $\mathcal{ALC}_{\mathcal{RASG}}$, namely $\mathcal{ALC}_{\mathcal{RASG}}$ extended with *unqualified number restrictions* (code letter $\mathcal{N}$) yielding the logic $\mathcal{ALCN}_{\mathcal{RASG}}$, is undecidable again. Finally, we conclude by discussing the impact of the insights gained from $\mathcal{ALC}_{\mathcal{RASG}}$ on future work.

2 Syntax and Semantics

In the following we will define the syntax and semantics of the three considered languages $\mathcal{ALC}_{\mathcal{RA}}$, $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, and $\mathcal{ALC}_{\mathcal{RASG}}$. We start with the set of well-formed concept expressions (concepts for short):

Definition 1 (Concept Expressions) Let $\mathcal{N}_\mathcal{C}$ be a set of concept names, and let $\mathcal{N}_\mathcal{R}$ be a set of role names (roles for short), such that $\mathcal{N}_\mathcal{C} \cap \mathcal{N}_\mathcal{R} = \emptyset$. The set of concept expressions (or concepts for short) is defined inductively:

1. Every concept name $C \in \mathcal{N}_\mathcal{C}$ is a concept.
2. If C and D are concepts, and $R \in \mathcal{N}_\mathcal{R}$ is a role, then the following expressions are concepts as well: $(\neg C)$, $(C \sqcap D)$, $(C \sqcup D)$, $(\exists R.C)$, and $(\forall R.C)$.

The set of concepts is the same as for the language $\mathcal{ALC}$. If a concept starts with "(", we call it a compound concept, otherwise a concept name or atomic concept. Brackets may be omitted for the sake of readability if the concept is still uniquely parsable. For example, instead of $((C \sqcap D) \sqcap E)$ we simply write $C \sqcap D \sqcap E$ etc., similarly for $\sqcup$ ($\sqcap$ and $\sqcup$ are considered as being left-associative). Generally, if C is a compound concept, the brackets of $C = (\ldots)$ may not be omitted within $\exists R.C$ and $\forall R.C$. If C is a concept name and we write $\exists R.C \sqcap D$, with this convention this means $(\exists R.C) \sqcap D$ and not $\exists R.(C \sqcap D)$. We may also omit the brackets when nesting $\exists$- and $\forall$-concepts: e.g. $\exists R.\exists R.\exists R.C \sqcap D$

means $((\exists R.(\exists R.(\exists R.C))) \sqcap D)$. If C is not a concept name we have to write $\exists R.\exists R.\exists R.(\ldots) \sqcap D$ to mean $(\exists R.(\exists R.(\exists R.(C \sqcap D))))$. We use the following abbreviations: if $R_1, \ldots, R_n$ are roles, and C is a concept, then we define $(\forall R_1 \sqcup \cdots \sqcup R_n.C) =_{def} (\forall R_1.C) \sqcap \cdots \sqcap (\forall R_n.C)$ and $\exists R_1 \sqcup \cdots \sqcup R_n.C =_{def} (\exists R_1.C) \sqcup \cdots \sqcup (\exists R_n.C)$. Additionally, for some $CN \in \mathcal{N_C}$ we define $\top =_{def} CN \sqcup \neg CN$ and $\bot =_{def} CN \sqcap \neg CN$ (therefore, $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} = \emptyset$). Before we can proceed, we need some auxiliary definitions. The set of *roles* being used within a concept C is defined:

Definition 2 (Used Roles, $\mathsf{roles}(C)$)

$$\mathsf{roles}(C) =_{def} \begin{cases} \emptyset & \text{if } C \in \mathcal{N_C} \\ \mathsf{roles}(D) & \text{if } C = (\neg D) \\ \mathsf{roles}(D) \cup \mathsf{roles}(E) & \text{if } C = (D \sqcap E) \\ & \text{or } C = (D \sqcup E) \\ \{R\} \cup \mathsf{roles}(D) & \text{if } C = (\exists R.D) \\ & \text{or } C = (\forall R.D) \end{cases}$$

For example, $\mathsf{roles}(((\forall R.(\exists S.C)) \sqcap \exists T.D)) = \{R, S, T\}$.

The set of *subconcepts* of a concept C is defined in the obvious way:

Definition 3 (Subconcepts of C, $\mathsf{sub}(C)$)

$$\mathsf{sub}(C) =_{def} \{C\} \cup \begin{cases} \emptyset & \text{if } C \in \mathcal{N_C} \\ \mathsf{sub}(D) & \text{if } C = (\neg D) \\ & \text{or } C = (\exists R.D) \\ & \text{or } C = (\forall R.D) \\ \mathsf{sub}(D) \cup \mathsf{sub}(E) & \text{if } C = (D \sqcap E) \\ & \text{or } C = (D \sqcup E) \end{cases}$$

For example, $\mathsf{sub}((C \sqcap (\forall R.D))) = \{(C \sqcap (\forall R.D)), C, (\forall R.D), D\}$. Obviously, $|\mathsf{sub}(C)| \leq |C|$, if $|C|$ denotes the length of the string expressing C.

As already noted, we are investigating the satisfiability of $\mathcal{ALC}$ concepts w.r.t. a set of role axioms of the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$. More formally, the syntax of these role axioms and of the considered role boxes containing these axioms is as follows. The following definitions and notions might appear to be complicated and artificial at a first sight, but the reasons for introducing them will become clear subsequently. As already noted, the primary objective of this paper is not to introduce a useful new description logic of utmost utility, but to *investigate* different kinds of restricted role boxes that ensure decidability. Even though some of the subsequent restrictions on role boxes are very strong and artificial we believe that some of the restrictions can be relaxed in order to define more and more useful description logics in the future, by exploiting some of the insights gained by this investigation:[2]

Definition 4 (Role Axioms, Role Box, Admissible Role Box) If $S, T, R_1, \ldots, R_n \in \mathcal{N_R}$, then the expression $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$, $n \geq 1$, is called a *role axiom*. If $ra = S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$, then $\mathsf{pre}(ra) =_{def} (S, T)$ and $\mathsf{con}(ra) =_{def} \{R_1, \ldots, R_n\}$.

If $n = 1$, then ra is called a *deterministic* role axiom. In this case we also write $T = \mathsf{con}(ra)$ instead of $T \in \mathsf{con}(ra)$.

A finite set $\mathfrak{R}$ of role axioms is called a *role box*.

Let $\mathsf{roles}(ra) =_{def} \{S, T, R_1, \ldots, R_n\}$, and $\mathsf{roles}(\mathfrak{R}) =_{def} \bigcup_{ra \in \mathfrak{R}} \mathsf{roles}(ra)$.

A role box $\mathfrak{R}$ is called *deterministic*, iff it contains only deterministic role axioms.

A role box $\mathfrak{R}$ is called *functional*, iff $\forall ra_1, ra_2 \in \mathfrak{R} : \mathsf{pre}(ra_1) = \mathsf{pre}(ra_2) \Rightarrow ra_1 = ra_2$. We can then use the function $\mathsf{ra}(S, T) = ra$ to refer to the unique role axiom ra with $\mathsf{pre}(ra) = (S, T)$ and define $\mathsf{con}(S, T) =_{def} \mathsf{con}(\mathsf{ra}(S, T))$.

A role box $\mathfrak{R}$ is called *complete*, iff $\forall R, S \in \mathsf{roles}(\mathfrak{R}) : \exists ra \in \mathfrak{R} : \mathsf{pre}(ra) = (R, S)$.

Let $\mathfrak{R}$ be a functional role box, C be a concept, and $R_? \in \mathcal{N_R}$, but $R_? \notin \mathsf{roles}(\mathfrak{R}) \cup \mathsf{roles}(C)$. Then, the *completion* of $\mathfrak{R}$ w.r.t. the concept C is defined as the role box

$$\mathfrak{R}(C) =_{def} \mathfrak{R} \cup \{\, R \circ S \sqsubseteq \sqcup_{T \in (\{R_?\} \cup \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R}))} T \mid \neg(\exists ra \in \mathfrak{R} : \mathsf{pre}(ra) = (R, S)), R, S \in (\{R_?\} \cup \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R}))\,\}.$$

Obviously, $\mathfrak{R}(C)$ is a complete and functional role box. Please note that, even if $\mathfrak{R}$ is already complete w.r.t. C such that $\forall R, S \in \mathsf{roles}(\mathfrak{R}) \cup \mathsf{roles}(C) : \exists ra \in \mathfrak{R} : \mathsf{pre}(ra) = (R, S)$, we would still have $\mathfrak{R}(C) \neq \mathfrak{R}$, due to the addition of role axioms $ra \in \mathfrak{R}(C) \setminus \mathfrak{R}$ with $R_? \in \mathsf{pre}(ra)$.

The role box $\mathfrak{R}$ is called *admissible* iff it is deterministic, functional, complete, and *associative*: $\forall R, S, T : \mathsf{con}(\mathsf{con}(R, S), T) = \mathsf{con}(R, \mathsf{con}(S, T))$.

The role box $\mathfrak{R}$ is called *admissible for the concept* C iff $\mathfrak{R}$ is admissible and additionally, $\mathsf{roles}(C) \subseteq \mathsf{roles}(\mathfrak{R})$.

[2] Almost all expressive description logics were also build in a step-by-step fashion, e.g., after $\mathcal{ALC}_{\mathcal{R}^+}$ had been introduced, $\mathcal{ALCH}_{\mathcal{R}^+}$ was defined, and so on.

According to the classes of allowed role boxes, we define $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$, $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$ and $\mathcal{ALC}_{\mathcal{RASG}}$ as follows:

- In $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$ we allow all role boxes.
- In $\mathcal{ALC}_{\mathcal{RA}}$ we allow only functional role boxes. The following holds: if $\mathfrak{R}$ is a functional role box, then $(C, \mathfrak{R})$ is satisfiable iff $(C, \mathfrak{R}(C))$ is, where $\mathfrak{R}(C)$ is the completion of $\mathfrak{R}$ w.r.t. C (the proof is pretty obvious and left out here for the sake of brevity). For the semantics, we require that all roles must be interpreted as disjoint, see below.[3]
- In $\mathcal{ALC}_{\mathcal{RASG}}$ we allow only role boxes that are *admissible* (see above) w.r.t. the considered concept C. We are studying the concept satisfiability problem of $(C, \mathfrak{R})$. Like in $\mathcal{ALC}_{\mathcal{RA}}$, we require that all roles must be interpreted as disjoint, see below. An admissible role box can be seen as defining the operation-table of a $\mathcal{S}$emi-$\mathcal{G}$roup (therefore the suffix $\mathcal{SG}$). For example, if we consider the operation-table of "+" modulo 4 on the natural numbers

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

  and interpret "+″ as "∘″, and assign for each number i a unique role name R_i, we get an admissible role box. Then, $(\exists R_1.\exists R_2.C) \sqcap \forall R_3.\neg C$ is unsatisfiable w.r.t. this role box.

  We like to mention that $\mathcal{ALC}_{\mathcal{RASG}}$, even though clearly a very restricted language, is a least as expressive as $\mathcal{ALC}_{\mathcal{R}^{+}}$ and $\mathcal{ALC}_{\mathcal{R}^{\oplus}}$.

[3]Please note that, due to the disjointness requirement, the restriction to functional role boxes is not really a restriction, since non-functional role boxes, e.g. with $\{R \circ S \sqsubseteq T_1, R \circ S \sqsubseteq T_2\} \subseteq \mathfrak{R}$ and $T_1 \neq T_2$, would not have a model anyway, if $R^{\mathcal{I}} \circ S^{\mathcal{I}} \neq \emptyset$.

In order to demonstrate the *consequences of disjointness for roles,* please consider

$$\mathfrak{R} = \{R \circ S \sqsubseteq A \sqcup B, S \circ T \sqsubseteq X \sqcup Y, A \circ T \sqsubseteq U, B \circ T \sqsubseteq V, R \circ X \sqsubseteq U, R \circ Y \sqsubseteq V\}.$$

Then,

$$(\exists R.((\exists S.\exists T.\top) \sqcap \forall Y.\bot) \sqcap \forall A.\bot, \mathfrak{R}(C))$$

is unsatisfiable, since $\forall A.\bot$ forces to choose $B \in \mathsf{con}(R, S)$, and $\forall Y.\bot$ forces to choose $X \in \mathsf{con}(S, T)$. Due to $B \circ T \sqsubseteq V$ and $R \circ X \sqsubseteq U$ there must be a non-empty intersection between U and V. The unsatisfiability is caused by a subtle interplay between the role box and the concept. Please observe that this role box is "associative" in a more general sense (since $(R \circ S) \circ T = \{U, V\} = R \circ (S \circ T)$), but not admissible in the sense defined here. Summing up, this example is satisfiable in $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, but unsatisfiable in $\mathcal{ALC}_{\mathcal{RA}}$. Since the role box is not admissible, the example is not expressible in $\mathcal{ALC}_{\mathcal{RASG}}$.

Additionally, a set of *concept inclusion axioms (GCIs)* can be specified. A set of these axioms is called a *general (or free) TBox:*

Definition 5 (Generalized Concept Inclusion Axiom, General TBox) If C and D are $\mathcal{ALC}$-concepts, then the expression $C \dot{\sqsubseteq} D$ is called a *generalized concept inclusion axiom,* or GCI for short. A finite set of GCI's is called a *general (or free) TBox*, $\mathfrak{T}$. We use $C \doteq D \in \mathfrak{T}$ as a shorthand for $\{C \dot{\sqsubseteq} D, D \dot{\sqsubseteq} C\} \subseteq \mathfrak{T}$. We define $\mathsf{sub}(\mathfrak{T}) =_{def} \{\, G \mid E \dot{\sqsubseteq} F \in \mathfrak{T}, G \in \mathsf{sub}(E) \cup \mathsf{sub}(F) \,\}$, and $\mathsf{roles}(\mathfrak{T}) =_{def} \{\, R \mid G \in \mathsf{sub}(\mathfrak{T}), R \in \mathsf{roles}(G) \,\}$.

The *semantics* of a concept is specified by giving a Tarski-style interpretation $\mathcal{I}$ that has to satisfy the following conditions:

Definition 6 (Interpretation) An *interpretation* $\mathcal{I} =_{def} (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty set $\Delta^\mathcal{I}$, called the domain of $\mathcal{I}$, and an interpretation function $\cdot^\mathcal{I}$ that maps every concept name to a subset of $\Delta^\mathcal{I}$, and every role name to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

In case of $\mathcal{ALC}_{\mathcal{RA}}$ and $\mathcal{ALC}_{\mathcal{RASG}}$, we additionally require that for all roles $R, S \in \mathcal{N}_\mathcal{R}$, $R \neq S$: $R^\mathcal{I} \cap S^\mathcal{I} = \emptyset$. All roles are interpreted as disjoint then.

The following functions on $\mathcal{I}$ will be used: The *universal relation* of $\mathcal{I}$ is defined as $\mathcal{UR}(\mathcal{I}) =_{def} \bigcup_{R \in \mathcal{N}_\mathcal{R}} R^\mathcal{I}$, and the universal relation w.r.t. a set of role names $\mathcal{R}$ as $\mathcal{UR}(\mathcal{I}, \mathcal{R}) =_{def} \bigcup_{R \in \mathcal{R}} R^\mathcal{I}$.

The *skeleton* of $\mathcal{I}$ is defined as $\mathcal{SKEL}(\mathcal{I}) =_{def}$
$\{\, \langle x, z \rangle \in \mathcal{UR}(\mathcal{I}) \mid \nexists y \in \Delta^\mathcal{I} : x \neq y, y \neq z, \langle x, y \rangle \in \mathcal{UR}(\mathcal{I}), \langle y, z \rangle \in \mathcal{UR}(\mathcal{I}) \,\}$.

If $<i,j> \in \mathcal{SKEL}(\mathcal{I})$, the edge is called a *direct edge*, otherwise an *indirect edge*.

If $<i,j> \in \mathcal{UR}(\mathcal{I})$, the edge is called an *incoming edge* for j.

Given an interpretation $\Delta^\mathcal{I}$, every (possibly compound) concept C can be uniquely interpreted (resp. "evaluated") by using the following definitions (we write $X^\mathcal{I}$ instead of $\cdot^\mathcal{I}(X)$):

$$\begin{array}{rll} (\neg C)^\mathcal{I} & =_{def} & \Delta^\mathcal{I} \setminus C^\mathcal{I} \\ (C \sqcap D)^\mathcal{I} & =_{def} & C^\mathcal{I} \cap D^\mathcal{I} \\ (C \sqcup D)^\mathcal{I} & =_{def} & C^\mathcal{I} \cup D^\mathcal{I} \\ (\exists R.C)^\mathcal{I} & =_{def} & \{\, i \in \Delta^\mathcal{I} \mid \exists j \in C^\mathcal{I} : <i,j> \in R^\mathcal{I} \,\} \\ (\forall R.C)^\mathcal{I} & =_{def} & \{\, i \in \Delta^\mathcal{I} \mid \forall j : <i,j> \in R^\mathcal{I} \Rightarrow j \in C^\mathcal{I} \,\} \end{array}$$

It is therefore sufficient to provide the interpretations for the concept *names* and roles, since the extension $C^\mathcal{I}$ of every concept C is uniquely determined then.

In the following we specify under which conditions a given interpretation is a *model* of a syntactic entity (we also say an interpretation *satisfies* a syntactic entity):

Definition 7 (The Model Relationship) An interpretation $\mathcal{I}$ is a model of a concept C, written $\mathcal{I} \models C$, iff $C^\mathcal{I} \neq \emptyset$.

An interpretation $\mathcal{I}$ is a model of a role axiom $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$, written $\mathcal{I} \models S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$, iff $S^\mathcal{I} \circ T^\mathcal{I} \subseteq R_1^\mathcal{I} \cup \cdots \cup R_n^\mathcal{I}$.

An interpretation $\mathcal{I}$ is a model of a role box $\mathfrak{R}$, written $\mathcal{I} \models \mathfrak{R}$, iff for all role axioms $ra \in \mathfrak{R}$: $\mathcal{I} \models ra$.

An interpretation $\mathcal{I}$ is a model of a GCI $C \dot\sqsubseteq D$, written $\mathcal{I} \models C \dot\sqsubseteq D$, iff $C^\mathcal{I} \subseteq D^\mathcal{I}$.

An interpretation $\mathcal{I}$ is a model of a TBox $\mathfrak{T}$, written $\mathcal{I} \models \mathfrak{T}$, iff for all GCIs $g \in \mathfrak{T}$: $\mathcal{I} \models g$.

An interpretation $\mathcal{I}$ is a model of $(C, \mathfrak{R})$, written $\mathcal{I} \models (C, \mathfrak{R})$, iff $\mathcal{I} \models C$ and $\mathcal{I} \models \mathfrak{R}$.

An interpretation $\mathcal{I}$ is a model of $(C, \mathfrak{T})$, written $\mathcal{I} \models (C, \mathfrak{T})$, iff $\mathcal{I} \models C$ and $\mathcal{I} \models \mathfrak{T}$.

An interpretation $\mathcal{I}$ is a model of $(C, \mathfrak{R}, \mathfrak{T})$, written $\mathcal{I} \models (C, \mathfrak{R}, \mathfrak{T})$, iff $\mathcal{I} \models C$, $\mathcal{I} \models \mathfrak{R}$ and $\mathcal{I} \models \mathfrak{T}$.

An important relationship between concepts is the subsumption relationship, which is a partial ordering on concepts w.r.t. their specificity:

Definition 8 (Subsumption Relationship) A concept D *subsumes* a concept C, $C \sqsubseteq D$ (w.r.t. to $\mathfrak{T}$ and/or $\mathfrak{R}$), iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$ (that are also models of $\mathfrak{T}$ and/or $\mathfrak{R}$, respectively).

