TECHNISCHE
UNIVERSITÄT
DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Decidability of $\mathcal{SHIQ}$ with Complex Role Inclusion Axioms

Ian Horrocks and Ulrike Sattler

LTCS-Report 02-06

Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Hans-Grundig-Str. 25
01062 Dresden
Germany

Decidability of $\mathcal{SHIQ}$ with Complex Role Inclusion Axioms

October 1, 2003

Abstract

Motivated by medical terminology applications, we investigate the decidability of an expressive and prominent DL, $\mathcal{SHIQ}$, extended with role inclusion axioms of the form $R \circ S \sqsubseteq T$. It is well-known that a naive such extension leads to undecidability, and thus we restrict our attention to axioms of the form $R \circ S \sqsubseteq R$ or $S \circ R \sqsubseteq R$, which is the most important form of axioms in the applications that motivated this extension. Surprisingly, this extension is still undecidable. However, it turns out that restricting our attention further to acyclic sets of such axioms, we regain decidability. We present a tableau-based decision procedure for this DL and report on its implementation, which behaves well in practise and provides important additional functionality in a medical terminology application.

Contents

1 Motivation

The description logic (DL) $\mathcal{SHIQ}$ [Horrocks *et al.*, 2000; Horrocks and Sattler, 2002] is an expressive knowledge representation formalism that extends $\mathcal{ALC}$ [Schmidt-Schauß and Smolka, 1991] (a notational variant of the multi modal logic **K** [Schild, 1991]) with qualifying number restrictions, inverse roles, role inclusion axioms (RIAs) $R \sqsubseteq S$, and transitive roles. The development of $\mathcal{SHIQ}$ was motivated and inspired by several applications, one of which was the representation of knowledge about complex physically structured domains found, e.g., in chemical engineering [Sattler, 2000] and medical terminology [Rector and Horrocks, 1997].

Although $\mathcal{SHIQ}$ allows many important properties of such domains to be captured (e.g., transitive and inverse roles), one extremely useful feature that it cannot express is the "propagation" of one property along another property [Padgham and Lambrix, 1994; Rector, 2002; Spackman, 2000]. E.g., it may be useful to express the fact that certain locative properties are transfered across certain partonomic properties so that a trauma or lesion located in a part of a body structure is recognised as being located in the body structure as a whole. This enables highly desirable inferences such as a fracture of the neck of the femur being inferred to be a kind of fracture of the femur, or an ulcer located in the gastric mucosa being inferred to be a kind of stomach ulcer.

The importance of these kinds of inference, particularly in medical terminology applications, is illustrated by the fact that the Grail DL [Rector *et al.*, 1997], which was specifically designed for use with medical terminology, is able to represent these kinds of propagation (although it is quite weak in other respects). Moreover, in another medical terminology application using the comparatively inexpressive DL $\mathcal{ALC}$, a rather complex "work around" is performed in order to represent similar propagations [Schulz and Hahn, 2001].[1] Similar expressiveness was also provided in the CycL DL by the `transfersThro` statement [Lenat and Guha, 1989].

It is quite straightforward to extend $\mathcal{SHIQ}$ so that this kind of propagation can be expressed: simply allow for role inclusion axioms of the form $R \circ S \sqsubseteq P$, which then enforces all models $\mathcal{I}$ to interpret the composition of $R^\mathcal{I}$ with $S^\mathcal{I}$ as a sub-relation of $P^\mathcal{I}$. E.g., the above examples translate into

$$\mathtt{hasLocation} \circ \mathtt{isDivisionOf} \sqsubseteq \mathtt{hasLocation},$$

which implies that

$$\mathtt{Fracture} \sqcap \exists \mathtt{hasLocation}.(\mathtt{Neck} \sqcap \exists \mathtt{isDivisionOf}.\mathtt{Femur})$$

is subsumed by/a specialization of

$$\mathtt{Fracture} \sqcap \exists \mathtt{hasLocation}.\mathtt{Femur}$$

Unfortunately, this extension leads to the undecidability of the interesting inference problems; see [Wessel, 2001] for an undecidability proof and [Baldoni, 1998;

[1] In this approach, so-called *SEP-triplets* are used both to compensate for the absence of transitive roles in $\mathcal{ALC}$, and to express the propagation of properties across a distinguished "part-of" role.

Baldoni *et al.*, 1998; Demri, 2001] for the closely related family of *Grammar Logics*. On closer inspection of the problem, we observe that only RIAs of the form $R \circ S \sqsubseteq S$ or $S \circ R \sqsubseteq S$ are required in order to express propagation. Surprisingly, it turns out that $\mathcal{SHIQ}$ extended with this restricted form of RIAs is still undecidable. Decidability can be regained, however, by further restricting the set of RIAs to be *acyclic*. This additional restriction does not seem too severe: acyclic sets of RIAs should suffice for many applications, and cycles in RIAs may even be an indicator of modeling flaws [Rector, 2002]. We will call this decidable logic $\mathcal{RIQ}$.