We like to mention that the three proposed logics are expressive enough to allow for the so-called *internalization* of general TBoxes.

Since a full negation operator if provided, the subsumption problem can be reduced to the concept satisfiability problem: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable.

It should be noticed that a satisfiability tester for any of the three proposed logics would also be able to determine satisfiability resp. subsumption w.r.t. general TBoxes (see below; internalization of $\mathfrak{T}$ can be used).

For the following proofs we need some auxiliary propositions which are obviously true and deserve no proofs:

Proposition 1 If $\mathcal{I} \models (C, \mathfrak{R})$, then for every interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}'})$ with $D^{\mathcal{I}'} = D^{\mathcal{I}}$ for all concept names $D \in \mathsf{sub}(C) \cap \mathcal{N_C}$ and $R^{\mathcal{I}'} = R^{\mathcal{I}}$ for all role names $R \in \mathsf{roles}(C, \mathfrak{R})$: $\mathcal{I}' \models (C, \mathfrak{R})$.

Proposition 2 $(C, \mathfrak{R})$ is satisfiable iff $(\mathsf{NNF}(C), \mathfrak{R})$ is satisfiable. $\mathsf{NNF}(C)$ returns the *negation normal form* of C, in which the negation sign appears only in front of atomic concept names. The NNF can be obtained by "pushing the negation sign inwards", e.g. by exhaustively applying the rules $\neg(\neg C_1) \to C_1$, $\neg(C_1 \sqcap C_2) \to \neg C_1 \sqcup \neg C_2$, $\neg(C_1 \sqcup C_2) \to \neg C_1 \sqcap \neg C_2$, $\neg\forall R.C_1 \to \exists R.\neg C_1$ and $\neg\exists R.C_1 \to \forall R.\neg C_1$. In the following we assume w.l.o.g. that C is in NNF.

Proposition 3 Neither $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$, $\mathcal{ALC}_{\mathcal{RA}}$, nor $\mathcal{ALC}_{\mathcal{RASG}}$ have the *finite model property*, i.e. there are pairs $(C, \mathfrak{R})$ that have no finite models.

Proof 1 As a counter-example to a finite model property assumption in $\mathcal{ALC}_{\mathcal{RA}}$, please consider

$$(\exists R.\exists R.\top) \sqcap (\forall S.\exists R.\top)$$

w.r.t.

$$\{R \circ R \sqsubseteq S, R \circ S \sqsubseteq S, S \circ R \sqsubseteq S, S \circ S \sqsubseteq S\},$$

which has no finite model (see [16],[15] for a proof). Since this is clearly also an *admissible* role-box for the considered concept term, this also shows that $\mathcal{ALC}_{\mathcal{RASG}}$ does not have the finite model property. Concerning $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$, the proposition follows indirectly, since $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$ is undecidable (see [15]). It is well-known that a description logic is decidable, if it has the finite model property and can be reduced to first-order predicate logic formulas. Since the latter is true, the former must be false, because $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ would be decidable otherwise. □

3 The Relationship to Grammar Logics

Recently it turned out that there is an interesting correspondence of $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ with a special class of normal multimodal logics called *inclusion modal logics*, or *grammar logics* (see [4],[3]). The correspondence has been pointed out by Demri ([4]). We will only briefly sketch the basic idea of inclusion modal logics, see [3] for a thorough discussion. Demri also provides an undecidability proof for $\mathcal{ALC}_{\mathcal{RA}^\ominus}$, based on the original proof of Baldoni who showed that context-free grammar logics are undecidable, but the $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ undecidability result in [15] has been found independently from Demri's and Baldoni's proofs and without any knowledge of the correspondence. The proof is given by a reduction from the (undecidable) non-empty intersection problem of a special class of context-free grammars which are similar to context-free grammars in Chomsky Normal Form.[4]

The original motivation for the introduction of grammar logics was to simulate the behavior of formal grammars by means of modal logic. Grammar logics are characterized by a set of modal inclusion axiom schemas of the form $[t_1][t_2]\ldots[t_n]\psi \Rightarrow [s_1][s_2]\ldots[s_m]\psi$. The intended idea was that each production rule $t_1t_2\ldots t_n \rightarrow s_1s_2\ldots s_m$ of a formal grammar gives rise to one modal inclusion axiom schema of the form $[t_1][t_2]\ldots[t_n]\psi \Rightarrow [s_1][s_2]\ldots[s_m]\psi$ (please note that $\Rightarrow$ means implication, whereas $\rightarrow$ means "derives in one step"). In [3] it is shown that a set of inclusion axiom schemas *characterizes* a special *class of frames* $\mathcal{F}_\mathcal{A}$, called inclusion frames, in which for each inclusion axiom schema an appropriate inclusion relationship between the accessibility relations $\mathcal{R}_i$ holds; as usual, $\mathcal{R}_i$ is the accessibility relation corresponding to the modality $[i]$. It is shown that each axiom schema $[t_1][t_2]\ldots[t_n]\psi \Rightarrow [s_1][s_2]\ldots[s_m]\psi$ characterizes the class of inclusion frames $\mathcal{F}_\mathcal{A}$ where $\mathcal{R}_{s_1} \circ \mathcal{R}_{s_2} \circ \cdots \circ \mathcal{R}_{s_m} \subseteq \mathcal{R}_{t_1} \circ \mathcal{R}_{t_2} \circ \cdots \circ \mathcal{R}_{t_n}$ holds; similar as, for example, the modal logic axiom schema $T = [a]\psi \Rightarrow \psi$ characterizes the class of frames where $\mathcal{R}_a$ is reflexive (see [5]).[5]

In description logic terminology, the accessibility relations correspond to inter-

[4]The non-empty intersection problem for two grammars $\mathcal{G}_1$ and $\mathcal{G}_2$ is to decide whether $\mathcal{L}(\mathcal{G}_1) \cap \mathcal{L}(\mathcal{G}_2) = \emptyset$.

[5]In order to evaluate the truth-value of a modal logic formula, a *Kripke interpretation* is needed. Such a Kripke interpretation consists of a *frame* and a *valuation function*. The frame provides a *set of worlds* and, for each modality i, an accessibility relation $\mathcal{R}_i$. The *valuation function* determines for each world and proposition whether this proposition holds at this world or not.

preted roles (e.g. the accessibility relation $\mathcal{R}_R$ corresponds to $R^\mathcal{I}$). If an appropriate class of inclusion frames $\mathcal{F}_\mathcal{A}$ is considered, an inclusion axiom schema of the form $[T]\psi \Rightarrow [R][S]\psi$ corresponds to a role axiom of the form $R \circ S \sqsubseteq T$. Appropriate means that $\mathcal{R}_R \circ \mathcal{R}_S \subseteq \mathcal{R}_T$ has to hold. As an example, w.r.t. the class of Kripke interpretations that are *based on such appropriate frames* $\mathcal{F}_\mathcal{A}$, the formula $(\langle R\rangle\langle S\rangle p) \wedge [T]\neg p$ is unsatisfiable. The same holds for $((\exists R.\exists S.p) \sqcap \forall T.\neg p, \{R \circ S \sqsubseteq T\})$, since the role axiom *enforces* the appropriate "inclusion property" $R^\mathcal{I} \circ S^\mathcal{I} \subseteq T^\mathcal{I}$. Considering the modal logic formula, one has to observe that there are indeed Kripke interpretations where this formula is satisfiable; i.e. a broader class of models than in the $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ equivalent is admitted. However, these Kripke interpretations are *not* based on appropriate inclusion frames $\mathcal{F}_\mathcal{A}$, where $\mathcal{R}_R \circ \mathcal{R}_S \subseteq \mathcal{R}_T$ holds.

By analyzing various types of grammars, certain decidability and undecidability results concerning satisfiability (validity) of formulas w.r.t. Kripke interpretations that are *based on appropriate inclusion frames* have been found: it is already shown in [3] that the validity problem for the class of context-free inclusion modal logics is undecidable. It should be noted that $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ even admits axioms of the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$ which correspond to modal axiom schemas of the form $([R_1]\psi) \wedge ([R_2]\psi) \wedge \cdots \wedge ([R_n]\psi) \Rightarrow [S][T]\psi$,[6] and these axiom schemas are not in the class of inclusion modal logics. However, in the undecidability proof of $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ it is shown that already $\mathcal{ALU}_{\mathcal{RA}^\ominus}$ with *deterministic* role boxes is undecidable. Therefore, this undecidable fragment can indeed be understood as an inclusion modal logic and one could claim that the undecidability of $\mathcal{ALC}_{\mathcal{RA}^\ominus}$ is already an obvious consequence from the work in [3]. However, the correspondence was unknown until Demri's paper (see [4]).

4 $\mathcal{ALC}_{\mathcal{RASG}}$ is EXPTIME-Complete

In the following, we design a decision procedure for $\mathcal{ALC}_{\mathcal{RASG}}$: given a concept C and a role box $\mathfrak{R}$ that is admissible for C (recall that a role box is admissible for a concept C iff it is deterministic, functional, complete, associative and $\mathsf{roles}(C) \subseteq \mathsf{roles}(\mathfrak{R})$) as input to the algorithm SAT, the algorithm decides whether there is a model $\mathcal{I}$ of $(C, \mathfrak{R})$: $\mathcal{I} \models (C, \mathfrak{R})$. We adopt and adapt the proof-techniques firstly used by Horrocks, Sattler et al. (e.g., see [7, 8]).

In a similar way as for other description logics, the decision procedure is given by means of a non-deterministic *tableaux algorithm* that constructs a so-called finite *completion tree*. *Soundness* of this algorithm is proven by showing that a so-called *tableau* can be constructed from a complete and clash-free completion tree

[6] Thanks to Stéphane Demri

that has been generated by the algorithm. *Completeness* is proven by showing how to construct a clash-free completion tree from a given tableau. Before we can proceed with the algorithm, we therefore need to define the auxiliary data structure tableau, which will turn out to be an appropriate abstraction of a special class of models.

At a first sight it might not make sense to define a tableaux algorithm for such a restricted description logic like $\mathcal{ALC}_{\mathcal{RASG}}$ (again, recall that it is at least as expressive as $\mathcal{ALC}_{\mathcal{R}^+}$). However, the calculus is presented *in the form given here* for a number of reasons:

1. It is very similar to the calculus which has been given for $\mathcal{ALC}_{\mathcal{RA}}$ in [16]. The basic ideas are identical. Therefore, defining the $\mathcal{ALC}_{\mathcal{RASG}}$ calculus *in this way* might yield new insights on the decidability status of $\mathcal{ALC}_{\mathcal{RA}}$.

2. The given calculus should be extendible. We are planning to investigate different (less restricted) classes of role boxes with the same calculus by making only small changes to the blocking condition and/or the so-called $\exists\forall$-rule (see below).

3. For the same reason, the calculus and the definition of tableaus are slightly more complicated than needed. This has also the additional advantage that some properties can be proven more easily. Hopefully, this allows for an adaptation of the framework if a less-restricted class of role boxes is considered, without having to redesign everything.

4. We wanted to demonstrate that the basic ideas behind the $\mathcal{ALC}_{\mathcal{R}^+}$ calculus (see [10]) can be further stretched, applying also to more general logics like $\mathcal{ALC}_{\mathcal{RASG}}$. In fact, the $\mathcal{ALC}_{\mathcal{RASG}}$ calculus might be seen as a generalization of some of the ideas present in the $\mathcal{ALC}_{\mathcal{R}^+}$ calculus ("propagating" constraints of the form "$\forall R.C$", where R is a transitively closed role).

5. The given (non-deterministic) tableaux calculus is not optimal in terms of computational complexity. As shown below, $\mathcal{ALC}_{\mathcal{RASG}}$ is EXPTIME-complete, but the calculus is *not* an EXPTIME-decision procedure. Therefore, in order to prove the upper-bound complexity of $\mathcal{ALC}_{\mathcal{RASG}}$, a different method was needed. We give an EXPTIME-reduction to $\mathcal{ALC}$-satisfiability w.r.t. general TBoxes. Therefore, an arbitrary $\mathcal{ALC}$-reasoner that can handle GCIs could be used to decide $\mathcal{ALC}_{\mathcal{RASG}}$-satisfiability. However, the reduction was designed to show EXPTIME-completeness and would likely not be efficient when used as an $\mathcal{ALC}_{\mathcal{RASG}}$-decision procedure. A starting-point for an $\mathcal{ALC}_{\mathcal{RASG}}$-reasoner would therefore be the given tableaux calculus, since it will be much more efficient in the average case and might also be augmentable with special optimization techniques.

4.1 The Tableau

Basically, a *tableau* is a possibly infinite tree whose edges $\mathcal{E}$ are labeled with role names, and whose nodes are labeled with constraints enforced on these nodes (see [8]). The node labeling function is called $\mathcal{L}_\mathcal{N}$, and we write $\mathcal{L}_\mathcal{N}(x)$ to refer to the label of node x. The elements of $\mathcal{L}_\mathcal{N}(x)$ are called *constraints.* Since edges are labeled with role names, we refer to the set of edges corresponding to a given role $R \in \mathcal{N}_\mathcal{R}$ as $\mathcal{E}_R$.

A tableau for $(C, \mathfrak{R})$ is just an other representation of a special model $\mathcal{I}$ of $(C, \mathfrak{R})$ (please note that C is already in NNF, see Proposition 2). We call these models *tree skeleton models.*

Let $\mathcal{I}$ be the tree skeleton model corresponding to some given tableau. Then, $\mathcal{SKEL}(\mathcal{I})$ corresponds to the labeled tree of this tableau. If a node y in the tableau is an R-successor of the node x due to some constraint $\exists R \ldots$ enforced on node x, $\exists R \ldots \in \mathcal{L}_\mathcal{N}(x)$, then w.r.t. $\mathcal{I}$ we have $<x, y> \in R^\mathcal{I} \cap \mathcal{SKEL}(\mathcal{I})$; i.e. $<x, y>$ is a *direct* R-edge. However, the *indirect edges* which might be present in $\mathcal{I}$ due to role axioms cannot be represented in the tableau in this way, since a tableau is a labeled *tree.* For example, if $R \circ S \sqsubseteq T$, then a model $\mathcal{I}$ with $<x_0, x_1> \in R^\mathcal{I}$, $<x_1, x_2> \in S^\mathcal{I}$ must satisfy $<x_0, x_2> \in T^\mathcal{I}$. Therefore, every (direct or indirect) incoming edge for a node x in the model is represented in the tableau by a special *annotated all constraint* of the form $(\forall U.D)_{S,w} \in \mathcal{L}_\mathcal{N}(x)$, where S represents the type of the incoming edge, and w is a word of role names, denoting a path in the tree, leading from the individual from which the edge originates to x. In our example we would have $(\forall \ldots)_{T,RS} \in \mathcal{L}_\mathcal{N}(x_2)$ due to $<x_0, x_2> \in T^\mathcal{I}$, $<x_0, x_1> \in \mathcal{E}_\mathcal{R}$, $<x_1, x_2> \in \mathcal{E}_\mathcal{S}$, and $(\forall \ldots)_{S,S} \in \mathcal{L}_\mathcal{N}(x_2)$ due to $<x_1, x_2> \in S^\mathcal{I}$, $<x_1, x_2> \in \mathcal{E}_\mathcal{S}$. Assume that $\forall U.D \in \mathcal{L}_\mathcal{N}(x_0)$. Then, the presence of the constraint $(\forall U.D)_{T,RS} \in \mathcal{L}_\mathcal{N}(x_2)$ is ensured (see below). Since x_2 is an indirect T-successor of x_0 and not a U-successor, $D \notin \mathcal{L}_\mathcal{N}(x_2)$. If we additionally had $\forall T.D \in \mathcal{L}_\mathcal{N}(x_0)$, then also $(\forall T.D)_{T,RS} \in \mathcal{L}_\mathcal{N}(x_2)$, and $D \in \mathcal{L}_\mathcal{N}(x_2)$. Whenever a constraint $(\forall U.D)_{T,w} \in \mathcal{L}_\mathcal{N}(x)$ with $U = T$ is encountered, $D \in \mathcal{L}_\mathcal{N}(x)$ is ensured, since the qualification is applicable to the node x. Since the indirect edges are represented by constraints of the form $(\forall U.D)_{T,w}$, it is necessary to ensure the presence of them. This is achieved by adding a "dummy constraint" of the form $\forall R_?.\top$ for some $R_? \in \mathcal{N}_\mathcal{R}$ to the label of each node in the tableau.

In order to facilitate the subsequent definitions, the following auxiliary data structure is defined:

Definition 9 (Concept Tree) Let C be a concept and $\mathfrak{R}$ be an admissible role box for C. A *concept tree* for $(C, \mathfrak{R})$ is a tuple $(\mathcal{N}, \mathcal{L}_\mathcal{N}, \mathcal{E}, \mathcal{L}_\mathcal{E})$, where $\mathcal{N}$ is a set of nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of edges. The graph $(\mathcal{N}, \mathcal{E})$ is a (possibly infinite) tree. The (total) labeling function $\mathcal{L}_\mathcal{E} : \mathcal{E} \rightarrow \mathsf{roles}(\mathfrak{R})$ associates edges with role names. For a role $R \in \mathsf{roles}(\mathfrak{R})$, the set of R-edges is $\mathcal{E}_R =_{def} \{ <x, y> \mid (<x, y>, R) \in \mathcal{L}_\mathcal{E} \}$. Additionally, $\mathcal{L}_\mathcal{N}$ is a (total) node labeling function:

$$\mathcal{L}_\mathcal{N} : \mathcal{N} \rightarrow \mathsf{sub}(C) \cup \{\forall R_?.\top\} \cup \{ (\forall R.C_1)_{S,w} \mid \forall R.C_1 \in \mathsf{sub}(C) \text{ or } \forall R.C_1 = \forall R_?.\top, R, S \in \mathsf{roles}(\mathfrak{R}), w \in \mathsf{roles}(\mathfrak{R})^+ \}.$$

The elements in the domain of $\mathcal{L}_\mathcal{N}$ are also called *constraints*.. If $<x, y> \in \mathcal{E}$, y is called a *successor* of x, and x is called a *predecessor* of y. If $<x, y> \in \mathcal{E}^+$, x is called an *ancestor* of y, and y is called a *descendant* of x. Let $\mathsf{w\_ancestor}(y, w) =_{def} x$ iff $w = R_1 R_2 \ldots R_n$, where $w \in \mathsf{roles}(\mathfrak{R})^+$, $<x, x_1> \in \mathcal{E}_{R_1}$, $<x_1, x_2> \in \mathcal{E}_{R_2}$, $\ldots$, $<x_{n-1}, y> \in \mathcal{E}_{R_n}$. Please note that the w_ancestor is uniquely defined, since we have a tree.

A *tableau* can then be defined as follows:

Definition 10 (Tableau) A tableau $\mathcal{T}$ for $(C, \mathfrak{R})$ is a concept tree for which the following additional conditions hold:

1. There is some node $x_0 \in \mathcal{N}$ with $C \in \mathcal{L}_\mathcal{N}(x_0)$.
2. For all $x, y \in \mathcal{N}$, for all $C_i \in \mathsf{sub}(C)$, and for all $(\forall R_i.C_i)_{S_i,w} \in \mathcal{L}$ with $R, R_i, S, S_i \in \mathsf{roles}(\mathfrak{R})$, $w \in \mathsf{roles}(\mathfrak{R})^+$, and for some $R_? \in \mathsf{roles}(\mathfrak{R})$, we have
    (a) if $C_1 \in \mathcal{L}_\mathcal{N}(x)$, then $(\neg C_1) \notin \mathcal{L}_\mathcal{N}(x)$,
    (b) if $(C_1 \sqcap C_2) \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$ and $C_2 \in \mathcal{L}_\mathcal{N}(x)$,
    (c) if $(C_1 \sqcup C_2) \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$ or $C_2 \in \mathcal{L}_\mathcal{N}(x)$,
    (d) if $(\exists R.C_1) \in \mathcal{L}_\mathcal{N}(x)$, then there is some y such that $<x, y> \in \mathcal{E}_R$ and $C_1 \in \mathcal{L}_\mathcal{N}(y)$,
    (e) if $(\forall R.C_1)_{R,w} \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$,
    (f) $(\forall R_?.\top) \in \mathcal{L}_\mathcal{N}(x)$,
    (g) $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}_\mathcal{N}(y)$ iff $S_1 = S_2$ and there is some x with $<x, y> \in \mathcal{E}_{S_1}$ and $(\forall R.C_1) \in \mathcal{L}_\mathcal{N}(x)$,
    (h) $(\forall R.C_1)_{S_3,wS_2} \in \mathcal{L}_\mathcal{N}(y)$ with $|w| \geq 1$ iff there is some x with $<x, y> \in \mathcal{E}_{S_2}$, $(\forall R.C_1)_{S_1,w} \in \mathcal{L}_\mathcal{N}(x)$, and $S_3 = \mathsf{con}(S_1, S_2)$.

Lemma 1 $(C, \mathfrak{R})$ is satisfiable iff there exists a tableau $\mathcal{T}$ for $(C, \mathfrak{R})$.

Proof 2 "$\Leftarrow$" If $\mathcal{T} = (\mathcal{N}, \mathcal{L}_\mathcal{N}, \mathcal{E}, \mathcal{L}_\mathcal{E})$ is a tableau for $(C, \mathfrak{R})$, a *tree skeleton model* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ of $(C, \mathfrak{R})$ can be constructed as follows:

- $\Delta^{\mathcal{I}} =_{def} \mathcal{N}$,
- $C_1^{\mathcal{I}} =_{def} \{\, x \mid C_1 \in \mathcal{L}_{\mathcal{N}}(x) \,\}$ for all $C_1 \in \mathcal{N}_{\mathcal{C}} \cap \mathsf{sub}(C)$,
- $C_1^{\mathcal{I}} =_{def} \emptyset$ for all other concept names $C_1 \in \mathcal{N}_{\mathcal{C}} \setminus \mathsf{sub}(C)$,
- $R^{\mathcal{I}} =_{def} \{\, \langle x, y \rangle \mid \mathsf{w\_ancestor}(y, w) = x, (\forall S.C_1)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y) \,\}$ for all role names $R \in \mathsf{roles}(\mathfrak{R}(C))$,
- $R^{\mathcal{I}} =_{def} \emptyset$ for all other roles $R \in \mathcal{N}_{\mathcal{R}} \setminus \mathsf{roles}(\mathfrak{R})$.

First of all, due to Proposition 1 we can safely interpret all unmentioned roles (those not in $\mathsf{roles}(\mathfrak{R})$) and concept names (those not in $\mathsf{sub}(C)$) with the empty set.

We show that all roles are interpreted as disjoint: assume the contrary. Then there must be some roles $R, S \in \mathcal{N}_{\mathcal{R}}$, $R \neq S$ with $R^{\mathcal{I}} \cap S^{\mathcal{I}} \neq \emptyset$. Due to the definition of $\cdot^{\mathcal{I}}$, $\langle x, y \rangle \in R^{\mathcal{I}} \cap S^{\mathcal{I}}$ with $R \neq S$ iff $x = \mathsf{w\_ancestor}(y, w)$, $(\forall S_1.C_1)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y)$ and $(\forall S_2.C_2)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$. Please note that S_1, S_2, C_1, C_2, etc. do not matter here – they are just used as place-holders. We will therefore write $(\forall ...)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y)$, $(\forall ...)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$. Using induction on w we show that for each node y, $(\forall ...)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y)$ and $(\forall ...)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$ implies $R = S$. If $|w| = 1$, this is an immediate consequence of Condition 2g and of the fact that y has only one predecessor, since we have a tree. For the induction step, $(\forall ...)_{R,wT} \in \mathcal{L}_{\mathcal{N}}(y)$ and $(\forall ...)_{S,wT} \in \mathcal{L}_{\mathcal{N}}(y)$ iff there is some x with $\langle x, y \rangle \in \mathcal{E}_T$, $(\forall ...)_{R',w} \in \mathcal{L}_{\mathcal{N}}(x)$, $(\forall ...)_{S',w} \in \mathcal{L}_{\mathcal{N}}(x)$. Due to the hypothesis, $R' = S'$. But then we have $R = \mathsf{con}(R', T)$, $S = \mathsf{con}(R', T)$, and since the role box is admissible and therefore contains only deterministic role axioms it follows that $R = S$.

We show $\mathcal{I} \models \mathfrak{R}$: assume the contrary. Then there must be some role axiom $ra \in \mathfrak{R}$ that ist not satisfied by $\mathcal{I}$. This is the case iff there are some x, y, z with $\langle x, y \rangle \in R^{\mathcal{I}}$, $\langle y, z \rangle \in S^{\mathcal{I}}$, and either $\langle x, z \rangle \notin \mathcal{UR}(\mathcal{I})$ (note that an admissible role box is complete, and therefore, $\langle x, z \rangle \notin \mathcal{UR}(\mathcal{I})$ cannot occur in a model of a complete role box), or $\langle x, z \rangle \in T^{\mathcal{I}}$, but $T \neq \mathsf{con}(ra) = \mathsf{con}(R, S)$. Due to the definition of $\cdot^{\mathcal{I}}$, $\langle x, y \rangle \in R^{\mathcal{I}}$, $\langle y, z \rangle \in S^{\mathcal{I}}$, iff $x = \mathsf{w\_ancestor}(y, w_1)$, $y = \mathsf{w\_ancestor}(z, w_2)$, $(\forall ...)_{R,w_1} \in \mathcal{L}_{\mathcal{N}}(y)$ and $(\forall ...)_{S,w_2} \in \mathcal{L}_{\mathcal{N}}(z)$. Let $w = w_1 w_2$. Obviously, $x = \mathsf{w\_ancestor}(z, w)$. In the first case, $\langle x, z \rangle \notin \mathcal{UR}(\mathcal{I})$ iff $(\forall ...)_{T,w} \notin \mathcal{L}_{\mathcal{N}}(z)$, for all $T \in \mathcal{N}_{\mathcal{R}}$. However, due to Property 2f, we have $\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x)$. Since $x = \mathsf{w\_ancestor}(z, w)$ and $\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x)$, an iterated exploitation of the Properties 2g and 2h shows that $(\forall R_?.\top)_{T,w} \in \mathcal{L}_{\mathcal{N}}(z)$, for some $T \in \mathsf{roles}(\mathfrak{R})$. Contradiction. In the second case, $\langle x, z \rangle \in T^{\mathcal{I}}$ iff $(\forall ...)_{T,w} \in \mathcal{L}_{\mathcal{N}}(z)$. Again use induction over $|w|$: if $|w| = 2$ we have $w = RS$ and it is obviously the case that $T = \mathsf{con}(R, S)$, due to Properties 2g and 2h. If $|w| \geq 3$, let $w = w'U = w_1 w_2$. Now, $(\forall ...)_{T,w'U} \in \mathcal{L}_{\mathcal{N}}(z)$ implies $(\forall ...)_{T',w'} \in \mathcal{L}_{\mathcal{N}}(y')$, with $\langle y', z \rangle \in \mathcal{E}_{\mathcal{U}}$, $T = \mathsf{con}(T', U)$. We also have $(\forall ...)_{S,w''U} \in \mathcal{L}_{\mathcal{N}}(z)$, where $w_2 = w''U$, and $(\forall ...)_{V,w''} \in \mathcal{L}_{\mathcal{N}}(y')$. Due to Property 2h, $S = \mathsf{con}(V,U)$. Please observe that $w' = w_1 w''$ and therefore, due to the induction hypothesis, $T' = \mathsf{con}(R,V)$. But now, due to the associativity of the role box $\mathsf{con}(\mathsf{con}(R,V),U) = \mathsf{con}(R,\mathsf{con}(V,U))$. Since $T' = \mathsf{con}(R,V)$, $S = \mathsf{con}(V,U)$ this means $\mathsf{con}(T',U) = \mathsf{con}(R,S)$. Since we know that $T = \mathsf{con}(T',U)$ we have obtained a contradiction to our assumption, since also $T = \mathsf{con}(R,S)$. Summing up we have shown that $\mathcal{I} \models \mathfrak{R}$.

In the following, we will show by structural induction on the concept E that if $E \in \mathcal{L}_{\mathcal{N}}(x)$, then also $x \in E^{\mathcal{I}}$. Assuming this, from $C \in \mathcal{L}_{\mathcal{N}}(x_0)$ (due to Property 1 in Definition 10) it follows that $x_0 \in C^{\mathcal{I}}$. Since $C^{\mathcal{I}} \neq \emptyset$, we have $\mathcal{I} \models C$.

Let $E \in \mathcal{L}_{\mathcal{N}}(x)$ with $E \in \mathsf{sub}(C)$:

1. If E is a concept name, then $x \in E^{\mathcal{I}}$ by definition (since $E \in \mathsf{sub}(C)$).
2. If $E = \neg C_1$, then $C_1 \in \mathcal{N}_{\mathcal{C}}$, because $C_1 \in \mathsf{sub}(C)$ and C is in NNF. Since $\neg C_1 \in \mathcal{L}_{\mathcal{N}}(x)$, it holds that $C_1 \notin \mathcal{L}_{\mathcal{N}}(x)$, because assuming $C_1 \in \mathcal{L}_{\mathcal{N}}(x)$ yields to to the conclusion that $\mathcal{T}$ is not a tableau, due to Property 2a. Since $C_1 \notin \mathcal{L}_{\mathcal{N}}(x)$ we have $x \notin C_1^{\mathcal{I}}$, and therefore $x \in (\neg C_1)^{\mathcal{I}}$.
3. If $E = (C_1 \sqcap C_2)$, then due to Property 2b in Definition 10 it holds that $C_1 \in \mathcal{L}_{\mathcal{N}}(x)$ and $C_2 \in \mathcal{L}_{\mathcal{N}}(x)$. Therefore, from the induction hypothesis we have $x \in C_1^{\mathcal{I}}$ and $x \in C_2^{\mathcal{I}}$. Hence $x \in (C_1 \sqcap C_2)^{\mathcal{I}}$.
4. If $E = (C_1 \sqcup C_2)$, then due to Property 2c in Definition 10 it holds that $C_1 \in \mathcal{L}_{\mathcal{N}}(x)$ or $C_2 \in \mathcal{L}_{\mathcal{N}}(x)$. Therefore, from the induction hypothesis we have $x \in C_1^{\mathcal{I}}$ or $x \in C_2^{\mathcal{I}}$. Hence $x \in (C_1 \sqcup C_2)^{\mathcal{I}}$.
5. If $E = (\exists S.C_1)$, then there is some $y \in \mathcal{N}$ such that $<x,y> \in \mathcal{E}_{\mathcal{S}}$ and $C_1 \in \mathcal{L}_{\mathcal{N}}(y)$, due to Property 2d in Definition 10. By the induction hypothesis it holds that $y \in C_1^{\mathcal{I}}$. Due to Property 2f, we have $\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x)$, and Property 2g ensures $(\forall R_?.\top)_{S,S} \in \mathcal{L}_{\mathcal{N}}(y)$. By the definition of $S^{\mathcal{I}}$ we finally have $<x,y> \in S^{\mathcal{I}}$, since $x = \mathsf{w\_ancestor}(y,S)$. We therefore have $x \in (\exists S.C_1)^{\mathcal{I}}$.
6. If $E = (\forall S.C_1)$, then we have to show that for every y with $<x,y> \in S^{\mathcal{I}}$ it holds that $C_1 \in \mathcal{L}_{\mathcal{N}}(y)$. Due to the induction hypothesis $C_1 \in \mathcal{L}_{\mathcal{N}}(y)$ implies $y \in C_1^{\mathcal{I}}$, and therefore $x \in (\forall S.C_1)^{\mathcal{I}}$ if we can show this for all y with $<x,y> \in S^{\mathcal{I}}$. Due to the definition of $S^{\mathcal{I}}$, $<x,y> \in S^{\mathcal{I}}$ iff $(\forall S_1.C_2)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$ and $\mathsf{w\_ancestor}(y,w) = x$. Since $\forall S.C_1 \in \mathcal{L}_{\mathcal{N}}(x)$ and $\mathsf{w\_ancestor}(y,w) = x$, we have $(\forall S.C_1)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y)$, for some role R, due to Properties 2g and 2h. We already observed (see above) that $(\forall S.C_1)_{R,w} \in \mathcal{L}_{\mathcal{N}}(y)$ and $(\forall S_1.C_2)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$ implies $R = S$, since the role box is deterministic. Hence $(\forall S.C_1)_{S,w} \in \mathcal{L}_{\mathcal{N}}(y)$, and finally $C_1 \in \mathcal{L}_{\mathcal{N}}(y)$ due to Property 2e.

Summing up, we have shown that $\mathcal{I} \models (C, \mathfrak{R})$, and all roles are interpreted as disjoint.

"$\Rightarrow$" If $\mathcal{I} \models (C, \mathfrak{R})$, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, then a tableau $\mathcal{T} = (\mathcal{N}, \mathcal{L}_{\mathcal{N}}, \mathcal{E}, \mathcal{L}_{\mathcal{E}})$ for $(C, \mathfrak{R})$ can be constructed. Since a tableau is required to be a (possibly infinite) tree, but a model may be an arbitrary graph, but not necessarily a tree skeleton model, we cannot simply assign $\mathcal{N} =_{def} \Delta^{\mathcal{I}}$. Intuitively, the tableau is constructed by *unraveling* or *traversing* the model, collecting the required information. Please note that this also gives a method how to concert an arbitrary model into a tree skeleton model: first transform it into a tableau as shown subsequently, and then transform this tableau back into a tree skeleton model using the model construction in the "$\Leftarrow$"-part of this proof, see above.

In the construction, each node $x \in \mathcal{N}$ in the tableau $\mathcal{T}$ corresponds to a *path* in $\mathcal{I}$. A path in $\mathcal{I}$ is inductively defined as follows:

- For some (but only one) $i_0 \in \Delta^{\mathcal{I}}$ with $i_0 \in C^{\mathcal{I}}$, $[i_0]$ is a path in $\mathcal{I}$.
- If $[i_0, \ldots, i_m]$ (possibly $p = [i_0]$) is a path in $\mathcal{I}$ and $i_m \in (\exists R.C_1)^{\mathcal{I}}$, $\langle i_m, i_n \rangle \in R^{\mathcal{I}}$ with $i_n \in C_1^{\mathcal{I}}$ for some $\exists R.C_1 \in \mathsf{sub}(C)$, then $[i_0, \ldots, i_m, i_n]$ is also a path in $\mathcal{I}$.

Note that a path *really* defines "which way to go" through the model, and the notion of "path" is therefore well-defined. However, this would not be the case without assumed role disjointness. For example, consider a model of $(\exists R.\top) \sqcap (\exists S.\top)$ with $\Delta^{\mathcal{I}} = \{i_0, i_1\}$, $R^{\mathcal{I}} = \{\langle x_0, x_1 \rangle\}$, $S^{\mathcal{I}} = \{\langle x_0, x_1 \rangle\}$. In this case, the path $[x_0, x_1]$ would be ambitious, since it is unclear whether it has been produced traversing $\langle x_0, x_1 \rangle \in R^{\mathcal{I}}$ or $\langle x_0, x_1 \rangle \in S^{\mathcal{I}}$. But due to the assumed role disjointness, such ambitious situations can not appear. This can be shown easily using induction over the length of the paths.

Let $\mathcal{P}(\mathcal{I})$ denote the set of paths (as defined above) in $\mathcal{I}$. We can now define $\mathcal{T} = (\mathcal{N}, \mathcal{L}_{\mathcal{N}}, \mathcal{E}, \mathcal{L}_{\mathcal{E}})$ inductively on the length of paths (e.g., in a breadth-first way) as follows:

- $\mathcal{N} =_{def} \mathcal{P}(\mathcal{I})$,
- $\mathcal{E} =_{def} \{ \langle p, q \rangle \mid p, q \in \mathcal{N}, p = [i_0, \ldots, i_n] \text{ (possibly } p = [i_0]), q = [i_0, \ldots, i_n, i_{n+1}], \langle i_n, i_{n+1} \rangle \in \mathcal{UR}(\mathcal{I}, \mathsf{roles}(\mathfrak{R})) \}$,

- $\mathcal{L}_\mathcal{E} =_{def} \{ (<p, q>, R) \mid <p, q> \in \mathcal{E},$
  $p = [i_0, \ldots, i_n]$ (possibly $p = [i_0]$),
  $q = [i_0, \ldots, i_n, i_{n+1}],$
  $<i_n, i_{n+1}> \in R^\mathcal{I} \},$

- For all $q \in \mathcal{N}$, $q = [i_0, \ldots, i_n]$:
  $\mathcal{L}_\mathcal{N}(q) =_{def} \{ C_1 \mid C_1 \in \mathsf{sub}(C), i_n \in C_1^\mathcal{I} \} \cup \{\forall R_?.\top\} \cup$
  $\{ (\forall R.C_1)_{S,w} \mid p = \mathsf{w\_ancestor}(q, w),$
  $p = [i_0, \ldots, i_m]$ (possibly $p = [i_0]$),
  $q = [i_0, \ldots, i_m, \ldots, i_n],$
  $<i_m, i_n> \in S^\mathcal{I}, S \in \mathsf{roles}(\mathfrak{R}),$
  $\forall R.C_1 \in \mathcal{L}_\mathcal{N}(p) \}.$

We have to prove that $\mathcal{T}$ is a tableau for $(C, \mathfrak{R})$ by showing that the tableau conditions are satisfied.

First of all, $(\mathcal{N}, \mathcal{E})$ is indeed a possibly infinite tree.

- Condition 1 (there is some node $x_0 \in \mathcal{N}$ with $C \in \mathcal{L}_\mathcal{N}(x_0)$) is satisfied for $x_0 = [i_0]$, since $[i_0] \in \mathcal{N}$, $i_0 \in C^\mathcal{I}$, and therefore, due to the definition of $\mathcal{L}_\mathcal{N}$, $C \in \mathcal{L}_\mathcal{N}([i_0])$.

- Condition 2a (if $C_1 \in \mathcal{L}_\mathcal{N}(x)$, then $\neg C_1 \notin \mathcal{L}_\mathcal{N}(x)$) is obviously satisfied: for all $p \in \mathcal{N}$ with $p = [i_0, \ldots, i_m]$, $C_1 \in \mathcal{L}_\mathcal{N}(p)$, $\neg C_1 \in \mathcal{L}_\mathcal{N}(p)$ iff $i_m \in C_1^\mathcal{I}$ and $i_m \in (\neg C_1)^\mathcal{I}$. However, then $\mathcal{I}$ would not be a model. Contradiction.

- Condition 2b (if $C_1 \sqcap C_2 \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$ and $C_2 \in \mathcal{L}_\mathcal{N}(x)$) is obviously satisfied: let $p \in \mathcal{N}$, $p = [i_0, \ldots, i_m]$. Then, $C_1 \sqcap C_2 \in \mathcal{L}_\mathcal{N}(p)$ iff $i_m \in (C_1 \sqcap C_2)^\mathcal{I}$. Due to the semantics, $i_m \in C_1^\mathcal{I}$ and $i_m \in C_2^\mathcal{I}$. However, then also $C_1 \in \mathcal{L}_\mathcal{N}(p)$ and $C_2 \in \mathcal{L}_\mathcal{N}(p)$.