Here, we present the above undecidability result and prove the decidability of $\mathcal{SHIQ}$ with acyclic RIAs via a tableau-based decision procedure for the satisfiability of concepts. The algorithm works by transforming concepts of the form $\forall R.C$, where R is a role, into concepts of the form $\forall \mathcal{A}.C$, where $\mathcal{A}$ is a non-deterministic finite automaton (NFA). These automata are derived from a set of RIAs $\mathcal{R}$ by first *unfolding* $\mathcal{R}$ into a set of implications $\exp(\mathcal{R})$ between regular expressions and roles, and then transforming the regular expressions into automata. The algorithm is of the same complexity as the one for $\mathcal{SHIQ}$—in the size of $\exp(\mathcal{R})$ and the length of the input concept—but, unfortunately, $\exp(\mathcal{R})$ is exponential in $\mathcal{R}$. We present a syntactic restriction that avoids this blow-up; investigating whether this blow-up can be avoided in general will be part of future work. Finally, in order to evaluate the practicability of this algorithm, we have extended the DL system FaCT [Horrocks, 1998] to deal with $\mathcal{RIQ}$. We discuss how the properties of NFAs are exploited in the implementation, and we present some preliminary results showing that the performance of the extended system is comparable with that of the original, and that it is able to compute inferences of the kind mentioned above w.r.t. the well known Galen medical terminology knowledge base [Rector and Horrocks, 1997; Horrocks, 1998].

2 Preliminaries

In this section, we introduce the DL $\mathcal{SH}^{+}\mathcal{IQ}$. This includes the definition of syntax, semantics, and inference problems.

Definition 1 *Let* $\mathbf{C}$ *be a set of* concept names *and* $\mathbf{R}$ *a set of* role names*. The set of* roles *is* $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$*. A* role inclusion axiom *is an expression of one of the following forms:*

$$\begin{array}{rcl} R_1 & \sqsubseteq & R_2, \\ R_1 \circ R_2 & \sqsubseteq & R_1, \text{ or} \\ R_1 \circ R_2 & \sqsubseteq & R_2, \end{array}$$

for roles R_i *(each of which can be inverse). A* generalised role hierarchy *is a set of role inclusion axioms.*

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *associates, with each role name* R*, a binary relation* $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$*. Inverse roles are interpreted as usual, i.e.,*

$$(R^-)^\mathcal{I} \quad = \quad \{\langle y,x\rangle \mid \langle x,y\rangle \in R^\mathcal{I}\} \quad \textit{for each role } R \in \mathbf{R}.$$

An interpretation $\mathcal{I}$ is a model *of a generalised role hierarchy $\mathcal{R}$ if it satisfies each inclusion assertion in $\mathcal{R}$, i.e., if*

$$\begin{array}{rcll} R_1^\mathcal{I} & \subseteq & R_2^\mathcal{I} & \textit{for each } R_1 \sqsubseteq R_2 \in \mathcal{R} \textit{ and} \\ R_1^\mathcal{I} \circ R_2^\mathcal{I} & \subseteq & R_3^\mathcal{I} & \textit{for each } R_1 \circ R_2 \sqsubseteq R_3 \in \mathcal{R}, \end{array}$$

where $\circ$ stands for the composition of binary relations.

We did not introduce *transitive role names* since adding $R \circ R \sqsubseteq R$ to the generalised role hierarchy is equivalent to saying that R is a transitive role.

To avoid considering roles such as R^{--}, we define a function Inv on roles such that $\mathsf{Inv}(R) = R^-$ if R is a role name, and $\mathsf{Inv}(R) = S$ if $R = S^-$.

Obviously, if $S \circ R \sqsubseteq S \in \mathcal{R}$ ($R \circ S \sqsubseteq S \in \mathcal{R}$ or $R \sqsubseteq S \in \mathcal{R}$), then each model of $\mathcal{R}$ also satisfies $\mathsf{Inv}(R) \circ \mathsf{Inv}(S) \sqsubseteq \mathsf{Inv}(S)$ ($\mathsf{Inv}(S) \circ \mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S)$ and $\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S)$). Thus, in the following, we assume that a generalised role hierarchy always contains both "directions" of a role inclusion axiom.

For a generalised role hierarchy $\mathcal{R}$, we define the relation $\underline{\sqsubseteq}^*$ to be the transitive-reflexive closure of $\sqsubseteq$ over $\mathcal{R}$. A role R is called a *sub-role* (resp. super-role) of a role S if $R \underline{\sqsubseteq}^* S$ (resp. $S \underline{\sqsubseteq}^* R$). Two roles R and S are *equivalent* ($R \equiv S$) if $R \underline{\sqsubseteq}^* S$ and $S \underline{\sqsubseteq}^* R$.

Now we are ready to define syntax and semantics of $\mathcal{SH}^+\mathcal{IQ}$-concepts.

Definition 2 *A role is* simple *if it does not have implied sub-roles, i.e., a simple role nor any of its sub-roles (or their inverse) occur on the right hand side of a role inclusion $R \circ S \sqsubseteq T$.*

*The set of $\mathcal{SH}^+\mathcal{IQ}$-*concepts *is the smallest set such that*

- *every concept name is a concept, and,*
- *if C, D are concepts, R is a role (possibly inverse), S is a simple role (possibly inverse) , and n is a nonnegative integer, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, $\exists R.C$, $(\geqslant nS.C)$, and $(\leqslant nS.C)$ are also concepts.*

A general concept inclusion axiom *(GCI) is an expression of the form $C \sqsubseteq D$ for two $\mathcal{SH}^+\mathcal{IQ}$-concepts C and D. A* terminology *is a set of GCIs.*