- Condition 2c (if $C_1 \sqcup C_2 \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$ or $C_2 \in \mathcal{L}_\mathcal{N}(x)$) is obviously satisfied: let $p \in \mathcal{N}$, $p = [i_0, \ldots, i_m]$. Then, $C_1 \sqcup C_2 \in \mathcal{L}_\mathcal{N}(p)$ iff $i_m \in (C_1 \sqcup C_2)^\mathcal{I}$. Due to the semantics, either $i_m \in C_1^\mathcal{I}$ or $i_m \in C_2^\mathcal{I}$ (or both). However, then also $C_1 \in \mathcal{L}_\mathcal{N}(p)$ or $C_2 \in \mathcal{L}_\mathcal{N}(p)$.

- Condition 2d (if $\exists R.C_1 \in \mathcal{L}_\mathcal{N}(x)$, then there is some y such that $<x, y> \in \mathcal{E}_R$ and $C_1 \in \mathcal{L}_\mathcal{N}(y)$) is satisfied: let $p \in \mathcal{N}$, $p = [i_0, \ldots, i_m]$. Then, $\exists R.C_1 \in \mathcal{L}_\mathcal{N}(p)$ iff $i_m \in (\exists R.C_1)^\mathcal{I}$. Due to the semantics, there is some $i_n \in C_1^\mathcal{I}$ and $<i_m, i_n> \in R^\mathcal{I}$. Due to the definition of path in $\mathcal{I}$, there is some path $q \in \mathcal{N}$, with $q = [i_0, \ldots, i_m, i_n]$. Since $<i_m, i_n> \in R^\mathcal{I}$, by the definition of $\mathcal{L}_\mathcal{E}$ we have $<p, q> \in \mathcal{E}_\mathcal{R}$ (resp. $(<p, q>, \mathcal{E}_\mathcal{R}) \in \mathcal{L}_\mathcal{E}$). Obviously, $C_1 \in \mathcal{L}_\mathcal{N}(q)$ because $i_n \in C_1^\mathcal{I}$.

- Condition 2e (if $(\forall R.C_1)_{R,w} \in \mathcal{L}_\mathcal{N}(x)$, then $C_1 \in \mathcal{L}_\mathcal{N}(x)$) is satisfied: due to the definition of $\mathcal{L}_\mathcal{N}$, $(\forall R.C_1)_{R,w} \in \mathcal{L}_\mathcal{N}(q)$ iff $p =$

$\mathsf{w\_ancestor}(q, w)$, $p = [i_0, \ldots, i_m]$ and $q = [i_0, \ldots, i_m, \ldots, i_n]$, $\langle i_m, i_n \rangle \in R^{\mathcal{I}}$, $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$. We have $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$ iff $i_m \in (\forall R.C_1)^{\mathcal{I}}$. Since $\langle i_m, i_n \rangle \in R^{\mathcal{I}}$, also $i_n \in C_1^{\mathcal{I}}$, and therefore, due to the definition of $\mathcal{L}_{\mathcal{N}}$, $C_1 \in \mathcal{L}_{\mathcal{N}}(q)$.

- Condition 2f ($\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x)$) is obviously satisfied.
- Condition 2g ($(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}_{\mathcal{N}}(x)$ iff $S_1 = S_2$ and there is some x with $\langle x, y \rangle \in \mathcal{E}_{S_1}$ and $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(x)$) is satisfied:

  "$\Rightarrow$" If $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}_{\mathcal{N}}(q)$, then, by the definition of $\mathcal{L}_{\mathcal{N}}$, $p = \mathsf{w\_ancestor}(q, S_2)$ and $\langle p, q \rangle \in \mathcal{E}_{S_2}$, $p = [i_0, \ldots, i_m]$, $q = [i_0, \ldots, i_m, i_n]$, $\langle i_m, i_n \rangle \in S_1^{\mathcal{I}}$, $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$. Since $\langle p, q \rangle \in \mathcal{E}$ and $\langle i_m, i_n \rangle \in S_1^{\mathcal{I}}$ we have $\langle p, q \rangle \in \mathcal{E}_{S_1}$, and therefore, $p = \mathsf{w\_ancestor}(q, S_1)$. Since $\mathcal{T}$ is a tree we have $S_1 = S_2$.

  "$\Leftarrow$" If $S_1 = S_2$, $\langle p, q \rangle \in \mathcal{E}_{S_1}$ and $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$, then $\langle p, q \rangle \in \mathcal{E}_{S_1}$ due to $\langle i_m, i_n \rangle \in S_1^{\mathcal{I}}$. By definition of $\mathcal{L}_{\mathcal{N}}$ we have $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}_{\mathcal{N}}(q)$, and hence $(\forall R.C_1)_{S_1,S_1} \in \mathcal{L}_{\mathcal{N}}(q)$ since $S_1 = S_2$.

- Condition 2h ($(\forall R.C_1)_{S_3,wS_2} \in \mathcal{L}_{\mathcal{N}}(y)$ with $|w| \geq 1$ iff there is some x with $\langle x, y \rangle \in \mathcal{E}_{S_2}$, $(\forall R.C_1)_{S_1,w} \in \mathcal{L}_{\mathcal{N}}(x)$, and $S_3 = \mathsf{con}(S_1, S_2)$) is satisfied:

  "$\Leftarrow$" If there is some p with $\langle p, q \rangle \in \mathcal{E}_{S_2}$, $(\forall R.C_1)_{S_1,w} \in \mathcal{L}_{\mathcal{N}}(p)$, and $S_3 = \mathsf{con}(S_1, S_2)$, then it is obviously the case that $|w| \geq 1$. Additionally, because of $(\forall R.C_1)_{S_1,w} \in \mathcal{L}_{\mathcal{N}}(p)$, there must be some $o = \mathsf{w\_ancestor}(p, w)$, with $o = [i_0, \ldots, i_l]$, $p = [i_0, \ldots, i_l, \ldots, i_m]$ such that $\langle i_l, i_m \rangle \in S_1^{\mathcal{I}}$, $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(o)$. Since $\langle p, q \rangle \in \mathcal{E}_{S_2}$ it holds that $q = [i_0, \ldots, i_l, \ldots, i_m, i_n]$ with $\langle i_m, i_n \rangle \in S_2^{\mathcal{I}}$. Because $S_3 = \mathsf{con}(S_1, S_2)$ and $\mathcal{I} \models \mathfrak{R}$ we also have $\langle i_l, i_n \rangle \in S_3^{\mathcal{I}}$, due to $S_1^{\mathcal{I}} \circ S_2^{\mathcal{I}} \subseteq S_3^{\mathcal{I}}$. But then, by the definition of $\mathcal{L}_{\mathcal{N}}$ we must also have $(\forall R.C_1)_{S_3,wS_2} \in \mathcal{L}_{\mathcal{N}}(q)$, since $o = \mathsf{w\_ancestor}(p, wS_2)$ and $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(o)$.

  "$\Rightarrow$" If $(\forall R.C_1)_{S_3,wS_2} \in \mathcal{L}_{\mathcal{N}}(r)$, and $|w| \geq 1$, then, by definition of $\mathcal{L}_{\mathcal{N}}$, $p = \mathsf{w\_ancestor}(r, wS_2)$, $p = [i_0, \ldots, i_l]$, $r = [i_0, \ldots, i_l, \ldots, i_m, i_n]$, $\langle i_l, i_n \rangle \in S_3^{\mathcal{I}}$, $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$. Obviously, there must also be some $q = [i_0, \ldots, i_l, \ldots, i_m]$, and therefore, $\langle i_l, i_m \rangle \in S_1^{\mathcal{I}}$, for some S_1. Since we know that $\langle p, q \rangle \in \mathcal{E}_{S_2}$ it must be the case that $\langle i_m, i_n \rangle \in S_2^{\mathcal{I}}$. Since $\mathcal{I} \models \mathfrak{R}$ and $\mathfrak{R}$ is a complete role box, and $\langle i_l, i_n \rangle \in S_3^{\mathcal{I}}$, it holds that $S_1^{\mathcal{I}} \circ S_2^{\mathcal{I}} \subseteq S_3^{\mathcal{I}}$, which is enforced by $S_3 = \mathsf{con}(S_1, S_2)$. Because $\langle i_l, i_m \rangle \in S_1^{\mathcal{I}}$, $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(p)$ and $p = \mathsf{w\_ancestor}(q, w)$, we have $(\forall R.C_1)_{S_1,w} \in \mathcal{L}_{\mathcal{N}}(q)$ by the definition of $\mathcal{L}_{\mathcal{N}}$.

Summing up we have shown that $\mathcal{T}$ is a tableau for $(C, \mathfrak{R})$. $\square$

$\sqcap$-rule:

if 1. $C_1 \sqcap C_2 \in \mathcal{L}_\mathcal{N}(x_i)$
2. $\{C_1, C_2\} \not\subseteq \mathcal{L}_\mathcal{N}(x_i)$

then
$\mathcal{L}_\mathcal{N}(x_i) := \mathcal{L}_\mathcal{N}(x_i) \cup \{C_1, C_2\}$

$\sqcup$-rule:

if
1. $C_1 \sqcup C_2 \in \mathcal{L}_\mathcal{N}(x_i)$
2. $\{C_1, C_2\} \cap \mathcal{L}_\mathcal{N}(x_i) = \emptyset$

then
$\mathcal{L}_\mathcal{N}(x_i) := \mathcal{L}_\mathcal{N}(x_i) \cup \{D\}$
for some $D \in \{C_1, C_2\}$

$\forall$-rule:

if 1. $(\forall R.D)_{R,w} \in \mathcal{L}_\mathcal{N}(x_i)$
2. $D \notin \mathcal{L}_\mathcal{N}(x_i)$

then
$\mathcal{L}_\mathcal{N}(x_i) := \mathcal{L}_\mathcal{N}(x_i) \cup \{D\}$

$\exists\forall$-rule:

if 1. $\exists R.C_1 \in \mathcal{L}_\mathcal{N}(x_i)$
2. neither the $\sqcap$- nor the $\sqcup$- nor the $\forall$-rule is applicable to x_i
3. $\neg\exists \langle x_i, x_j \rangle \in \mathcal{E}_\mathcal{R} : C_1 \in \mathcal{L}_\mathcal{N}(x_j)$
4. x_i is not **blocked**

then
create a new node x_j with
$\mathcal{L}_\mathcal{E}(\langle x_i, x_j \rangle) := R$,
$\mathcal{L}_\mathcal{N}(x_j) := \{C_1, \forall R_?.\top\} \cup \{ (\forall T.D)_{R,R} \mid \forall T.D \in \mathcal{L}_\mathcal{N}(x_i) \} \cup \{ (\forall T.D)_{\mathsf{con}(S,R),wR} \mid (\forall T.D)_{S,w} \in \mathcal{L}_\mathcal{N}(x_i) \}$

Figure 1: The tableaux expansion rules

4.2 The Calculus

We are now ready to discuss the algorithm, which works on so-called *completion trees*:

Definition 11 (Completion Tree) A *completion tree* $\mathcal{CT}$ for $(C, \mathfrak{R})$ is a *finite* concept tree. A completion tree is said to contain a *clash* iff there is some node x with $\{C, \neg C\} \subseteq \mathcal{L}_\mathcal{N}(x)$ for some $C \in \mathcal{N}_\mathcal{C}$. A node $y \in \mathcal{N}$ is said to be *blocked* iff there is an ancestor node x of y with with $x \equiv y$. In this case, we say y is *blocked by* x. The relation $\equiv$ is defined as follows: $x \equiv y$ iff $\forall c_x : (c_x \in \mathcal{L}_\mathcal{N}(x) \Rightarrow \exists c_y \in \mathcal{L}_\mathcal{N}(y) : c_x \equiv c_y) \wedge \forall c_y : (c_y \in \mathcal{L}_\mathcal{N}(y) \Rightarrow \exists c_x \in \mathcal{L}_\mathcal{N}(x) : c_x \equiv c_y)$, where $c_x \equiv c_y$ iff $c_x = (\forall R.C)_{S,w} \wedge c_y = (\forall R.C)_{S,v} \vee c_x = c_y$.[7]

The tableaux algorithm works as follows: in order to decide the satisfiability of $(C, \mathfrak{R})$, the algorithm starts with the initial completion tree

$$\mathcal{CT}_0 = (\{x_0\}, \{(x_0, \{C, \forall R_?.\top\})\}, \emptyset, \emptyset)$$

and exhaustively applies the non-deterministic *tableaux expansion rules* (see Figure 1) until either the completion tree contains a clash or none of the rules is

[7] In fact, instead of the used *equal* blocking, also *subset* blocking would work for $\mathcal{ALC}_{\mathcal{RASG}}$; however, this has no impact on the worst-case complexity of the algorithm.

applicable anymore, i.e. the tree is *complete.*

The rules either expand the tree by generating new successor nodes due to the presence of ∃-constraints, or expand the label of some node x by adding constraints to it, according to the decomposition into subconcepts. If the completion rules can be applied in such a way that they construct a complete and clash-free completion tree, $(C, \mathfrak{R})$ is satisfiable, otherwise $(C, \mathfrak{R})$ is unsatisfiable. Due to the non-determinism this is the case if *all* possible computations yield a completion tree containing a clash. The algorithm therefore behaves like a non-deterministic Turing machine: SAT returns TRUE iff there is *some* successful computation. More formally, SAT can be specified as follows (note that apply_rule is a non-deterministic procedure):

$$
\begin{array}{l}
\mathsf{SAT}(C, \mathfrak{R}) =_{def} \\
\quad \mathcal{CT}_0 := (\{x_0\}, \{(x_0, \{C, \forall R_?.\top\})\}, \emptyset, \emptyset) \\
\quad i := 0 \\
\quad \texttt{WHILE } (\neg\mathsf{complete?}(\mathcal{CT}_i) \wedge \neg\mathsf{clash?}(\mathcal{CT}_i)) \\
\quad \texttt{DO} \\
\quad\quad \mathcal{CT}_{i+1} := \mathsf{apply\_rule}(\mathcal{CT}_i) \\
\quad\quad i := i + 1 \\
\quad \texttt{OD} \\
\quad \texttt{RETURN } \neg\mathsf{clash?}(\mathcal{CT}_i)
\end{array}
$$

A *blocking mechanism* is needed to ensure the termination of the tableaux algorithm, e.g. for $((\exists R.C) \sqcap (\forall R.\exists R.C), \{R \circ R \sqsubseteq R\})$. Note that this concept is also expressible in $\mathcal{ALC}_{\mathcal{R}^+}$, since R is declared as a transitively closed role. Unlike for $\mathcal{ALC}_{\mathcal{R}^+}$, an infinite model (tableau) must be constructed if blocking occurred. This is due to the fact that $\mathcal{ALC}_{\mathcal{RASG}}$ does not have the finite model property (see example above).

4.3 Formal Properties of the Calculus

In the following we will prove that the algorithm always terminates, and that it is sound and complete – the algorithm is therefore a decision procedure. Additionally, we prove EXPTIME-completeness of $(C, \mathfrak{R})$-satisfiability in $\mathcal{ALC}_{\mathcal{RASG}}$.

4.3.1 Termination

Lemma 2 (Termination) For each pair $(C, \mathfrak{R})$, where C is an $\mathcal{ALC}_{\mathcal{RASG}}$ concept and $\mathfrak{R}$ is a role box that is admissible for C, $\mathsf{SAT}((C, \mathfrak{R}))$ terminates.

Proof 3 We have to show that there is no infinite sequence of rule applications. To obtain an infinite sequence of rule applications, at least one rule must be applied infinitely often, since we have a finite number of rules. We show that no rule can be applied infinitely often.

Inspecting the tableaux expansion rules, it is obviously the case that for any node x in the tree, each rule can be applied at most once. This is due to the fact that the applicability of each rule is appropriately "guarded" by a precondition. For example, the $\sqcap$-rule can only be applied to a node x if one of the conjuncts it is going to add is missing in the label $\mathcal{L}_{\mathcal{N}}(x)$ of the node x. Additionally one has to observe that none of the rules ever removes elements from $\mathcal{L}(x)$ what could re-trigger the applicability of the rules $\sqcap$, $\sqcup$ and $\forall$. Also, none of the rules ever removes already created (successor) nodes, what could re-trigger the applicability of the $\exists\forall$-rule.

Then, in order to obtain an infinite number of rule applications, the algorithm must create an infinite number of nodes and therefore an infinite completion tree. Exploiting the fact that new nodes are only generated by the $\exists\forall$-rule it is obvious that we have a finitely branching tree whose branching factor is bounded by $|\mathsf{sub}(C)|$, since there cannot be more than $|\mathsf{sub}(C)|$ concepts of the form $\exists R.C$ in the label of each node. Then, due to König's Lemma, in order to obtain an infinite tree, there must be an infinite path, since the tree is finitely branching. We show that there is no infinite path:

With respect to blocking, it suffices to view the nodes as if they were annotated with labels from

$$\mathcal{Q} = \mathsf{sub}(C) \cup \{\forall R_?.\top\} \cup \{\, (\forall R.C_1)_S \mid \forall R.C_1 \in \mathsf{sub}(C), S \in \mathsf{roles}(\mathfrak{R})\,\},$$

since $(\forall R.C_1)_{S,w} \equiv (\forall R.C_1)_{S,v}$, even if $w \neq v$. Obviously, there are at most $n = 2^{|\mathcal{Q}|}$ different node labels. In terms of complexity, if $x = |C|$ and $y = |\mathfrak{R}|$, please note that $|\mathsf{sub}(C)|$ is bound by x and $|\mathsf{roles}(\mathfrak{R})|$ is bound by $\sqrt{\frac{y}{3}}$ (e.g. a minimal representation of a composition table would be set of triples; note that the composition table is complete). Then, $n \leq (2^{x+x*\sqrt{\frac{y}{3}}}) \leq 2^{cx}$ for an appropriate c. Considering the size of the input $z = x + y$ as input parameter ("combined complexity"), $n \leq (2^{x+x*\sqrt{y}}) \leq 2^{z^2}$. Any path of a length greater than n must therefore contain two nodes x, y with $x \equiv y$, and blocking occurs (Pigeonhole Principle). Since no rule is applicable to a blocked node, there can be no infinite path(s). Summing up we have shown that SAT always terminates. $\square$

4.3.2 Soundness

We will prove soundness by showing how to construct a tableau from a complete and clash-free completion tree that has been derived by SAT. In some cases, if blocking occurred during the construction of the completion tree, the constructed tableau will be infinite. Again, an unraveling technique is used for this purpose.

Lemma 3 (Soundness) Let C be an $\mathcal{ALC}_{\mathcal{RASG}}$ concept, and $\mathfrak{R}$ be an admissible role box for C. If there is a computation such that $\mathsf{SAT}((C, \mathfrak{R}))$ returns TRUE, then $(C, \mathfrak{R})$ has a tableau.

Proof 4 Intuitively, the definition of a tableau $\mathcal{T} = (\mathcal{N}', \mathcal{L}'_{\mathcal{N}}, \mathcal{E}', \mathcal{L}'_{\mathcal{E}})$ from the complete and clash-free completion tree $\mathcal{CT}_i = (\mathcal{N}, \mathcal{L}_{\mathcal{N}}, \mathcal{E}, \mathcal{L}_{\mathcal{E}})$ that has been constructed by a successful computation of SAT works as follows (the following explanation is borrowed from Horrocks, Sattler et. al in [7, 8]): An individual in $\mathcal{N}'$ corresponds to a *path* in the completion tree from the root node to some node that is not blocked. To obtain infinite tableaus, these paths may by *cyclic.* Instead of going to a blocked node, these paths go "back" to the blocking node, and this for an arbitrary number of times. Thus, if blocking occurred while constructing a tableau, we obtain an infinite tableau. Let $\mathcal{CT}_i = (\mathcal{N}, \mathcal{L}_{\mathcal{N}}, \mathcal{E}, \mathcal{L}_{\mathcal{E}})$ be the complete and clash-free completion tree.

Paths in $\mathcal{CT}_i$ are defined inductively as follows (we simply write $\mathcal{CT}$ in the following):

1. For the root node $x_0 \in \mathcal{N}$, $[x_0]$ is a path in $\mathcal{CT}$.
2. For a path $[x_0, \ldots, x_n]$ and a node $x_i \in \mathcal{N}$, $[x_0, \ldots, x_n, x_i]$ is a path in $\mathcal{CT}$ iff
   (a) $<x_n, x_i> \in \mathcal{E}$, and x_i is not blocked, or
   (b) $<x_n, x_{n+1}> \in \mathcal{E}$, and x_{n+1} is blocked by x_i $(x_i \equiv x_{n+1})$.

Whenever we write $[x_0, \ldots, x_n]$, we also include the possibility that $[x_0, \ldots, x_n] = [x_0]$; and $[x_0, \ldots, x_n, x_{n+1}]$ includes the possibility that $[x_0, \ldots, x_n, x_{n+1}] = [x_0, x_1]$.

First of all, a path is a finite object. Additionally, a path $p = [x_0, \ldots, x_{pt}]$ contains no blocked nodes, but may contain nodes that act as blocking nodes for nodes that are blocked in the completion tree. A path may contain more that one node acting as a blocking node (for *different* blocked nodes).

As already noted, an individual $path_q \in \mathcal{N}'$ corresponds to a *path* q. Given a path q we denote the last individual from $\mathcal{N}$ lying on this path with x_{qt} (for $q's$ tail): $q = [x_0, \ldots, x_{qt}]$. The corresponding individual from $\mathcal{N}'$ is $path_q = path_{[x_0, \ldots, x_{qt}]}$. The tableau $\mathcal{T} = (\mathcal{N}', \mathcal{L}'_{\mathcal{N}}, \mathcal{E}', \mathcal{L}'_{\mathcal{E}})$ for $(C, \mathfrak{R})$ is defined as follows:

- $\mathcal{N}' =_{def} \{\, path_p \mid p \text{ is a path in } \mathcal{CT} \,\}$

- $\mathcal{E}' =_{def} \{ <path_p, path_q> \mid p = [x_0, \ldots, x_{pt}],$
  $q = [x_0, \ldots, x_{pt}, x_{qt}],$
  $<x_{pt}, x_{qt}> \in \mathcal{E}, x_{qt}$ is not blocked, or
  $<x_{pt}, x_i> \in \mathcal{E}, x_i$ is blocked by x_{qt} $(x_{qt} \equiv x_i)$ $\}$

- $\mathcal{L}'_{\mathcal{E}} =_{def} \{ (<path_p, path_q>, R) \mid p = [x_0, \ldots, x_{pt}],$
  $q = [x_0, \ldots, x_{pt}, x_{qt}],$
  $<x_{pt}, x_{qt}> \in \mathcal{E}_{\mathcal{R}}, x_{qt}$ is not blocked, or
  $<x_{pt}, x_i> \in \mathcal{E}_R, x_i$ is blocked by x_{qt} $(x_{qt} \equiv x_i)$ $\}$

For all $path_q \in \mathcal{N}'$, $\mathcal{L}'_{\mathcal{N}}(path_q)$ is defined inductively on the length of the denoted paths, starting with $path_{[x_0]}$ (e.g. by using a breadth-first traversation).

- $\mathcal{L}'_{\mathcal{N}}(path_{[x_0]}) =_{def} \mathcal{L}_{\mathcal{N}}(x_0)$

If $\mathcal{L}'_{\mathcal{N}}(path_p)$ with $p = [x_0, \ldots, x_{pt}]$ is already defined, and $<path_p, path_q> \in \mathcal{E}'_{\mathcal{R}}$ for some role R, we then define $\mathcal{L}'_{\mathcal{N}}(path_q)$ for $q = [x_0, \ldots, x_{pt}, x_{qt}]$ as follows:

- $\mathcal{L}'_{\mathcal{N}}(path_q) =_{def}$ $\{\forall R_?.\top\} \cup$
  $\mathcal{L}_{\mathcal{N}}(x_{qt}) \cap \mathsf{sub}(C) \cup$
  $\{ (\forall T.D)_{R,R} \mid \forall T.D \in \mathcal{L}_{\mathcal{N}}(x_{pt}) \} \cup$
  $\{ (\forall T.D)_{\mathsf{con}(S,R),wR} \mid (\forall T.D)_{S,w} \in \mathcal{L}'_{\mathcal{N}}(path_p) \}$

Lemma 4 Let $\mathcal{CT} = (\mathcal{N}, \mathcal{L}_{\mathcal{N}}, \mathcal{E}, \mathcal{L}_{\mathcal{E}})$ be the clash-free and complete completion tree constructed by SAT, and let $\mathcal{T} = (\mathcal{N}', \mathcal{L}'_{\mathcal{N}}, \mathcal{E}', \mathcal{L}'_{\mathcal{E}})$ be defined as above. Then, for all $path_p \in \mathcal{N}'$: $path_p \equiv x_{pt}$.

Proof 5 This can be easily shown using induction, starting with $path_{[x_0]}$. The key-observation is that an admissible role box must be deterministic. □

Now it can be shown that $\mathcal{T} = (\mathcal{N}', \mathcal{L}'_{\mathcal{N}}, \mathcal{E}', \mathcal{L}'_{\mathcal{E}})$ is a tableau for $(C, \mathfrak{R})$.

1. First of all, the tableau is a possibly infinite concept tree.

2. Condition 1 is satisfied, since $C \in \mathcal{L}_{\mathcal{N}}(x_0)$ for the root node x_0 of the completion tree $\mathcal{CT}$. Since the root node cannot be blocked there is an individual $path_{[x_0]} \in \mathcal{N}'$. Due to Lemma 4, $C \in \mathcal{L}'_{\mathcal{N}}(path_{[x_0]})$.

3. Condition 2f is satisfied. $\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x_0)$ since the SAT algorithm started with $\mathcal{CT}_0 = (\{x_0\}, \{(x_0, \{C, \forall R_?.\}\top\}), \emptyset, \emptyset)$, and for the other nodes, $\forall R_?.\top \in \mathcal{L}_{\mathcal{N}}(x_i)$ due to the $\exists\forall$-rule. Due to Lemma 4, $\forall R_?.\top \in \mathcal{L}'_{\mathcal{N}}(path_p)$ for all $path_p \in \mathcal{N}'$.

4. Due to Lemma 4, Conditions 2a, 2b, 2c and 2e are trivially satisfied, since $path_p \equiv x_{pt}$, and $x_{pt} \in \mathcal{N}$ is a node in a clash-free and *complete* completion tree. For example, consider Property 2b: if $C_1 \sqcap C_2 \in \mathcal{L}'_{\mathcal{N}}(path_p)$, then $C_1 \in \mathcal{L}'_{\mathcal{N}}(path_p)$, $C_2 \in \mathcal{L}'_{\mathcal{N}}(path_p)$. Because the completion tree is complete, the $\sqcap$-rule is not applicable to x_{pt}. This shows that $C_1 \in \mathcal{L}_{\mathcal{N}}(x_{pt})$ and $C_2 \in \mathcal{L}_{\mathcal{N}}(x_{pt})$. Since $path_p \equiv x_{pt}$ the same holds for $\mathcal{L}'_{\mathcal{N}}(path_p)$.

5. Condition 2d (if $\exists R.C_1 \in \mathcal{L}'_{\mathcal{N}}(path_p)$, then there is some $path_q \in \mathcal{N}'$ such that $<path_p, path_q> \in \mathcal{E}'_{\mathcal{R}}$ and $C_1 \in \mathcal{L}'_{\mathcal{N}}(path_q)$) is satisfied. Since $\exists R.C_1 \in \mathcal{L}'_{\mathcal{N}}(path_p)$, it must be the case that $\exists R.C_1 \in \mathcal{L}_{\mathcal{N}}(x_{pt})$. Note that x_{pt} is not blocked, by the definition of "path in $\mathcal{CT}$". Therefore, the $\exists\forall$-rule has created an R successor x_i, with $<x_{pt}, x_i> \in \mathcal{E}_{\mathcal{R}}$ and $C_1 \in \mathcal{L}_{\mathcal{N}}(x_i)$. There are two possibilities:

   (a) x_i is not blocked. In this case, due to the definition of $\mathcal{E}'_{\mathcal{R}}$, we have $x_{qt} = x_i$, and $<path_p, path_q> \in \mathcal{E}'_{\mathcal{R}}$. Therefore, $C_1 \in \mathcal{L}'_{\mathcal{N}}(path_q)$, since $C_1 \in \mathcal{L}_{\mathcal{N}}(x_{qt})$.

   (b) x_i is blocked. In this case, instead of going to the R successor x_i in the completion tree, the path q is obtained from p by "going back" to the blocking node x_{qt}, $x_{qt} \equiv x_i$. However, by the definition of $\mathcal{E}'_{\mathcal{R}}$, we have again $<path_p, path_q> \in \mathcal{E}'_{\mathcal{R}}$. By definition of $\mathcal{L}'_{\mathcal{N}}$ and the blocking condition it holds that $C_1 \in \mathcal{L}'_{\mathcal{N}}(path_q)$, since $C_1 \in \mathcal{L}_{\mathcal{N}}(x_{qt})$.

6. Condition 2g (if $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}'_{\mathcal{N}}(path_q)$ iff $S_1 = S_2$ and there is some $path_p$ with $<path_p, path_q> \in \mathcal{E}'_{S1}$ and $(\forall R.C_1) \in \mathcal{L}'_{\mathcal{N}}(path_p)$) is satisfied:

   "$\Rightarrow$" If $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}'_{\mathcal{N}}(path_q)$, then by definition of $\mathcal{L}'_{\mathcal{N}}$, $<path_p, path_q> \in \mathcal{E}'_{S_1}$, $S_1 = S_2$ and $\forall R.C_1 \in \mathcal{L}_{\mathcal{N}}(x_{pt})$. Because of $path_p \equiv x_{pt}$ we also have $(\forall R.C_1) \in \mathcal{L}'_{\mathcal{N}}(path_p)$.

   "$\Leftarrow$" If there is some $path_p$ with $<path_p, path_q> \in \mathcal{E}'_{S_1}$ and $(\forall R.C_1) \in \mathcal{L}'_{\mathcal{N}}(path_p)$, and $S_1 = S_2$, then also $(\forall R.C_1)_{S_1,S_2} \in \mathcal{L}'_{\mathcal{N}}(path_q)$, because $(\forall R.C_1) \in \mathcal{L}_{\mathcal{N}}(x_{pt})$, and by definition of $\mathcal{L}'_{\mathcal{N}}$ we have $(\forall R.C_1)_{S_1,S_1} \in \mathcal{L}'_{\mathcal{N}}(path_q)$.

7. Condition 2h (if $(\forall R.C_1)_{S_3,wS_2} \in \mathcal{L}'_{\mathcal{N}}(path_q)$ with $|w| \geq 1$ iff there is some $path_p$ with $<path_p, path_q> \in \mathcal{E}'_{S2}$, $(\forall R.C_1)_{S_1,w} \in \mathcal{L}'_{\mathcal{N}}(path_p)$, and $S_3 = \mathsf{con}(S_1, S_2)$) is satisfied by definition of $\mathcal{L}'_{\mathcal{N}}$, which is obvious. □

4.3.3 Completeness

We show how to construct a complete and clash-free completion tree $\mathcal{CT}_i$ from a given tableau $\mathcal{T}$. Intuitively, instead of doing a "blind search", the application of the tableau rules is now guided by the information in the tableau $\mathcal{T}$. For example, when the $\sqcup$-rule is to be applied to a node x in the completion tree $\mathcal{CT}$ and has the choice whether to add, for example, C_1 or C_2 to the label of this node, the *node label of the corresponding node in the tableau* $\mathcal{T}$ is used to decide whether C_1 or C_2 should be added. Note that we cannot make wrong decisions, since otherwise $\mathcal{T}$ would not be a tableau. However, the question is how to find the corresponding node in $\mathcal{T}$ containing the required information.

For this purpose, a *mapping function* $\pi \; : \; \mathcal{N}' \rightarrow \mathcal{N}$ is used to associate nodes in the completion tree ($\mathcal{N}'$) with nodes in the tableau ($\mathcal{N}$). Now, the non-deterministic $\sqcup$-rule is always applied in a way such that for all $x' \in \mathcal{N}'$: $\mathcal{L}'_{\mathcal{N}}(x') \subseteq \mathcal{L}_{\mathcal{N}}(\pi(x'))$ (note that $x = \pi(x')$, $x \in \mathcal{N}$). To achieve this, the $\sqcup$-rule is replaced by the $\sqcup'$-rule, which "looks up" the required information using π. π is constructed incrementally by the modified $\exists\forall'$ rule, which replaces the $\exists\forall$-rule. These modified rules are shown in Figure 2. Using this technique, we can show *completeness* of the calculus (resp. of the procedure SAT):

$\sqcup'$-rule:
if 1. $C_1 \sqcup C_2 \in \mathcal{L}'_{\mathcal{N}}(x'_i)$
2. $\{C_1, C_2\} \cap \mathcal{L}'_{\mathcal{N}}(x'_i) = \emptyset$
then $\mathcal{L}'_{\mathcal{N}}(x'_i) := \mathcal{L}'_{\mathcal{N}}(x'_i) \cup \{D\}$
for some $D \in \{C_1, C_2\} \cap \mathcal{L}_{\mathcal{N}}(\pi(x'_i))$

$\exists\forall'$-rule:
if 1. $\exists R.C_1 \in \mathcal{L}'_{\mathcal{N}}(x'_i)$
2. neither the $\sqcap$- nor the $\sqcup'$- nor the $\forall$-rule is applicable to x'_i
3. $\neg\exists{<}x'_i, x'_j{>} \in \mathcal{E}'_{\mathcal{R}} : C_1 \in \mathcal{L}'_{\mathcal{N}}(x'_j)$
4. x'_i is not **blocked**
then create a new node x'_j with
$\pi(x'_j) =_{def} x_j$ for
some ${<}\pi(x'_i), x_j{>} \in \mathcal{E}_R$ with $C_1 \in \mathcal{L}_{\mathcal{N}}(x_j)$ and let
$\mathcal{L}'_{\mathcal{E}}({<}x'_i, x'_j{>}) := R$,
$\mathcal{L}'_{\mathcal{N}}(x'_j) := \{C_1, \forall R_?.\top\} \cup$
$\{\, (\forall T.D)_{R,R} \mid \forall T.D \in \mathcal{L}'_{\mathcal{N}}(x'_i) \,\} \cup$
$\{\, (\forall T.D)_{\mathsf{con}(S,R),wR} \mid (\forall T.D)_{S,w} \in \mathcal{L}'_{\mathcal{N}}(x'_i) \,\}$

Figure 2: The modified tableau expansion rules for $\mathcal{ALC}_{\mathcal{RASG}}$

Lemma 5 (Completeness) Let C be an $\mathcal{ALC}_{\mathcal{RASG}}$ concept, and $\mathfrak{R}$ be an admissible role box for C, and let $\mathcal{T}$ be a tableau for $(C, \mathfrak{R})$. Then, $\mathsf{SAT}(C, \mathfrak{R})$

returns TRUE – there is a successful computation yielding a complete and clash-free completion tree $\mathcal{CT}_i$.

Proof 6 If there is a successful computation $\mathsf{SAT}(C, \mathfrak{R})$, then the tableaux expansion rules can be applied in such a way to

$$\mathcal{CT}_0 = (\{x_0\}, \{(x_0, \{C, \forall R_?.\top\})\}, \emptyset, \emptyset)$$

that a complete and clash-free completion tree $\mathcal{CT}_i$ is created. We show how to construct $\mathcal{CT}_i$ using the information in the tableaux $\mathcal{T} = (\mathcal{N}, \mathcal{L}_\mathcal{N}, \mathcal{E}, \mathcal{L}_\mathcal{E})$. We will use the *modified tableaux expansion rules* $\{\sqcap, \sqcup', \forall, \exists\forall'\}$. It is easy to see that, if the completion tree $\mathcal{CT}_i$ was generated using the *modified* expansion rules, then $\mathcal{CT}_i$ could have also been constructed using the (original) expansion rules in the first place. A successful computation of SAT using the *modified* expansion rules implies the existence of a successful computation of SAT using the *original* expansion rules, therefore proving the Lemma.

We still need to argue formally that the completion tree that has been constructed with the modified expansion rules is clash free. Obviously, the tableau is "clash-free" – there is no node x with $C_1, \neg C_1 \subseteq \mathcal{L}_\mathcal{N}(x)$ for some $C_1 \in \mathcal{N}_\mathcal{C}$. We therefore show $\mathcal{L}'_\mathcal{N}(x') \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$ for all nodes $x' \in \mathcal{N}'$, using induction over the *sequence of completion trees* that is generated during the construction of $\mathcal{CT}_i$ and π:

- We start with $\mathcal{CT}_0 = (\{x'_0\}, \{(x'_0, \{C, \forall R_?.\top\})\}, \emptyset, \emptyset)$ and $\pi(x'_0) =_{def} x_0$. Note that $x_0 \in \mathcal{N}$ is the root node of the tableau. Due to Tableaux Condition 2f we have $\mathcal{L}'_\mathcal{N}(x') \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$.
- For the induction step, let $\mathcal{CT}'$ be the completion tree that has been generated by the application of some rule to $\mathcal{CT}$. According to the different rules, we can distinguish four cases:
  - If $\mathcal{CT}'$ has is derived from $\mathcal{CT}$ by an application of the $\sqcap$-rule to some node $x' \in \mathcal{N}'$ with $C_1 \sqcap C_2 \in \mathcal{L}'_\mathcal{N}(x')$, then both C_1 and C_2 are added to $\mathcal{L}'_\mathcal{N}(x')$, yielding $\mathcal{CT}'$. Due to the induction hypothesis $\mathcal{L}'_\mathcal{N}(x') \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$ holds for $\mathcal{CT}$. But since $\mathcal{T}$ is a tableau we already have $\{C_1, C_2\} \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$ and therefore still $\mathcal{L}'_\mathcal{N}(x') \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$ after the rule application.
  - If the $\sqcup'$-rule can be applied to x' in $\mathcal{CT}$ with $C_1 \sqcup C_2 \in \mathcal{L}'_\mathcal{N}(x')$, then also $C_1 \in \mathcal{L}_\mathcal{N}(\pi(x'))$ or $C_2 \in \mathcal{L}_\mathcal{N}(\pi(x'))$ due to the induction hypothesis. In the former case, the $\sqcup'$-rule adds C_1 (and not C_2), and in latter it adds C_2 (and not C_1) to $\mathcal{L}'_\mathcal{N}$. Since the "correct" disjunct is added, we have $\mathcal{L}'_\mathcal{N}(x') \subseteq \mathcal{L}_\mathcal{N}(\pi(x'))$ in $\mathcal{CT}'$.