An interpretation *$\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a set $\Delta^\mathcal{I}$, called the* domain *of $\mathcal{I}$, and a* valuation *$\cdot^\mathcal{I}$ which maps every concept to a subset of $\Delta^\mathcal{I}$ and every role to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ such that, for all concepts C, D, roles R, S, and non-negative integers n, the following equations are satisfied, where $\sharp M$ denotes the cardinality of a set M:*

$$\begin{array}{rll} (C \sqcap D)^\mathcal{I} = & C^\mathcal{I} \cap D^\mathcal{I} & \textit{(conjunction)} \\ (C \sqcup D)^\mathcal{I} = & C^\mathcal{I} \cup D^\mathcal{I} & \textit{(disjunction)} \\ (\neg C)^\mathcal{I} = & \Delta^\mathcal{I} \setminus C^\mathcal{I} & \textit{(negation)} \\ (\exists R.C)^\mathcal{I} = & \{x \mid \exists y.\langle x,y\rangle \in R^\mathcal{I} \textit{ and } y \in C^\mathcal{I}\} & \textit{(exists restriction)} \\ (\forall R.C)^\mathcal{I} = & \{x \mid \forall y.\langle x,y\rangle \in R^\mathcal{I} \textit{ implies } y \in C^\mathcal{I}\} & \textit{(value restriction)} \\ (\geqslant nR.C)^\mathcal{I} = & \{x \mid \sharp\{y.\langle x,y\rangle \in R^\mathcal{I} \textit{ and } y \in C^\mathcal{I}\} \geqslant n\} & \textit{(at least restriction)} \\ (\leqslant nR.C)^\mathcal{I} = & \{x \mid \sharp\{y.\langle x,y\rangle \in R^\mathcal{I} \textit{ and } y \in C^\mathcal{I}\} \leqslant n\} & \textit{(at most restriction)} \end{array}$$

An interpretation $\mathcal{I}$ is a model *of a terminology $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each GCI $C \sqsubseteq D$ in $\mathcal{T}$.*

A concept C is called satisfiable *iff there is an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$. A concept D* subsumes *a concept C (written $C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for each interpretation. Two concepts are* equivalent *(written $C \equiv D$) if they mutually subsume each other. The above inference problems can be defined w.r.t. a generalised role hierarchy $\mathcal{R}$ and/or a terminology $\mathcal{T}$ in the usual way, i.e., by replacing* interpretation *with* model *of $\mathcal{R}$ and/or $\mathcal{T}$.*

For an interpretation $\mathcal{I}$, an element $x \in \Delta^{\mathcal{I}}$ is called an instance *of a concept C iff $x \in C^{\mathcal{I}}$.*

Some remarks are in order here: please note that number restrictions $(\geqslant nR.C)$ and $(\leqslant nR.C)$ are restricted to *simple* roles (intuitively these are (possibly inverse) roles that are not implied by others). The reason for this restriction is that satisfiability and subsumption of $\mathcal{SHIQ}$-concepts without this restriction are undecidable [Horrocks *et al.*, 1999], even without inverse roles and with *unqualifying* number restrictions only (these are number restrictions of the form $(\geqslant nR.\top)$ and $(\leqslant nR.\top)$ for $\top$ an abbreviation for $A \sqcup \neg A$).

For DLs that are closed under negation, subsumption and (un)satisfiability can be mutually reduced: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable, and C is unsatisfiable iff $C \sqsubseteq A \sqcap \neg A$ for some concept name A. It is straightforward to extend these reductions to generalised role hierarchies and terminologies. In contrast, the reduction of inference problems w.r.t. a terminology to pure concept inference problems (possibly w.r.t. a role hierarchy), deserve special care: in [Baader, 1991; Schild, 1991; Baader *et al.*, 1993], the *internalisation* of GCIs is introduced, a technique that realises exactly this reduction. For $\mathcal{SH}^+\mathcal{IQ}$, this technique only needs to be slightly modified. The following Lemma shows how general concept inclusion axioms can be *internalised* using a "universal" role U, that is, a transitive super-role of all roles occurring in $\mathcal{T}$ or $\mathcal{R}$ and their respective inverses.

Lemma 1 *Let C, D be concepts, $\mathcal{T}$ a terminology, and $\mathcal{R}$ a generalised role hierarchy. We define*

$$C_{\mathcal{T}} := \bigsqcap_{C_i \sqsubseteq D_i \in \mathcal{T}} \neg C_i \sqcup D_i.$$

Let U be a role that does not occur in $\mathcal{T}$, C, D, or $\mathcal{R}$. We set

$$\mathcal{R}_U := \mathcal{R} \cup \{U \circ U \sqsubseteq U\} \cup \{R \sqsubseteq U, \mathsf{Inv}(R) \sqsubseteq U \mid R \textit{ occurs in } \mathcal{T}, C, D, \textit{ or } \mathcal{R}\}.$$

- *C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is satisfiable w.r.t. $\mathcal{R}_U$.*
- *D subsumes C with respect to $\mathcal{T}$ and $\mathcal{R}$ iff $C \sqcap \neg D \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is unsatisfiable w.r.t. $\mathcal{R}_U$.*

The proof of Lemma 1 is similar to the ones that can be found in [Schild, 1991; Baader, 1991]. Most importantly, it must be shown that, (a) if a $\mathcal{SH}^+\mathcal{IQ}$-concept C is satisfiable with respect to a terminology $\mathcal{T}$ and a generalised role hierarchy $\mathcal{R}$, then $C, \mathcal{T}$ have a *connected* model, i. e., a model where any two elements are connect by a role path over those roles occurring in C and $\mathcal{T}$, and (b) if y is reachable from x via a role path (possibly involving inverse roles), then $\langle x, y \rangle \in U^{\mathcal{I}}$. These are easy consequences of the semantics and the definition of U.