- If the $\forall$-rule can be applied to x' in $\mathcal{CT}$, then $(\forall R.C)_{R,w} \in \mathcal{L}'_{\mathcal{N}}(x')$. Due to the induction hypothesis we also have $(\forall R.C)_{R,w} \in \mathcal{L}_{\mathcal{N}}(\pi(x'))$, and since $\mathcal{T}$ is a tableau, $C \in \mathcal{L}_{\mathcal{N}}(\pi(x'))$. C is added by the $\forall$-rule to $\mathcal{L}'_{\mathcal{N}}(x')$. Therefore, we still have $\mathcal{L}'_{\mathcal{N}}(x') \subseteq \mathcal{L}_{\mathcal{N}}(\pi(x'))$.
- Assume the $\exists\forall'$-rule can be applied to x' in $\mathcal{CT}$ due to $\exists R.C_1 \in \mathcal{L}'_{\mathcal{N}}(x')$. Due to the induction hypothesis we also have $\exists R.C_1 \in \mathcal{L}_{\mathcal{N}}(\pi(x'))$ and therefore, there is some node $y \in \mathcal{N}$ with $C_1 \in \mathcal{L}_{\mathcal{N}}(y)$ and $<\pi(x'), y> \in \mathcal{E}_R$, due to tableaux Condition 2d. The $\exists\forall'$-rule does not change the label of x', but creates a new node y' with $\mathcal{L}'_{\mathcal{N}}(y') := \{C_1, \forall R_?.\top\} \cup \{\, (\forall T.D)_{R,R} \mid \forall T.D \in \mathcal{L}'_{\mathcal{N}}(x') \,\} \cup \{\, (\forall T.D)_{\mathsf{con}(S,R),wR} \mid (\forall T.D)_{S,w} \in \mathcal{L}'_{\mathcal{N}}(x') \,\}$ and assigns $\pi(y') =_{def} y$ for some appropriate y. Due to the induction hypothesis, $\{\, (\forall T.D)_{S,w} \mid (\forall T.D)_{S,w} \in \mathcal{L}'_{\mathcal{N}}(x') \,\} \subseteq \mathcal{L}_{\mathcal{N}}(x)$, $\{\, \forall T.D \mid \forall T.D \in \mathcal{L}'_{\mathcal{N}}(x) \,\} \subseteq \mathcal{L}_{\mathcal{N}}(x)$. According to the tableaux Conditions 2e, 2d, 2g and 2h, not only $\{C_1, \forall R_?.\top\} \subseteq \mathcal{L}_{\mathcal{N}}(y)$, but also $\mathcal{L}'_{\mathcal{N}}(y') \subseteq \mathcal{L}_{\mathcal{N}}(\pi(y'))$. This is ensured by the fact that the role box and therefore the $\exists\forall'$-rule is deterministic; i.e. $\mathcal{L}'_{\mathcal{N}}(y')$ is solely determined by $\mathcal{L}'_{\mathcal{N}}(x')$, and this holds for $\mathcal{T}$ as well. □

The following theorem summarizes Lemma 2, Lemma 3 and Lemma 5:

Theorem 1 SAT is a decision procedure for $(C, \mathfrak{R})$-satisfiability in $\mathcal{ALC}_{\mathcal{RASG}}$ (provided C is an $\mathcal{ALC}_{\mathcal{RASG}}$ concept and $\mathfrak{R}$ is an admissible role box for C).

4.3.4 Computational Complexity

In the following, we consider the computational complexity of $(C, \mathfrak{R})$-satisfiability in $\mathcal{ALC}_{\mathcal{RASG}}$.

Theorem 2 $(C, \mathfrak{R})$-satisfiability in $\mathcal{ALC}_{\mathcal{RASG}}$ is EXPTIME-complete.

Proof 7 Lower Complexity Bound: EXPTIME-hardness It is well-known that satisfiability of $\mathcal{ALC}$ concepts w.r.t. *general* TBoxes containing GCIs (see above) is EXPTIME-complete (see [13]).

Using a technique called *internalization* (see [1], [6]), we can easily show EXPTIME-hardness of $\mathcal{ALC}_{\mathcal{RASG}}$.

Internalization for $\mathcal{ALC}_{\mathcal{RASG}}$ works as follows: given an arbitrary $\mathcal{ALC}$ concept C and a general TBox $\mathfrak{T}$, we fix some role $S \in \mathcal{N}_{\mathcal{R}}$ that does neither appear in C nor in $\mathfrak{T}$ ($S \notin \mathsf{roles}(C) \cup \bigcup_{D \sqsubseteq E \in \mathfrak{T}} (\mathsf{roles}(D) \cup \mathsf{roles}(E))$). Let $C' =_{def} C \sqcap M_{\mathfrak{T}} \sqcap \forall(\sqcup_{R \in \mathsf{roles}(C \cup \{S\})} R).M_{\mathfrak{T}}$, $M_{\mathfrak{T}} =_{def} \sqcap_{D \sqsubseteq E \in \mathfrak{T}} \neg D \sqcup E$ and $\mathfrak{R} =_{def} \{\, R_1 \circ R_2 \sqsubseteq S \mid R_1, R_2 \in \mathsf{roles}(C') \,\}$. Please note that S is the fixed role. The concept $M_{\mathfrak{T}}$ is the so-called *meta-constraint* corresponding to the TBox $\mathfrak{T}$, and $\mathfrak{R}$ is obviously an admissible role box for C'.

We show the following: the $\mathcal{ALC}$ concept C is satisfiable w.r.t. the TBox $\mathfrak{T}$ iff C' is satisfiable w.r.t. the role box $\mathfrak{R}$.

"⇒" If $\mathcal{I} \models (C, \mathfrak{T})$, then we construct $\mathcal{I}'$ such that $\mathcal{I}' \models (C', \mathfrak{R})$. Without loss of generality we may assume that $\mathcal{I}$ has the form of a (possibly) infinite tree.[8] If $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we then define $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ with $\Delta^{\mathcal{I}'} =_{def} \Delta^{\mathcal{I}}$, and for all concept names $D \in \mathsf{sub}(C)$, $D^{\mathcal{I}'} =_{def} D^{\mathcal{I}}$, and all role names $R \in \mathsf{roles}(C)$, $R^{\mathcal{I}'} =_{def} R^{\mathcal{I}}$. Additionally, we set $S^{\mathcal{I}'} =_{def} (\mathcal{UR}(\mathcal{I}, \mathsf{roles}(C))^+ \setminus \mathcal{SKEL}(\mathcal{I}, \mathsf{roles}(C)))$. Due to $\mathcal{I} \models \mathfrak{T}$, for each GCI $D \sqsubseteq E \in \mathfrak{T}$: $D^{\mathcal{I}} \subseteq E^{\mathcal{I}}$. Obviously, then also $x \in ((\neg D) \sqcup E)^{\mathcal{I}}$ for each $x \in \Delta^{\mathcal{I}}$ and every $D \sqsubseteq E \in \mathfrak{T}$, and therefore, $x \in M_{\mathfrak{T}}^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$. If $x_0 \in C^{\mathcal{I}}$, then also $x_0 \in (C \sqcap M_{\mathfrak{T}})^{\mathcal{I}'}$. Now, for each $x_i \in \Delta^{\mathcal{I}}$ with $x_0 \neq x_i$ either $<x_0, x_i> \in \mathcal{UR}(\mathcal{I}, \mathsf{roles}(C))$ and therefore, $<x_0, x_i> \in \forall(\sqcup_{R \in \mathsf{roles}(C)} R).M_{\mathfrak{T}}^{\mathcal{I}'}$, or $<x_0, x_i> \in S^{\mathcal{I}'}$. Since $x_i \in M_{\mathfrak{T}}^{\mathcal{I}'}$, this shows that $x_0 \in \forall(\sqcup_{R \in \mathsf{roles}(C) \cup \{S\}} R).M_{\mathfrak{T}}^{\mathcal{I}'}$, and therefore, $x_0 \in C'^{\mathcal{I}}$. Due to the definition of $S^{\mathcal{I}'}$ and since our model has the form of a tree, $\mathcal{I}' \models \mathfrak{R}$ and all roles are still interpreted as disjoint. This shows that $\mathcal{I}' \models (C', \mathfrak{R})$.

"⇐" If $\mathcal{I}' \models (C', \mathfrak{R})$, then $\mathcal{I}$ can be constructed such that $\mathcal{I} \models (C, \mathfrak{T})$. Without loss of generality we may assume that $\mathcal{SKEL}(\mathcal{I}', \mathsf{roles}(C'))$ has the form of a possibly infinite tree – $\mathcal{I}'$ is a tree skeleton model. We simply define $\Delta^{\mathcal{I}} =_{def} \Delta^{\mathcal{I}'} \cap M_{\mathfrak{T}}^{\mathcal{I}'}$ and for all concept names $D \in \mathsf{sub}(C)$: $D^{\mathcal{I}} =_{def} D^{\mathcal{I}'}$, and for all role names $R \in \mathsf{roles}(C)$: $R^{\mathcal{I}} =_{def} R^{\mathcal{I}'}$. It is obvious that $\mathcal{I} \models (C, \mathfrak{T})$.

Upper Complexity Bound: Membership in EXPTIME We show that satisfiability of $(C, \mathfrak{R})$ in $\mathcal{ALC}_{\mathcal{RASG}}$ can be reduced to satisfiability of C in $\mathcal{ALC}$ w.r.t. a certain general TBox $\mathfrak{T}$. This TBox $\mathfrak{T}$ is extracted from C and $\mathfrak{R}$. $(C, \mathfrak{T})$ will be equi-satisfiable to $(C, \mathfrak{R})$, which is proven in the subsequent Lemma (see below). The basic idea of the TBox $\mathfrak{T}$ is to handle universal value restrictions $\forall R.C$ correctly by "propagating" annotated atomic marker concepts of the form $\boxed{(\forall R.D)_S}$ to successors individuals, simulating the propagation of $(\forall R.D)_{S,w}$-constraints as done by the $\exists\forall$-rule of the tableaux calculus. Therefore, also *indirect* R-successors will be elements of $D^{\mathcal{I}}$. The TBox $\mathfrak{T}$ (which will be cyclical in most cases) is defined as follows:

[8] Please recall that $\mathcal{ALC}$ looses the *finite* tree model property if general TBoxes are taken into account. There are either cyclical models, or infinite tree models, and this can be shown easily using an unraveling construction like the ones given above.

$$
\begin{aligned}
\mathfrak{T}_0 := \; & \{ \forall R.D \sqsubseteq \boxed{\forall R.D} \mid \forall R.D \in \mathsf{sub}(C) \} \cup \\
& \{ \boxed{\forall R.D} \sqsubseteq \forall (\sqcup_{S \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R})} S).\boxed{(\forall R.D)_S} \mid \forall R.D \in \mathsf{sub}(C) \}
\end{aligned}
$$

$\mathfrak{T}_1 := \mathfrak{T}_0 \cup \{ \boxed{(\forall R.D)_R} \sqsubseteq D \mid \boxed{(\forall R.D)_R} \in \mathsf{sub}(\mathfrak{T}_0) \}$

$i := 1$

`WHILE` $\mathfrak{T}_i \neq \mathfrak{T}_{i-1}$

`DO`

  $i := i + 1$

  $\mathfrak{T}_i := \mathfrak{T}_{i-1} \cup \{ \boxed{(\forall R.D)_S} \sqsubseteq \forall (\sqcup_{T \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R})} T).\boxed{(\forall R.D)_{\mathsf{con}(S,T)}} \mid \boxed{(\forall R.D)_S} \in \mathsf{sub}(\mathfrak{T}_{i-1}) \}$

`OD`

It is obvious that the TBox $\mathfrak{T}$ can be computed in polynomial time and therefore, satisfiability of $(C, \mathfrak{R})$ is clearly in EXPTIME, since satisfiability of $\mathcal{ALC}$ with general TBoxes is in EXPTIME.

□

A few words regarding the construction of $\mathfrak{T}$ seem to be appropriate: All concepts of the form $\boxed{\cdots}$ are concept names from $\mathcal{N}_{\mathcal{C}}$. Such a concept name might be (partly) computed, e.g. if we write $\boxed{(\forall R.D)_{\mathsf{con}(S,T)}}$ and $\mathsf{con}(S,T) = U$, then $\boxed{(\forall R.D)_{\mathsf{con}(S,T)}} = \boxed{(\forall R.D)_U}$. The following example will make the transformation more transparent: if

$$C = (\exists S.(\neg D \sqcup \exists R.\exists R.\exists R.\neg D)) \sqcap (\forall S.(D \sqcap \forall R.D)),$$

$$\mathfrak{R} = \{R \circ R \sqsubseteq R, S \circ R \sqsubseteq R, R \circ S \sqsubseteq R, S \circ S \sqsubseteq R\},$$

then the resulting TBox $\mathfrak{T}$ will be

$$
\begin{aligned}
\mathfrak{T} = \{ \; & \forall S.(D \sqcap \forall R.D) \sqsubseteq \boxed{\forall S.(D \sqcap \forall R.D)}, \\
& \forall R.D \sqsubseteq \boxed{\forall R.D}, \\
& \boxed{(\forall R.D)_R} \sqsubseteq D, \\
& \boxed{(\forall S.(D \sqcap \forall R.D))_S} \sqsubseteq (D \sqcap \forall R.D), \\
& \boxed{\forall S.(D \sqcap \forall R.D)} \sqsubseteq \forall S.\boxed{(\forall S.(D \sqcap \forall R.D))_S}, \\
& \boxed{\forall S.(D \sqcap \forall R.D)} \sqsubseteq \forall R.\boxed{(\forall S.(D \sqcap \forall R.D))_R}, \\
& \boxed{\forall R.D} \sqsubseteq \forall S.\boxed{(\forall R.D)_S}, \\
& \boxed{\forall R.D} \sqsubseteq \forall R.\boxed{(\forall R.D)_R}, \\
& \boxed{(\forall S.(D \sqcap \forall R.D))_S} \sqsubseteq \forall S.\boxed{(\forall S.(D \sqcap \forall R.D))_R}, \\
& \boxed{(\forall S.(D \sqcap \forall R.D))_S} \sqsubseteq \forall R.\boxed{(\forall S.(D \sqcap \forall R.D))_R}, \\
& \boxed{(\forall S.(D \sqcap \forall R.D))_R} \sqsubseteq \forall S.\boxed{(\forall S.(D \sqcap \forall R.D))_R}, \\
& \boxed{(\forall S.(D \sqcap \forall R.D))_R} \sqsubseteq \forall R.\boxed{(\forall S.(D \sqcap \forall R.D))_R}, \\
& \boxed{(\forall R.D)_S} \sqsubseteq \forall S.\boxed{(\forall R.D)_R}, \\
& \boxed{(\forall R.D)_S} \sqsubseteq \forall R.\boxed{(\forall R.D)_R}, \\
& \boxed{(\forall R.D)_R} \sqsubseteq \forall S.\boxed{(\forall R.D)_R}, \\
& \boxed{(\forall R.D)_R} \sqsubseteq \forall R.\boxed{(\forall R.D)_R} \quad \}.
\end{aligned}
$$

The following Lemma closes the gap in the previous proof of EXPTIME-completeness of $\mathcal{ALC}_{\mathcal{RASG}}$:

Lemma 6 Let C be in NNF, and $\mathfrak{R}$ be an admissible role box for $\mathfrak{R}$, and let $\mathfrak{T}$ be the TBox as defined above. Then, $(C, \mathfrak{T})$ is satisfiable iff $(C, \mathfrak{R})$ is.

Proof 8 $\Leftarrow$ Let $\mathcal{I} \models (C, \mathfrak{R})$ be a tree skeleton model. $\mathcal{SKEL}(\mathcal{I})$ has the form of a possibly infinite tree (according to the Proof of Lemma 1 there is always such a model if $(C, \mathfrak{R})$ is satisfiable at all), $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. We can then define $\mathcal{I}' = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}'})$ such that $\mathcal{I}' \models (C, \mathfrak{T})$:

- $R^{\mathcal{I}'} =_{def} R^{\mathcal{I}} \cap \mathcal{SKEL}(\mathcal{I})$, for all roles $R \in \mathsf{roles}(C)$,
- $D^{\mathcal{I}'} =_{def} D^{\mathcal{I}}$, for all concept names $D \in \mathsf{sub}(C) \cap \mathcal{N}_{\mathcal{C}}$,
- $\boxed{\forall R.D}^{\mathcal{I}'} =_{def} (\forall R.D)^{\mathcal{I}}$, for all $\forall R.D \in \mathsf{sub}(C)$,
- $\boxed{(\forall R.D)_S}^{\mathcal{I}'} =_{def} \{\, x \mid \exists y : \langle x, y\rangle \in S^{\mathcal{I}}, x \in (\forall R.D)^{\mathcal{I}} \,\}$, for all $\boxed{(\forall R.D)_S} \in \mathsf{sub}(\mathfrak{T})$, and $S \in \mathsf{roles}(\mathfrak{T}) \cup \mathsf{roles}(C)$ (please note that $\mathsf{roles}(\mathfrak{R}) = \mathsf{roles}(\mathfrak{T})$),
- $\emptyset$, for all other not mentioned concept and role names.

Because $\mathcal{I} \models C$ and $\mathcal{I}$ is a tree skeleton model, it is still the case that $\mathcal{I}' \models C$. We can safely interpret the irrelevant concept and role names with the empty set, due to Proposition 1. Even though the interpretation of the roles has been changed by discarding the "indirect" edges in the interpretations, it is still the case that the existential value restrictions which occur in C are interpreted correctly: for all $\exists R.D \in \mathsf{sub}(C)$, if $x \in (\exists R.D)^{\mathcal{I}}$, then also $x \in (\exists R.D)^{\mathcal{I}'}$, since $\mathcal{I}$ is a tree skeleton model. For the other types of subconcepts this is also obvious. Therefore, $\mathcal{I}' \models C$.

We also have to show that $\mathcal{I}' \models \mathfrak{T}$. According to the transformation, there are four types of GCIs, and we show that each GCI is satisfied by $\mathcal{I}'$:

- $\forall R.D \sqsubseteq \boxed{\forall R.D}$: obvious by the definition of $\mathcal{I}'$.
- $\boxed{\forall R.D} \sqsubseteq \forall(\sqcup_{S \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R})} S).\boxed{(\forall R.D)_S}$: suppose there is some individual $x \in \Delta^{\mathcal{I}}$ with $x \in (\boxed{\forall R.D}^{\mathcal{I}'})$, but $x \notin (\forall(\sqcup_{S \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R})} S).\boxed{(\forall R.D)_S})^{\mathcal{I}'}$. Then there must be some individual y, with $\langle x, y\rangle \in T^{\mathcal{I}'}$, $T \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{T})$, with $y \notin \boxed{(\forall R.D)_T}^{\mathcal{I}'}$. According to the definition of $T^{\mathcal{I}'}$, also $\langle x, y\rangle \in T^{\mathcal{I}}$. But since $x \in \boxed{\forall R.D}^{\mathcal{I}'}$ we have by construction of $\mathcal{I}'$ also $x \in (\forall R.D)^{\mathcal{I}}$ and therefore $y \in \boxed{(\forall R.D)_T}^{\mathcal{I}'}$, contradicting the assumption.
- $\boxed{(\forall R.D)_R} \sqsubseteq D$: if $y \in \boxed{(\forall R.D)_R}^{\mathcal{I}'}$, then there is some x with $\langle x, y\rangle \in R^{\mathcal{I}}$, and $x \in (\forall R.D)^{\mathcal{I}}$. Since $\mathcal{I} \models C$, it must be the case that $y \in D^{\mathcal{I}}$.

- $\boxed{(\forall R.D)_S} \sqsubseteq \forall(\sqcup_{T\in\mathsf{roles}(C)\cup\mathsf{roles}(\mathfrak{R})}T).\boxed{(\forall R.D)_{\mathsf{con}(S,T)}}$: suppose there is some node $x \in \Delta^{\mathcal{I}}$ with $x \in \boxed{(\forall R.D)_S}^{\mathcal{I}}$, but $x \notin (\forall(\sqcup_{T\in\mathsf{roles}(C)\cup\mathsf{roles}(\mathfrak{R})}T).\boxed{(\forall R.D)_{\mathsf{con}(S,T)}})^{\mathcal{I}}$. Then there must be some individual y, with $<x,y> \in T^{\mathcal{I}'}$, $T \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{T})$, with $y \notin \boxed{(\forall R.D)_{\mathsf{con}(S,T)}}^{\mathcal{I}'}$. However, due to $x \in \boxed{(\forall R.D)_S}^{\mathcal{I}}$ there is some $w \in (\forall R.D)^{\mathcal{I}}$, $<w,x> \in S^{\mathcal{I}}$. Since $\mathfrak{R}$ is admissible for C, there must be some role axiom ra with $\mathsf{con}(ra) = \mathsf{con}(S,T)$. Due to $\mathcal{I} \models \mathfrak{R}$, $<w,y> \in (\mathsf{con}(S,T))^{\mathcal{I}}$, which shows that $y \in \boxed{(\forall R.D)_{\mathsf{con}(S,T)}}^{\mathcal{I}'}$ by definition of $\mathcal{I}'$, contradicting the assumption.

$\Rightarrow$ Let $\mathcal{I} \models (C,\mathfrak{T})$ be a possibly infinite tree model. Again we emphasize that there is no loss of generality by considering only this class of models. We then define $\mathcal{I}'$ such that that $\mathcal{I}' \models (C,\mathfrak{R})$. $\mathcal{I}'$ is a tree skeleton model. If $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, then $\mathcal{I}' = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}'})$ is defined as follows:

- $D^{\mathcal{I}'} =_{def} D^{\mathcal{I}}$, for all concept names $D \in \mathcal{N}_{\mathcal{C}} \cap \mathsf{sub}(C)$
- $R^{\mathcal{I}'} =_{def} R^{\mathcal{I}}$, for all role names $R \in \mathcal{N}_{\mathcal{R}} \cap \mathsf{roles}(C)$
- `while` $\mathcal{I}' \not\models \mathfrak{R}$ `do`
    `for each` $R \circ S \sqsubseteq T \in \mathfrak{R}$ `do`
      $T^{\mathcal{I}'} := T^{\mathcal{I}'} \cup R^{\mathcal{I}'} \circ S^{\mathcal{I}'}$
    `od`
  `od`
- $\emptyset$, for all other not mentioned concept and role names.

It is obvious by definition of $\cdot^{\mathcal{I}'}$ that $\mathcal{I}' \models \mathfrak{R}$.

However, since the interpretations of the relevant roles have been *extended* ($R^{\mathcal{I}} \subseteq R^{\mathcal{I}'}$; if $R^{\mathcal{I}'}$ has been set to $\emptyset$, then R is irrelevant) it is not obvious that the universal value restrictions are still interpreted correctly – for all other subconcepts of C it is obvious that they are still interpreted correctly.

If $x \in (\forall R.D)^{\mathcal{I}}$, then we have to show that still $x \in (\forall R.D)^{\mathcal{I}'}$. Assume that $x \in (\forall R.D)^{\mathcal{I}}$, but $x \notin (\forall R.D)^{\mathcal{I}'}$. Then, there must be some $y \in \Delta^{\mathcal{I}}$, with $<x,y> \in R^{\mathcal{I}'}$, $<x,y> \notin R^{\mathcal{I}}$, and $y \notin D^{\mathcal{I}'}$, $y \notin D^{\mathcal{I}}$, since otherwise already $x \notin (\forall R.D)^{\mathcal{I}}$. We have to show that $y \in D^{\mathcal{I}}$: let w be the path in the tree model connecting x and y, $w = R_1 R_2 \ldots R_n$, and call the nodes lying on the path x_i, with $x = x_0$, $y = x_n$, $<x_0,x_1> \in R_1^{\mathcal{I}}$, $<x_1,x_2> \in R_2^{\mathcal{I}}$, ..., $<x_{n-1},x_n> \in R_n^{\mathcal{I}}$. Since $\mathcal{I} \models \mathfrak{T}$, $x_0 \in (\forall R.D)^{\mathcal{I}}$ and $\forall R.D \sqsubseteq \boxed{\forall R.D} \in \mathfrak{T}$, also $x_0 \in (\boxed{\forall R.D})^{\mathcal{I}}$. Since $x_0 \in (\boxed{\forall R.D})^{\mathcal{I}}$ and $\boxed{\forall R.D} \sqsubseteq \forall(\sqcup_{S\in\mathsf{roles}(C)\cup\mathsf{roles}(\mathfrak{R})}S).\boxed{(\forall R.D)_S} \in \mathfrak{T}$ with $<x_0,x_1> \in R_1^{\mathcal{I}}$, also $x_1 \in \boxed{(\forall R.D)_{R1}}^{\mathcal{I}}$. From $x_1 \in \boxed{(\forall R.D)_{R1}}^{\mathcal{I}}$ and $<x_1,x_2> \in R_2^{\mathcal{I}}$ we conclude that $x_2 \in \boxed{(\forall R.D)_{\mathsf{con}(R1,R2)}}^{\mathcal{I}}$, due to

$\boxed{(\forall R.D)_S} \sqsubseteq \forall(\sqcup_{T \in \mathsf{roles}(C) \cup \mathsf{roles}(\mathfrak{R})} T).\boxed{(\forall R.D)_{\mathsf{con}(S,T)}} \in \mathfrak{T}$. An iterative application of this GCI finally gives $y = x_n \in \boxed{(\forall R.D)_R}^{\mathcal{I}}$. Due to $\boxed{(\forall R.D)_R} \sqsubseteq D \in \mathfrak{T}$ we have $y \in D^{\mathcal{I}}$. Please note that we really know that $y \in \boxed{(\forall R.D)_R}^{\mathcal{I}}$ and not only $y \in \boxed{(\forall R.D)_S}^{\mathcal{I}}$, for some role S – if $\langle x, y \rangle \in R^{\mathcal{I}'}$ and w is the path connecting x and y, then due to the associativity of the role box, every composition possibility yields the same role, in this case R. By which sequence of composition possibilities $\langle x, y \rangle \in R^{\mathcal{I}'}$ has been obtained is irrelevant. For example, if $w = R_1R_2R_3$, we know that $y \in \boxed{(\forall R.D)_R}^{\mathcal{I}}$, since the result of $\mathsf{con}(\mathsf{con}(R1, R2), R3)$ must be R, no matter how $\langle x, y \rangle \in R^{\mathcal{I}'}$ has been obtained in the while loop (e.g., it might have been obtained by $(\mathsf{con}(R_2, R_3))^{\mathcal{I}'} := R_2^{\mathcal{I}} \circ R_3^{\mathcal{I}}$ followed by $R^{\mathcal{I}'} = (\mathsf{con}(R_1, \mathsf{con}(R_2, R_3)))^{\mathcal{I}'} := R_1^{\mathcal{I}} \circ (\mathsf{con}(R_2, R_3))^{\mathcal{I}'}$). This shows that $\mathcal{I}' \models C$, and summing up we have shown that $\mathcal{I}' \models (C, \mathfrak{R})$. □

5 $\mathcal{ALCN}_{\mathcal{RASG}}$ is Undecidable

Since $\mathcal{ALC}_{\mathcal{RASG}}$ is decidable, the question arises naturally whether further extensions of $\mathcal{ALC}_{\mathcal{RASG}}$, e.g. by *inverse roles* ($\mathcal{I}$) or *unqualified number restrictions* ($\mathcal{N}$), still remain decidable. We show that even if only unqualified number restrictions (which are usually an easy and unproblematic extension of DLs) are added to $\mathcal{ALC}_{\mathcal{RASG}}$, the resulting logic $\mathcal{ALCN}_{\mathcal{RASG}}$ is undecidable again. Obviously, since $\mathcal{ALC}_{\mathcal{RA}}$ is a sub-logic of $\mathcal{ALC}_{\mathcal{RASG}}$, $\mathcal{ALCN}_{\mathcal{RASG}}$ is undecidable as well; and the same obviously also holds for *qualified number restrictions* ($\mathcal{ALCQ}_{\mathcal{RASG}}$, $\mathcal{ALCQ}_{\mathcal{RA}}$). The proof is very similar to the ones given by Sattler in [11], where it is shown that the languages $\mathcal{ALC}_{+}\mathcal{N}(\circ, \sqcup)$, $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$, $\mathcal{ALCN}(\circ, \sqcap)$ and $\mathcal{ALC}_{+}\mathcal{N}(\circ)$ are all undecidable.

To obtain $\mathcal{ALCN}_{\mathcal{RASG}}$, the syntax of concept terms is extended by incorporating the constructor called $\mathcal{N}$, with the following additional concept formation rule:

- If $R \in \mathcal{N}_{\mathcal{R}}$ is a role and $n \in \mathbb{N} \cup \{0\}$, then the expressions $(\geq R\ n)$ and $(\leq R\ n)$ are concepts as well.

The operators roles and sub are appropriately modified, e.g. $\mathsf{roles}((\geq R\ n)) = \{R\}$, etc. Still, for a role box $\mathfrak{R}$ that is admissible for a concept C, $\mathsf{roles}(C) \subseteq \mathsf{roles}(\mathfrak{R})$ must hold, but with the modified definition of the operator roles.

The semantics is defined in the obvious way:

- $(\geq R\ n)^{\mathcal{I}} =_{def} \{\, i \in \Delta^{\mathcal{I}} \mid \#(R^{\mathcal{I}} \cap (\{i\} \times \Delta^{\mathcal{I}})) \geq n \,\}$
- $(\leq R\ n)^{\mathcal{I}} =_{def} \{\, i \in \Delta^{\mathcal{I}} \mid \#(R^{\mathcal{I}} \cap (\{i\} \times \Delta^{\mathcal{I}})) \leq n \,\}$.

The undecidability proof of $\mathcal{ALCN}_{\mathcal{RASG}}$ is given by a reduction from the so-called *domino problem*:

Definition 12 (Domino System) A domino system $\mathcal{DOM}$ is a triple $(\mathcal{D}, \mathcal{H}, \mathcal{V})$, where $\mathcal{D} = \{d_1, \ldots, d_n\}$ is a non-empty set of so-called *domino types*, $\mathcal{V} \subseteq \mathcal{D} \times \mathcal{D}$ is the vertical matching relation, and $\mathcal{H} \subseteq \mathcal{D} \times \mathcal{D}$ is the horizontal matching relation.

A *solution* of a domino system is a function $\mathsf{f} : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{D}$ (in the following we assume that $0 \in \mathbb{N}$) such that the matching relationships of the domino types are respected, i.e. for all $(i, j) \in \mathbb{N} \times \mathbb{N}$: $(\mathsf{f}(i, j), \mathsf{f}(i + 1, j)) \in \mathcal{H}$ and $(\mathsf{f}(i, j), \mathsf{f}(i, j + 1)) \in \mathcal{V}$.

It is an undecidable problem whether a domino system has a solution; i.e., f is a non-recursive (uncomputable) function. Given an arbitrary domino system $\mathcal{DOM}$ we construct an $\mathcal{ALCN}_{\mathcal{RASG}}$ concept C and a role box $\mathfrak{R}$ that is admissible for C such that $(C, \mathfrak{R})$ is satisfiable iff $\mathcal{DOM}$ has a solution. The role box $\mathfrak{R}$ corresponds to the following composition table:

$\circ$	R_X	R_Y	R_Z	R_U
R_X	R_U	R_Z	R_U	R_U
R_Y	R_Z	R_U	R_U	R_U
R_Z	R_U	R_U	R_U	R_U
R_U	R_U	R_U	R_U	R_U

Therefore, $\mathfrak{R} =_{def} \{R_X \circ R_X \sqsubseteq R_U, R_X \circ R_Y \sqsubseteq R_Z, \ldots\}$. Obviously, $\mathfrak{R}$ is associative. Let $\mathcal{DOM} = (\mathcal{D}, \mathcal{H}, \mathcal{V})$ and further assume, that $\mathcal{D} \subseteq \mathcal{N_C}$. Now, the concept C can be defined:

$$C =_{def} X \sqcap (\forall R_X.X) \sqcap (\forall R_Y.X) \sqcap (\forall R_Z.X) \sqcap (\forall R_U.X), \text{ where}$$

$$\begin{aligned} X =_{def} \; & M \sqcap (\geq R_X\ 1) \sqcap (\geq R_Y\ 1) \sqcap \\ & (\leq R_X\ 1) \sqcap (\leq R_Y\ 1) \sqcap (\leq R_Z\ 1), \text{ and} \end{aligned}$$

$$\begin{aligned} M =_{def} \; & \sqcup_{D_i \in \mathcal{D}} (D_i \sqcap (\sqcap_{D_j \in \mathcal{D}, D_i \neq D_j} \neg D_j)) \sqcap \\ & \sqcap_{D_i \in \mathcal{D}} (D_i \Rightarrow (\ \forall R_X.(\sqcup_{(D_i,D_j) \in \mathcal{H}} D_j) \sqcap \\ & \qquad \forall R_Y.(\sqcup_{(D_i,D_j) \in \mathcal{V}} D_j))) \end{aligned}$$

Obviously, $\mathfrak{R}$ is an admissible role box for C.

Lemma 7 $(C, \mathfrak{R})$ is satisfiable iff the domino system $\mathcal{DOM} = (\mathcal{D}, \mathcal{H}, \mathcal{V})$ has a solution.

Proof 9 **"⇒"** Let $\mathcal{I} \models (C, R)$, with $x_{0,0} \in C^{\mathcal{I}}$. Using $\mathcal{I}$ we construct a solution f of $\mathcal{DOM}$ as follows:

- $\mathsf{f}(m,n) =_{def} D$ iff $<x_{0,0}, x_{m,n}> \in (R_X^{\mathcal{I}})^m \circ (R_Y^{\mathcal{I}})^n$, for some $m, n \in \mathbb{N} \cup \{0\}$ and $x_{m,n} \in D^{\mathcal{I}}$ for some $D \in \mathcal{D}$.
  Please note that $(R^{\mathcal{I}})^n =_{def} \underbrace{R^{\mathcal{I}} \circ R^{\mathcal{I}} \circ \ldots \circ R^{\mathcal{I}}}_{n \text{ times}}$, and
  $(R^{\mathcal{I}})^0 =_{def} \{ <x, x> \mid x \in \Delta^{\mathcal{I}} \}$. Hence $<x_{0,0}, x_{0,0}> \in (R_X^{\mathcal{I}})^0 \circ (R_Y^{\mathcal{I}})^0$.

We have to show that

1. f is well-defined; i.e. that f really defines a *total function* $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{D}$, and that
2. f is indeed a *solution* to the domino problem; i.e. that the matching conditions $(\mathsf{f}(m,n), \mathsf{f}(m+1,n)) \in \mathcal{H}$ and $(\mathsf{f}(m,n), \mathsf{f}(m,n+1)) \in \mathcal{V}$ are satisfied for all $(m,n) \in \mathbb{N} \times \mathbb{N}$.

To show 1, we observe that $x_{0,0} \in X^{\mathcal{I}}$, also $x_{0,0} \in M^{\mathcal{I}}$, and therefore, $x_{0,0} \in D^{\mathcal{I}}$ for *exactly one* $D \in \mathcal{D}$ (see definition of M, first conjunct). Due to $<x_{0,0}, x_{0,0}> \in (R_X^{\mathcal{I}})^0 \circ (R_Y^{\mathcal{I}})^0$ for $m = 0, n = 0$ we have $\mathsf{f}(0,0) = D$. Now, since $x_{0,0} \in ((= R_X\ 1) \sqcap (= R_Y\ 1))^{\mathcal{I}}$ there is *exactly one* successor $x_{1,0}$ with $<x_{0,0}, x_{1,0}> \in R_X^{\mathcal{I}}$ and *exactly one* successor $x_{0,1}$ with $<x_{0,0}, x_{0,1}> \in R_Y^{\mathcal{I}}$. Note that it is important that these successors are uniquely defined, since f would be ambiguous otherwise. Due to $x_{0,0} \in ((\forall R_X.X) \sqcap (\forall R_Y.X))^{\mathcal{I}}$ we have $x_{1,0}, x_{0,1} \in X^{\mathcal{I}}$. We conclude that also $x_{1,0} \in D_i^{\mathcal{I}}$ and $x_{0,1} \in D_j^{\mathcal{I}}$, for exactly one D_i resp. D_j. Since $<x_{0,0}, x_{1,0}> \in (R_X^{\mathcal{I}})^1 \circ (R_Y^{\mathcal{I}})^0$ and $<x_{0,0}, x_{0,1}> \in (R_X^{\mathcal{I}})^0 \circ (R_Y^{\mathcal{I}})^1$ this shows that $\mathsf{f}(1,0) = D_i$ and $\mathsf{f}(0,1) = D_j$ are well-defined. Again, due to $x_{1,0}, x_{0,1} \in X^{\mathcal{I}}$, both have exactly one R_X and one R_Y successor. Since $\{R_X \circ R_Y \sqsubseteq R_Z, R_Y \circ R_X \sqsubseteq R_Z\} \subseteq \mathfrak{R}$ these successors are also R_Z-successors of $x_{0,0}$, and since $x_{0,0} \in (\leq R_Z\ 1)^{\mathcal{I}}$ the $(R_X^{\mathcal{I}})^1 \circ (R_Y^{\mathcal{I}})^1$-successor of $x_{0,0}$ coincides with its $R_Y^{\mathcal{I}} \circ R_X^{\mathcal{I}}$-successor, for $n = 1, m = 1$. Because $x_{0,0} \in (\forall R_Z.X)^{\mathcal{I}}$ we have $x_{1,1} \in X^{\mathcal{I}}$, $x_{1,1} \in M^{\mathcal{I}}$ and finally $x_{1,1} \in D^{\mathcal{I}}$, for exactly one D. This shows that $\mathsf{f}(1,1) = D$ is well-defined. Again, due to $\{x_{1,0}, x_{0,1}, x_{1,1}\} \subseteq X^{\mathcal{I}}$, all have exactly one R_X and R_Y successor. Due to $\{R_Z \circ R_X \sqsubseteq R_U, R_Z \circ R_Y \sqsubseteq R_U, R_X \circ R_X \sqsubseteq R_U, R_Y \circ R_Y \sqsubseteq R_U\} \subseteq \mathfrak{R}$, these successors are R_U-successors of $x_{0,0}$, and therefore they are members of $X^{\mathcal{I}}$ as well, because $x_{0,0} \in (\forall R_U.X)^{\mathcal{I}}$. Again, due to $\{R_U \circ R_X \sqsubseteq R_U, R_U \circ R_Y \sqsubseteq R_U\} \subseteq \mathfrak{R}$, their successors will also R_U-successor of $x_{0,0}$, etc. Summing up, if $<x_{0,0}, x_{m,n}> \in (R_X^{\mathcal{I}})^m \circ (R_Y^{\mathcal{I}})^n$, then also $<x_{0,0}, x_{m,n}> \in (R_Y^{\mathcal{I}})^n \circ (R_X^{\mathcal{I}})^m$, and $<x_{0,0}, x_{m,n}> \in R_i^{\mathcal{I}}$ for exactly one $R_i \in \{R_X, R_Y, R_Z, R_U\}$, which shows that for all individuals reachable via $<x_{0,0}, x_{m,n}> \in (R_X^{\mathcal{I}})^m \circ (R_Y^{\mathcal{I}})^n$, $x_{m,n} \in X^{\mathcal{I}}$ holds. Therefore, $\mathsf{f}(m,n)$ is well-defined.