Theorem 1 *Satisfiability and subsumption of $\mathcal{SH}^+\mathcal{IQ}$-concepts w.r.t. terminologies and generalised role hierarchies are polynomially reducible to (un)satisfiability of $\mathcal{SH}^+\mathcal{IQ}$-concepts w.r.t. generalised role hierarchies.*

2.1 Relationship with Grammar Logics

Grammar logics [Fariñas del Cerro and Penttonen, 1988] are a class of propositional multi modal logics where the accessibility relations are "axiomatised" through a grammar. More precisely, for σ_i, τ_j modal parameters, the production rule $\sigma_1 \ldots \sigma_m \to \tau_1 \ldots \tau_n$ can be viewed as an abbreviation for the axioms

$$[\sigma_1] \ldots [\sigma_m]p \Rightarrow [\tau_1] \ldots [\tau_n]p,$$

or as being a notational variant for the role inclusion axiom

$$\tau_1 \circ \ldots \circ \tau_n \sqsubseteq \sigma_1 \circ \ldots \circ \sigma_m.$$

Analogously to the description logic case, the semantics of a grammar logic is defined by taking into account only those frames/relational structures that "satisfy the grammar".

Now grammars are traditionally organised in (refinements of) the Chomsky hierarchy (see any textbook on formal languages, e,g., [Hopcroft and Ullman, 1997]), which induces also classes of grammar logics, e.g., the class of *context free* grammar logics is the class of those propositional multi modal logics where the accessibility relations are axiomatised through a *context free* grammar. Unsurprisingly, the expressiveness of the grammars influences the expressiveness of the corresponding grammar logics. It was shown that satisfiability of *regular* grammar logics is ExpTime-complete [Demri, 2001], whereas this problem is undecidable for context free grammar logics [Baldoni, 1998; Baldoni *et al.*, 1998]. The latter result is closely related to the undecidability proof in [Wessel, 2001]. In this paper, we are concerned with

- grammars that are *not* regular, but we do not allow for arbitrary context-free grammars (or any known normal forms thereof), and
- multi modal logics that provide a converse operator on modal parameters. I.e., for σ a modal parameter, both $[\sigma]\varphi$ and $[\sigma^-]\varphi$ are formulae of our logic. Moreover, $\mathcal{SH}^+\mathcal{IQ}$ provides *graded* modalities that restrict the number of accessible worlds, see, e.g., [Tobies, 2001; Kupferman *et al.*, 2002].

For example, in our undecidability proof in Section 3, the main difficulty was to develop a grammar that generates the language $\{(ab)^n(cd)^n \mid n \geq 0\}$ where each production is of the form $R \to RS$ or $R \to SR$. We were not able to construct such a grammar, but used a grammar G such that the language generated by G, when intersected with $(ab)^*(cd)^*$, equals $\{(ab)^n(cd)^n \mid n \geq 0\}$. This grammar G contains the four production rules

$$\begin{array}{lcl} D & \to & AD, \\ A & \to & AC, \\ C & \to & BC, \\ B & \to & BD, \quad A \to a, \ldots D \to d \end{array}$$

and can be found in four versions as the last axioms of $\mathcal{R}_\mathcal{D}$ in Figure 2, where we use x_i, y_i, and their inverses instead of $A, \ldots, B$.

2.2 Role value maps

The role inclusion axioms we investigate here are closely related to *role value maps* [Brachman and Schmolze, 1985; Schmidt-Schauss, 1989], i.e., concepts of the form $R_1 \ldots R_m \dot{\sqsubseteq} S_1 \ldots S_n$ for R_i, S_i roles. The semantics of these concepts is defined as follows:

$$(R_1 \ldots R_m \dot{\sqsubseteq} S_1 \ldots S_n)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid (R_1 \ldots R_m)^\mathcal{I}(x) \subseteq (S_1 \ldots S_n)^\mathcal{I}(x)\},$$

where $(R_1 \ldots R_m)^\mathcal{I}(x)$ denotes the set of those $y \in \Delta^\mathcal{I}$ that are reachable from x via $R_1^\mathcal{I} \circ \ldots \circ R_m^\mathcal{I}$.

Thus the role inclusion axioms $R \circ S \sqsubseteq T$ is equivalent to the general concept inclusion axiom $\top \sqsubseteq (RS \dot{\sqsubseteq} T)$, i.e., both axioms have the same models. The role value maps used to show the undecidability of KL-ONE [Schmidt-Schauss, 1989] are of a more general form than $(RS \dot{\sqsubseteq} T)$, i.e., it uses role chains of unbounded length on both sides of the $\dot{\sqsubseteq}$, and hence there is no straightforward translation of the undecidability proof in [Schmidt-Schauss, 1989] to our logic.

3 $\mathcal{SH}^+\mathcal{IQ}$ is undecidable

Due to the syntactic restriction on role inclusion axioms, neither the undecidability proof for $\mathcal{ALC}$ with context-free or linear grammars in [Baldoni, 1998; Baldoni *et al.*, 1998; Demri, 2001] nor the one for $\mathcal{ALC}$ with role boxes [Wessel, 2001] can be adapted to prove undecidability of $\mathcal{SH}^+\mathcal{IQ}$ satisfiability. In the following, we reduce the (undecidable) domino problem [Berger, 1966] to $\mathcal{SH}^+\mathcal{IQ}$ satisfiability.