To show 2, it suffices to observe that all individuals that are reachable from $<x_{0,0}, x_{m,n}> \in (R_X^{\mathcal{I}})^m \circ (R_Y^{\mathcal{I}})^n$, for some $m, n \in \mathbb{N} \cup \{0\}$, $x_{m,n} \in X^{\mathcal{I}}$ holds, and therefore also $x_{m,n} \in M^{\mathcal{I}}$. The second conjunct of M $(D_i \Rightarrow (\forall R_X.(\sqcup_{(D_i,D_j)\in\mathcal{H}} D_j) \sqcap \forall R_Y.(\sqcup_{(D_i,D_j)\in\mathcal{V}} D_j)))$ obviously enforces that the horizontal and vertical matching conditions are satisfied, and (as already noted) the first condition ensures that $x_{m,n} \in D_i^{\mathcal{I}}$, for some $D_i \in \mathcal{D}$.

Summing up we have shown that f (as constructed above) is a solution to $\mathcal{DOM}$.

"$\Leftarrow$" Let f be a solution of $\mathcal{DOM} = (\mathcal{D}, \mathcal{H}, \mathcal{V})$. Using f we construct an infinite interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $\mathcal{I} \models (C, \mathfrak{R})$ and all roles are interpreted as disjoint:

- $\Delta^{\mathcal{I}} =_{def} \{ x_{i,j} \mid (i,j) \in \mathbb{N} \times \mathbb{N} \}$
- $D^{\mathcal{I}} =_{def} \{ x_{i,j} \mid \mathsf{f}(i,j) = D \}$ for all $D \in \mathcal{DOM}$
- $R_X^{\mathcal{I}} =_{def} \{ (x_{i,j}, x_{i+1,j}) \mid (i,j) \in \mathbb{N} \times \mathbb{N} \}$
- $R_Y^{\mathcal{I}} =_{def} \{ (x_{i,j}, x_{i,j+1}) \mid (i,j) \in \mathbb{N} \times \mathbb{N} \}$
- $R_Z^{\mathcal{I}} =_{def} \{ (x_{i,j}, x_{i+1,j+1}) \mid (i,j) \in \mathbb{N} \times \mathbb{N} \}$
- $R_U^{\mathcal{I}} =_{def} (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus (\{ (i,i) \mid i \in \Delta^{\mathcal{I}} \} \cup R_X^{\mathcal{I}} \cup R_Y^{\mathcal{I}} \cup R_Z^{\mathcal{I}})$

First of all its is obviously the case that all roles are interpreted as disjoint. It is also easy to check that $\mathcal{I} \models \mathfrak{R}$, i.e. $R_X^{\mathcal{I}} \circ R_Y^{\mathcal{I}} \subseteq R_Z^{\mathcal{I}}$, $R_Y^{\mathcal{I}} \circ R_X^{\mathcal{I}} \subseteq R_Z^{\mathcal{I}}$, etc. We show that $\mathcal{I} \models C$, due to $x_{0,0} \in C^{\mathcal{I}}$. We first prove that $M^{\mathcal{I}} = \Delta^{\mathcal{I}}$ (recall that $M =_{def} \sqcup_{D_i\in\mathcal{D}}(D_i \sqcap (\sqcap_{D_j\in\mathcal{D}, D_i\neq D_j} \neg D_j)) \sqcap (\sqcap_{D_i\in\mathcal{D}}(D_i \Rightarrow (\forall R_X.(\sqcup_{(D_i,D_j)\in\mathcal{H}} D_j) \sqcap \forall R_Y.(\sqcup_{(D_i,D_j)\in\mathcal{V}} D_j)))))$. Suppose that $x_{x,y} \notin M^{\mathcal{I}}$. Then, either $x_{x,y} \notin (\sqcup_{D_i\in\mathcal{D}}(D_i \sqcap (\sqcap_{D_j\in\mathcal{D}, D_i\neq D_j} \neg D_j)))^{\mathcal{I}}$ or
$x_{x,y} \notin (\sqcap_{D_i\in\mathcal{D}}(D_i \Rightarrow (\forall R_X.(\sqcup_{(D_i,D_j)\in\mathcal{H}} D_j) \sqcap \forall R_Y.(\sqcup_{(D_i,D_j)\in\mathcal{V}} D_j))))^{\mathcal{I}}$. In the first case, the contradiction is derived immediately, since f is a *total function* that maps elements from $\mathbb{N} \times \mathbb{N}$ to $\mathcal{D}$. In the second case, the contradiction is due to the fact that f is a solution of $\mathcal{DOM}$: if $x_{x,y} \in D_i^{\mathcal{I}}$ for some $D_i \in \mathcal{D}$, then $\mathsf{f}(x,y) = D_i$, and due to the matching conditions also $(D_i, \mathsf{f}(x+1,y)) \in \mathcal{H}$ and $(D_i, \mathsf{f}(x,y+1)) \in \mathcal{V}$. If $\mathsf{f}(x+1,y) = D_j$ and $\mathsf{f}(x,y+1) = D_k$, then $(D_i, D_j) \in \mathcal{H}$, $(D_i, D_k) \in \mathcal{V}$ and by construction of $\cdot^{\mathcal{I}}$ also $x_{x+1,y} \in D_j^{\mathcal{I}}$ and $x_{x,y+1} \in D_k^{\mathcal{I}}$. But since $\{x_{x+1,y}\} = \{ x_{x',y'} \mid <x_{x,y}, x_{x',y'}> \in R_X^{\mathcal{I}} \}$ and $\{x_{x,y+1}\} = \{ x_{x',y'} \mid <x_{x,y}, x_{x',y'}> \in R_Y^{\mathcal{I}} \}$ (by definition of $\cdot^{\mathcal{I}}$) we have $x_{x,y} \in ((\forall R_X.D_j) \sqcap (\forall R_Y.D_k))^{\mathcal{I}}$ and therefore $x_{x,y} \in (\forall R_X.(\sqcup_{(D_i,D_j)\in\mathcal{H}} D_j) \sqcap \forall R_Y.(\sqcup_{(D_i,D_j)\in\mathcal{V}} D_j))^{\mathcal{I}}$. Contradiction. This shows that $\Delta^{\mathcal{I}} = M^{\mathcal{I}}$. Now it is easy to see that also $\Delta^{\mathcal{I}} = X^{\mathcal{I}}$: for all $x_{x,y} \in \Delta^{\mathcal{I}}$ we even have $x_{x,y} \in (M \sqcap (= R_X\ 1) \sqcap (= R_Y\ 1) \sqcap (= R_Z\ 1))^{\mathcal{I}}$ (obviously, $(= R\ 1)$ is an abbreviation for $(\leq R\ 1) \sqcap (\geq R\ 1)$), due to $\{x_{x+1,y}\} = \{ x_{x',y'} \mid <x_{x,y}, x_{x',y'}> \in R_X^{\mathcal{I}} \}$, $\{x_{x,y+1}\} = \{ x_{x',y'} \mid <x_{x,y}, x_{x',y'}> \in R_Y^{\mathcal{I}} \}$,

and $\{x_{x+1,y+1}\} = \{\, x_{x',y'} \mid <x_{x,y}, x_{x',y'}> \in R_Z^{\mathcal{I}} \,\}$. But then it is also clear that $x_{x,y} \in (X \sqcap (\forall R_X.X) \sqcap (\forall R_Y.X) \sqcap (\forall R_Z.X) \sqcap (\forall R_U.X))^{\mathcal{I}}$ and hence $x_{0,0} \in C^{\mathcal{I}}$. □

Summing up we have shown that $\mathcal{ALCN}_{\mathcal{RASG}}$ is undecidable, and therefore also $\mathcal{ALCN}_{\mathcal{RA}}$, as well as other super-logics like $\mathcal{ALCQ}_{\mathcal{RASG}}$, $\mathcal{ALCQ}_{\mathcal{RA}}$, etc. ($\mathcal{Q}$ for *qualified number restrictions*). Of course, it does not make sense to consider the languages $\mathcal{ALCH}_{\mathcal{RASG}}$ and $\mathcal{ALCH}_{\mathcal{RA}}$, since *role hierarchies* (operator $\mathcal{H}$) provide an immediate contradiction to the assumed roles disjointness. Obviously it also doesn't make sense to consider $\mathcal{ALCH}_{\mathcal{RA}^{\ominus}}$, since $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$ is already undecidable.

However, it might be possible that the language $\mathcal{ALCI}_{\mathcal{RASG}}$ is still decidable. This has to be investigated in the future.

6 Summary, Discussion and Future Work

We have identified the language $\mathcal{ALC}_{\mathcal{RASG}}$ as a decidable fragment both of $\mathcal{ALC}_{\mathcal{RA}}$ and $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$. Due to the admissibility of role boxes it is impossible to enforce non-empty role intersections in $\mathcal{ALC}_{\mathcal{RASG}}$. Please note that this does not hold for $\mathcal{ALC}_{\mathcal{RA}}$:

Please observe that

$$(\exists R.((\exists S.\exists T.\top) \sqcap \forall Y.\bot) \sqcap \forall A.\bot, \mathfrak{R}(C))$$

w.r.t.

$$\{R \circ S \sqsubseteq A \sqcup B, S \circ T \sqsubseteq X \sqcup Y, A \circ T \sqsubseteq U, B \circ T \sqsubseteq V, R \circ X \sqsubseteq U, R \circ Y \sqsubseteq V\}$$

enforces a non-empty intersection between U and V. This example is therefore satisfiable in $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$, but not in $\mathcal{ALC}_{\mathcal{RA}}$. Obviously, any $(C, \mathfrak{R})$ which is satisfiable in $\mathcal{ALC}_{\mathcal{RA}}$ is also satisfiable in $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$, but the converse does not hold, as the given example demonstrates.

In [16], we have proposed a tableaux calculus for $\mathcal{ALC}_{\mathcal{RA}}$ which is similar to the one given here, but we were unable to prove soundness, completeness and termination. Like for $\mathcal{ALC}_{\mathcal{RASG}}$, a *complete* role box $\mathfrak{R}$ was needed for this calculus (see [16], [15] for a discussion). Comparing the $\mathcal{ALC}_{\mathcal{RA}}$ tableaux calculus with the $\mathcal{ALC}_{\mathcal{RASG}}$ calculus, one can see that a major source of non-determinism is absent in the latter. A major difference is that the $\exists\forall$-rule was *non-deterministic* in the $\mathcal{ALC}_{\mathcal{RA}}$ calculus, see Figure 3. The different choices for the labels of $\mathcal{L}_{\mathcal{N}}(x_j)$ correspond to the different composition-possibilities caused by the disjunctions

$\exists\forall$-rule for $\mathcal{ALC}_{\mathcal{RA}}$:

if 1. $\exists R.C_1 \in \mathcal{L}_{\mathcal{N}}(x_i)$

2. neither the $\sqcap$- nor the $\sqcup$- nor the $\forall$-rule is applicable to x_i

3. $\neg\exists {<}x_i, x_j{>} \in \mathcal{E}_{\mathcal{R}} : C_1 \in \mathcal{L}_{\mathcal{N}}(x_j)$

4. x_i is not **blocked**

then create a new node x_j with

$\mathcal{L}_{\mathcal{E}}({<}x_i, x_j{>}) := R$, $\mathcal{L}_{\mathcal{N}}(x_j) := \mathcal{L}$,

where $\mathcal{W} = \{ (w, S) \mid (\forall T.D)_{S,w} \in \mathcal{L}_{\mathcal{N}}(x_i) \}$,

and $\mathcal{L}$ is some set that can non-deterministically be constructed by:

for all $(w, S) \in \mathcal{W}$*:*

choose some $U \in \mathsf{con}(S, R)$*:*

$\mathcal{C}(w) = \{ (\forall T.D)_{U,wR} \mid (\forall T.D)_{S,w} \in \mathcal{L}_{\mathcal{N}}(x_i) \}$

$$\mathcal{L} = \{C_1\} \cup \{\forall R_?.\top\} \cup \bigcup_{(w,S)\in\mathcal{W}} \mathcal{C}(w) \cup \{ (\forall T.D)_{R,R} \mid \forall T.D \in \mathcal{L}_{\mathcal{N}}(x_i) \}$$

Figure 3: The $\exists\forall$-rule for $\mathcal{ALC}_{\mathcal{RA}}$

on the right-hand side of the role axioms $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$. However, non-deterministic role axioms are not allowed in admissible $\mathcal{ALC}_{\mathcal{RASG}}$ role boxes, and therefore, this source of complexity resp. non-determinism is absent.

When constructing the completion tree with the $\mathcal{ALC}_{\mathcal{RA}}$ calculus, an additional *clash-trigger* was needed. In the previous example which enforces a non-empty intersection between U and V, a so-called role-box clash would be detected. In order to be able to detect such clashes, this clash-trigger had to be "path global". In fact, the clash-trigger had to consider the labels of all individual lying on one path and check whether there are some edges R, S, T corresponding to a situation where ${<}x, y{>} \in R^{\mathcal{I}}$, ${<}y, z{>} \in S^{\mathcal{I}}$, ${<}x, z{>} \in T^{\mathcal{I}}$, but $T \notin \mathsf{con}(R, S)$. Of course, this corresponds to a violation of the disjointness requirement.

The problem with the $\mathcal{ALC}_{\mathcal{RA}}$ calculus was the lack of an appropriate blocking condition (see the fourth precondition of the $\exists\forall$-rule in Figure 3). When constructing the infinite tableau by "unraveling" the blocked completion tree, the blocking condition has to ensure that each path can be extended infinitely without producing a violation of the disjointness requirement at some time in the unraveling process. This infinite unraveling of the completion tree is unproblematic in $\mathcal{ALC}_{\mathcal{RASG}}$, due to the associativity, and ensures that no new composition possibilities can arise that might violate the disjointness requirement at some point during the unraveling. This problem is unsolved for $\mathcal{ALC}_{\mathcal{RA}}$ and might very well be undecidable. More precisely, the question is: under which circumstances is it possible to infinitely continue a path in the completion tree such that the same "pattern" of role combinations reproduces itself? Having identified the major differences between $\mathcal{ALC}_{\mathcal{RA}}$ and $\mathcal{ALC}_{\mathcal{RASG}}$, we have gained new insights that might be fruitful when reconsidering $\mathcal{ALC}_{\mathcal{RA}}$ again, since it is still left open whether $\mathcal{ALC}_{\mathcal{RA}}$ might be decidable or not.

Having identified the major differences between $\mathcal{ALC}_{\mathcal{RA}}$ and $\mathcal{ALC}_{\mathcal{RASG}}$, we have gained new insights that might be fruitful when reconsidering $\mathcal{ALC}_{\mathcal{RA}}$ again, since it is still left open whether $\mathcal{ALC}_{\mathcal{RA}}$ might be decidable or not. We have shown that $\mathcal{ALCN}_{\mathcal{RASG}}$ (and therefore also $\mathcal{ALCN}_{\mathcal{RA}}$) is again undecidable. Therefore, extending $\mathcal{ALC}_{\mathcal{RASG}}$ is not an easy task. For example, whether *inverse roles* can be incorporated into the language (yielding $\mathcal{ALCI}_{\mathcal{RASG}}$) is an open question.

Unfortunately, the proposed admissibility criterion for role boxes is very strong, singling out a lot of useful role boxes as invalid. However, we believe that admissible role boxes can be constructed, e.g. for spatial reasoning tasks like in [15]. For example, if one considers the relational composition table of the RCC8 calculus (in fact, the underlying structure is even a *relation algebra*), then this composition table *is* associative, but in a more general sense, since *disjunctions* appear on the right hand side of role axioms. An other promising idea might be to gives *names* to disjunctions of base relations and "expand" the composition table such that no disjunctions appear any longer. A role axiom of the form $S \circ T \sqsubseteq R_1 \sqcup \cdots \sqcup R_n$would be substituted by $S \circ T \sqsubseteq \boxed{R_1 R_2 \ldots R_n}$, where the new role name $\boxed{R_1 R_2 \ldots R_n} \in \mathcal{N_R}$ represents the disjunction of the listed roles, and the composition table would be augmented by additional role axioms, such that the overall structure is maintained. Of course, a role name must be given to every possible disjunction, yielding an exponential blow up in the number of roles. Even though the composition table would be exponentially larger than the original one, it might probably be associative and useful for spatial reasoning. These transformations and applications of $\mathcal{ALC}_{\mathcal{RASG}}$ have to be worked out in the future.

For the further exploration of $\mathcal{ALC}$ in combination with role boxes it seems that one key is to consider certain kinds of role boxes such that the mathematical properties of the underlying structures can be exploited. It might also be possible to exploit automata-based techniques (e.g. Büchi-automata on infinite trees).

7 Acknowledgments

We would like to thank Stéphane Demri, Harald Ganzinger, Volker Haarslev, Carsten Lutz, Ralf Möller and Bernd Neumann for valuable discussions on the topics covered in this paper. I am especially grateful to Stéphane Demri who helped me understand the relationships to grammar logics.

References

[1] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Twelfth International Conference on Artificial Intelligence, Darling Harbour, Sydney, Australia, Aug. 24-30, 1991*, pages 446–451, August 1991.

[2] F. Baader. Logic-based knowledge representation. In M.J. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today, Recent Trends and Developments*, number 1600 in Lecture Notes in Computer Science, pages 13–41. Springer Verlag, 1999.

[3] M. Baldoni. *Normal Multimodal Logics: Automated Deduction and Logic Programming.* PhD thesis, Università degli Studi di Torino, 1998.

[4] S. Demri. The Complexity of Regularity in Grammar Logics and Related Modal Logics. Submitted, October 2000.

[5] M. Fitting. Basic Modal Logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artifical Intelligence*, volume 1, pages 368–449. Oxford University Press, 1993.

[6] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics.* PhD thesis, University of Manchester, 1997.

[7] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[8] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, 1999.

[9] R. Molitor. Konsistenz von Wissensbasen in Beschreibungslogiken mit Rollenoperatoren. Diplomarbeit, RWTH Aachen, Germany, 1997.

[10] U. Sattler. A concept language extended with different kinds of transitive roles. In G. Görz and S. Hölldobler, editors, *20. Deutsche Jahrestagung für Künstliche Intelligenz*, number 1137 in Lecture Notes in Artificial Intelligence, pages 333–345. Springer Verlag, Berlin, 1996.

[11] U. Sattler. *Terminological knowledge representation systems in a process engineering application.* PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, 1998.

[12] U. Sattler. Description logics for the representation of aggregated objects. In W.Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence*. IOS Press, Amsterdam, 2000.

[13] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Twelfth International Conference on Artificial Intelligence, Darling Harbour, Sydney, Australia, Aug. 24-30, 1991*, pages 466–471, August 1991.

[14] M. Schmidt-Schauß. Subsumption in KL-ONE is Undecidable. In *Principle of Knowledge Representation and Reasoning – Proceedings of the First International Conference KR '89*, 1989.

[15] M. Wessel. Obstacles on the way to spatial reasoning with description logics – undecidability of $\mathcal{ALC}_{\mathcal{RA}^{\ominus}}$. Technical Report FBI–HH–M–297/00, University of Hamburg, Computer Science Department, October 2000. Available at `http://kogs-www.informatik.uni-hamburg.de/~mwessel/report4.{ps.gz | pdf}`.

[16] M. Wessel, V. Haarslev, and R. Möller. $\mathcal{ALC}_{\mathcal{RA}}$ – $\mathcal{ALC}$ with Role Axioms. In F. Baader and U. Sattler, editors, *Proceedings of the International Workshop in Description Logics 2000 (DL2000)*, number 33 in CEUR-WS, pages 21–30, Aachen, Germany, August 2000. RWTH Aachen. Proceedings online available from `http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-33/`.