This problem asks whether, for a set of domino types, there exists a *tiling* of an $\mathbb{N}^2$ grid such that each point of the grid is covered with exactly one of the domino types, and adjacent dominoes are "compatible" with respect to some predefined criteria.

Definition 3 *A domino system $\mathcal{D} = (D, H, V)$ consists of a non-empty set of domino types $D = \{D_1, \ldots, D_n\}$, and of sets of horizontally and vertically matching pairs $H \subseteq D \times D$ and $V \subseteq D \times D$. The problem is to determine if, for a given $\mathcal{D}$, there exists a* tiling *of an $\mathbb{N} \times \mathbb{N}$ grid such that each point of the grid is covered with a domino type in D and all horizontally and vertically adjacent pairs of domino types are in H and V respectively, i.e., a mapping $t : \mathbb{N} \times \mathbb{N} \to D$ such that for all $m, n \in \mathbb{N}$, $\langle t(m,n), t(m+1,n)\rangle \in H$ and $\langle t(m,n), t(m,n+1)\rangle \in V$.*

Given a domino system $\mathcal{D}$, it is undecidable whether a tiling for $\mathcal{D}$ exists [Berger, 1966].

In Figure 2, for a domino system $\mathcal{D}$, we define a $\mathcal{SH}^+\mathcal{IQ}$-concept $C_\mathcal{D}$, a terminology $\mathcal{T}_D$ (that can be internalised, see Theorem 1), and a generalised role hierarchy $\mathcal{R}_D$ such that $\mathcal{D}$ has a tiling iff $C_\mathcal{D}$ is satisfiable w.r.t. $\mathcal{R}_\mathcal{D}$ and $\mathcal{T}_\mathcal{D}$. For a better readability, we use $C \Rightarrow D$ as an abbreviation for $\neg C \sqcup D$.

Ensuring that a point is associated with exactly one domino type, that it has at most one vertical and at most one horizontal successor, and that these successors satisfy the horizontal and vertical matching conditions induced by H and V is standard and is done in the first GCI of $\mathcal{T}_\mathcal{D}$.

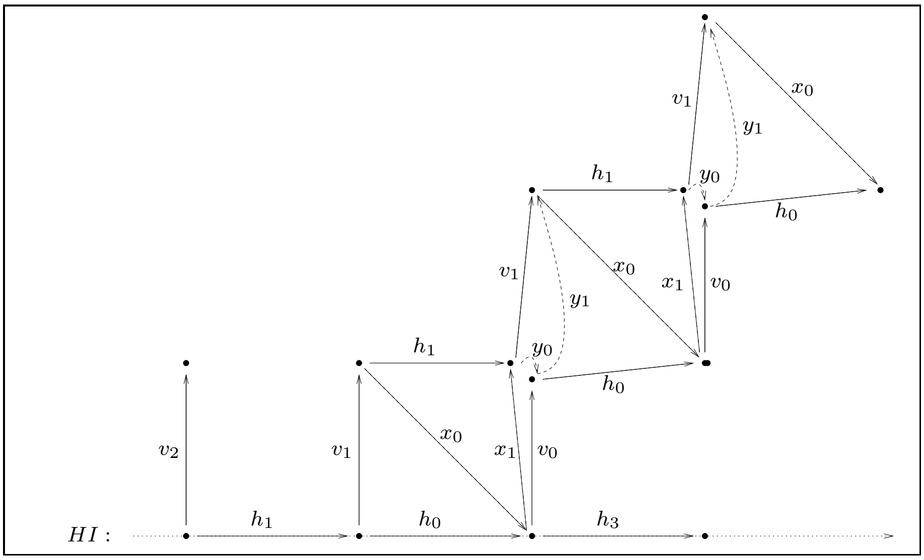

Figure 1: The staircase model structure and the effects of the last 16 axioms in $\mathcal{R}_\mathcal{D}$.

The next step is rather special: we do not enforce a grid structure, but a structure with "staircases", which is illustrated in Figure 1. To this purpose, we introduce four sub-roles $v_0, \ldots, v_3$ of v and four sub-roles $h_0, \ldots, h_3$ of h, and ensure that we only have "staircases". An i-staircase is an alternating chain of v_i and h_i edges, without any other v_j- or h_j-successors. At each point on the x-axis, two staircases start that need not meet again, one i-staircase starting with v_i and one $i \ominus 1$-staircase starting with $h_{i\ominus 1}$ (we use $\oplus$ and $\ominus$ to denote addition and subtraction modulo four). We use a concept HI for those points on the x-axis, VI for those nodes on the y-axis, and enforce a symmetric behaviour for the nodes on the y-axis. The second GCI in $\mathcal{T}_\mathcal{D}$ introduces the concept I for all "initial" points, and then the third GCI enforces the staircase structure. It contains four implications: one for the vertical and one for the horizontal successorships, and these two implications once for the "non-initial" points (i.e., instances of $\neg I$), and once for the "initial points" (i.e., instances of HI or VI).

It remains to make sure that two elements b, b' representing the same point in the grid have the same domino type associated, where b and b' "represent the same point" if there is an n and an instance a of I such that each of them is reachable following a staircase starting at a for n steps, i.e., if there is

- a $v_i h_i$-path (resp. $h_i v_i$-path) of length $2n$ from a to b, and
- a $h_{i\ominus 1} v_{i\ominus 1}$-path (resp. $v_{i\oplus 1} h_{i\oplus 1}$-path) of length $2n$ from a to b'.

To this purpose, we add super roles x_i of h_i and y_i of v_i (for which we use dashed arrows in Figure 1), and the last group of role inclusion axioms in $\mathcal{R}_\mathcal{D}$. These role inclusion axioms enforce appropriate, additional role successorships between elements, and we use the additional roles x_i and y_i since we only want to have at most one v_i or h_i-successor. For each 2 staircases starting at the same element on one of the axes, these role inclusions ensure that each pair of elements representing the same point is related by y_i. That is, each element on an $i \oplus 1$-staircase that is an $x_{i\oplus 1}$-successor is related via y_i to the element on the i-staircase (which is a v_i-successor) representing the same point, see Figure 1. To see this, start considering the consequences of the role inclusion axioms for elements neighbouring instances of, say, HI representing the four points $(1,0),\ldots,(2,1)$, and start with the last but first axiom. Next, "apply" the last but second, and finally the last but third one. Then, starting with the last role inclusion axiom, consider elements representing the four points $(2,1),\ldots,(3,2)$, and continue to work up the role inclusion axioms and up the staircase.

Hence the last GCI in $\mathcal{T}_\mathcal{D}$ ensures that two elements representing the same points in the grid have indeed the same domino type associated.

The above observations imply that the concept $C_\mathcal{D}$ is satisfiable w.r.t. $\mathcal{T}_\mathcal{D}$ and $\mathcal{R}_\mathcal{D}$ iff $\mathcal{D}$ has a solution. Hence, together with Theorem 1, we have the following:

Theorem 2 *Satisfiability of $\mathcal{SH}^+\mathcal{IQ}$-concepts w.r.t. generalized role hierarchies is undecidable.*

4 $\mathcal{RIQ}$ is decidable

In this section, we show that $\mathcal{SHIQ}$ with *acyclic* generalised role hierarchies is decidable. We present a tableau-based algorithm that decides satisfiability

$$
\begin{array}{lll}
C_{\mathcal{D}} & := & HI \sqcap VI \sqcap \exists h_0.HI \sqcap \exists v_1.VI \\
\mathcal{T}_{\mathcal{D}} & := \{ & \top \doteq (\bigsqcup_{1\leq i\leq n} D_i) \sqcap (\bigsqcap_{1\leq i<j\leq n} \neg(D_i \sqcap D_j)) \sqcap \\
& & \quad \bigsqcap_{1\leq i\leq n} D_i \Rightarrow ((\leqslant 1 v.\top) \sqcap (\forall v. \bigsqcup_{(D_i,D_j)\in V} D_j)) \sqcap \\
& & \quad \bigsqcap_{1\leq i\leq n} D_i \Rightarrow ((\leqslant 1 h.\top) \sqcap (\forall h. \bigsqcup_{(D_i,D_j)\in H})D_j)\} \\
& & I \doteq HI \sqcup VI \\
& & \top \doteq \bigsqcap_{0\leq i\leq 3} \ (\exists v_i^-.\top \sqcap \neg I) \Rightarrow \big(\exists h_i.\neg I \sqcap \bigsqcap_{j} \forall v_j.\bot \sqcap \bigsqcap_{j\neq i} \forall h_j.\bot\big)\big)\sqcap \\
& & \qquad (\exists h_i^-.\top \sqcap \neg I) \Rightarrow \big(\exists v_i.\neg I \sqcap \bigsqcap_{j\neq i} \forall v_j.\bot \sqcap \bigsqcap_{j} \forall h_j.\bot\big)\big)\sqcap \\
& & \qquad (\exists h_i^-.\top \sqcap HI) \Rightarrow \big(\exists v_i.\neg I \sqcap \exists h_{i\ominus 1}.HI \sqcap \\
& & \qquad\qquad \bigsqcap_{j\neq i\ominus 1} \forall h_j.\bot \sqcap \bigsqcap_{j\neq i} \forall v_j.\bot\big)\sqcap \\
& & \qquad (\exists v_i^-.\top \sqcap VI) \Rightarrow \big(\exists h_i.\neg I \sqcap \exists v_{i\oplus 1}.VI \sqcap \\
& & \qquad\qquad \bigsqcap_{j\neq i\oplus 1} \forall v_j.\bot \sqcap \bigsqcap_{j\neq i} \forall h_j.\bot\big)\sqcap \\
& & \top \doteq \bigsqcap_{0\leq i\leq 3} \bigsqcap_{1\leq j\leq n} \exists x_{i\oplus 1}^-.\top \Rightarrow (D_j \Rightarrow \forall y_i.D_j) \\
\mathcal{R}_{\mathcal{D}} & := & \{v_i \sqsubseteq y_i,\ v_i \sqsubseteq v,\ h_i \sqsubseteq x_i,\ h_i \sqsubseteq h \mid 0 \leq i \leq 3\} \cup \\
& & \{\ x_{i\oplus 1}^- y_i \ \sqsubseteq \ y_i \\
& & \quad x_{i\oplus 1}^- x_i \ \sqsubseteq \ x_{i\oplus 1}^- \\
& & \quad y_{i\oplus 1}^- x_i \ \sqsubseteq \ x_i \\
& & \quad y_{i\oplus 1}^- y_i \ \sqsubseteq \ y_{i\oplus 1}^- \mid 0 \leq i \leq 3\}
\end{array}
$$

Figure 2: Reduction terminology, generalised role hierarchy, and concept.

of $\mathcal{RIQ}$-concepts w.r.t. *acyclic* generalised role hierarchies, and therefore also subsumption in $\mathcal{RIQ}$ and, with Theorem 1, both inferences w.r.t. terminologies. The tableau algorithm implemented in the FaCT system [Horrocks, 1998] was extended to the one presented here, and the empirical results are reported in Section 6.

The algorithm tries to construct, for a $\mathcal{RIQ}$-concept C, a *tableau* for C, that is, an abstraction of a model of C. Given the appropriate notion of a tableau, it is then quite straightforward to prove that the algorithm is a decision procedure for $\mathcal{RIQ}$-satisfiability. Before specifying this algorithm, we transform the role hierarchy to make the presentation of the algorithm easier—basically, we *unfold* or *expand* the role hierarchy to make all implications explicit.

We start with a definition of *acyclic* generalised role hierarchies, an explicit form of generalised role hierarchies obtained by using a form of unfolding, corresponding finite automata, and then finally, a closure of concepts w.r.t. role hierarchies.

Definition 4 *Let $\mathcal{R}$ be a generalised role hierarchy (containing $R_1 \circ R_2 \sqsubseteq R_3$ iff it contains $\mathsf{Inv}(R_2) \circ \mathsf{Inv}(R_1) \sqsubseteq \mathsf{Inv}(R_3)$, and containing $R \sqsubseteq S$ iff it contains $\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S)$; see above). A role R* directly affects *a role S if $R \neq S$ and*

- *$R \sqsubseteq S \in \mathcal{R}$,*
- *$R \circ S \sqsubseteq S \in \mathcal{R}$, or*
- *$S \circ R \sqsubseteq S \in \mathcal{R}$.*

Let "affects" *be the transitive closure of "directly affects". We call a role that is not affected by other roles* unaffected. *A generalised role hierarchy is* acyclic *if "affects" has no cycles, i.e., if, for all roles R, R does not affect R.*

Please note that, w.l.o.g., we can assume that $\overset{*}{\sqsubseteq}$ is acyclic: in case $\mathcal{R}$ contains $\overset{*}{\sqsubseteq}$ cycles, we can simply choose one role name R from each cycle and replace all other role names on this cycle with R, both in the input role hierarchy and the input concept.

Please note also that, in acyclic role hierarchies, we can no longer say that a role R is symmetric using $R \sqsubseteq R^-$ and $R^- \sqsubseteq R$ since this would yield an "affects" cycle of length 2.

4.1 Syntactic transformations

In a first step, we unfold an acyclic generalised role hierarchy $\mathcal{R}$ into an explicit form called $\exp(\mathcal{R})$ as follows:

- First, for each role R occurring in $\mathcal{R}$, define
$$\tau_R := (\bigcup_{\substack{S \circ R \sqsubseteq R \in \mathcal{R} \\ S \neq R}} S)^* R (\bigcup_{\substack{R \circ T \sqsubseteq R \in \mathcal{R} \\ T \neq R}} T)^*.$$
- Secondly, set
$$\rho_R := \begin{cases} \tau_R & \text{if } R \circ R \sqsubseteq R \notin \mathcal{R} \\ (\tau_R)^+ & \text{if } R \circ R \sqsubseteq R \in \mathcal{R}. \end{cases}$$
- In the third step, we iteratively replace roles in ρ_R with unions of regular expressions of roles, working our way up the affecting relation. We start with roles S which are "almost" minimal w.r.t. affected, i.e., that are affected only by *unaffected* roles. We proceed with roles directly affected by roles that are already treated or unaffected and do the following:
$$\begin{array}{rcl} \rho_R & := & (\rho_R \text{ with } R \text{ replaced with } R \cup \bigcup_{\substack{P \overset{*}{\sqsubseteq} R \\ P \neq R}} \rho_P) \text{ and} \\ \multicolumn{3}{l}{\text{for each } S \neq R \text{ occurring in } \rho_R \text{ do}} \\ \rho_R & := & (\rho_R \text{ with } S \text{ replaced with } \bigcup_{P \overset{*}{\sqsubseteq} S} \rho_P). \end{array}$$

  After this recursion, we set $\exp(\mathcal{R}) := \{\rho_R \sqsubseteq R \mid R \text{ occurs in } \mathcal{R}\}$.

Due to the acyclicity of $\mathcal{R}$, the recursion in this transformation terminates after at most n steps for n the number of role inclusion axioms in $\mathcal{R}$. Please note that, by construction, for each (possibly inverse) role R occurring in $\mathcal{R}$, $\exp(\mathcal{R})$ contains exactly one inclusion $\rho_R \sqsubseteq R$.

Let us first consider an example. Given the role inclusion axioms $\mathcal{R}$ consisting of

$$\begin{array}{ll} R \circ S & \sqsubseteq S, \\ S \circ W & \sqsubseteq S, \\ T_1 \circ R_1 & \sqsubseteq R_1, \\ R_2 \circ T_2 & \sqsubseteq R_2, \\ V \circ T_1 & \sqsubseteq T_1 \end{array}$$

with $R_1, R_2 \mathrel{\underline{\ast}} R$, the above transformation yield a set $\exp(\mathcal{R})$ consisting of

$$\begin{array}{rl} (R \cup R_2 T_2^* \cup (V^* T_1)^* R_1)^* S W^* & \sqsubseteq S, \\ R \cup R_2 T_2^* \cup (V^* T_1)^* R_1 & \sqsubseteq R, \\ R_2 T_2^* \sqsubseteq R_2, \quad (V^* T_1)^* R_1 \sqsubseteq R_1, \quad V^* T_1 & \sqsubseteq T_1. \end{array}$$

Unfortunately, the size of $\exp(\mathcal{R})$ can be exponential in the size of $\mathcal{R}$. For $n \in \mathbb{N}$, let $\mathcal{R}_n$ be the following acyclic generalised role hierarchy:

$$\begin{array}{rclcrcl} S_n \circ R_n & \sqsubseteq & R_n & & R_n \circ S_n & \sqsubseteq & R_n \\ R_{n-1} & \sqsubseteq & S_n & & & & \\ S_{n-1} \circ R_{n-1} & \sqsubseteq & R_{n-1} & & R_{n-1} \circ S_{n-1} & \sqsubseteq & R_{n-1} \\ R_{n-2} & \sqsubseteq & S_{n-1} & & & & \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ R_1 & \sqsubseteq & S_2 & & & & \\ S_1 \circ R_1 & \sqsubseteq & R_1 & & R_1 \circ S_1 & \sqsubseteq & R_1 \end{array}$$

and it is easily checked that the size of ρ_n is exponential in n. A further syntactic restriction which prohibits this exponential blow-up is described in Section 5.

The regular role terms on the left hand side of $\exp(\mathcal{R})$ are then read with the standard semantics for regular role expressions, (i.e., using union, composition, and transitive closure of binary relations, see, e.g., [Schild, 1991]). We use $L(\rho)$ to denote the language described by a regular expression ρ. By definition of $\exp(\mathcal{R})$, we have the following Lemma:

Lemma 2 *(i) For each $\rho_R \sqsubseteq R \in \exp(\mathcal{R})$ we have $R \in L(\rho_R)$.*
(ii) If $R \circ S \sqsubseteq S \in \mathcal{R}$, then $RS \in L(\rho_P)$ for all P with $S \mathrel{\underline{\ast}} P$.
(iii) If $S \circ R \sqsubseteq S \in \mathcal{R}$, then $SR \in L(\rho_P)$ for all P with $S \mathrel{\underline{\ast}} P$.
(iv) The size $|\exp(\mathcal{R})| := \sum_{\rho \sqsubseteq R \in \exp(\mathcal{R})} |\rho|$ of $\exp(\mathcal{R})$ is at most exponential in the number of role inclusion axioms in $\mathcal{R}$.

Proof: (i) is obvious since ρ_R is of the form

$$(\ldots)^*(\ldots \cup R \cup \ldots)(\ldots)^* \text{ or } ((\ldots)^*(\ldots \cup R \cup \ldots)(\ldots)^*)^+,$$

and thus we have $R \in L(\rho_R)$.

For (ii), let $R \circ S \sqsubseteq S \in \mathcal{R}$ and $S \,\underline{\circledast}\, P$. Hence after the second step, we have

ρ_S is of the form	$(\ldots \cup R \cup \ldots)^* S (\ldots)^*$	or	$((\ldots \cup R \cup \ldots)^* S (\ldots)^*)^+$
ρ_P is of the form	$(\ldots)^* P (\ldots)^*$	or	$((\ldots)^* P (\ldots)^*)^+$

In the third step, we replace R with ρ_R in ρ_S and, by (i), we have $R \in L(\rho_R)$. Again, by (i), we have $S \in L(\rho_S)$, and thus $RS \in L(\rho_S)$. Now, since $S \,\underline{\circledast}\, P$, S affects P, and we replace P with $P \cup \rho_S \cup \ldots$ in ρ_P, which thus yields $RS \in L(\rho_P)$.

(iii) is symmetric to (ii).

(iv) is a simple consequence of the fact that a tree whose depth and breadth are bounded by n has at most exponentially many nodes in n. Due to acyclicity of $\mathcal{R}$, the term tree of each ρ_R is of breadth and depth bounded by the number of axioms in $\mathcal{R}$. ∎

It remains to prove that this transformation preserves the semantics, which is defined as follows: an interpretation $\mathcal{I}$ is a *model* of an explicit role hierarchy $\exp(\mathcal{R})$ if

$$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}} \text{ for each } \rho_R \sqsubseteq R \in \exp(\mathcal{R}) \text{ and each } R_1 \ldots R_n \in L(\rho_R), \quad (1)$$

where $\circ$ denotes standard composition of binary relations.

Lemma 3 *An interpretation $\mathcal{I}$ is a model of an acyclic generalised role hierarchy $\mathcal{R}$ iff $\mathcal{I}$ is a model of* $\exp(\mathcal{R})$.

Proof: "$\Leftarrow$": let $R \sqsubseteq S \in \mathcal{R}$ and let $\mathcal{I}$ be a model of $\exp(\mathcal{R})$. Due to Lemma 2, we have that $R \in L(\rho_R)$. Moreover, in the iterative substitution, we have replaced S in ρ_S with $\ldots \cup \rho_R \cup \ldots$, and thus $R \in L(\rho_S)$ and $\rho_S \sqsubseteq S \in \exp(\mathcal{R})$. Hence $\mathcal{I}$ satisfies $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.

Now let $R \circ S \sqsubseteq S \in \mathcal{R}$ and let $\mathcal{I}$ be a model of $\exp(\mathcal{R})$. By Lemma 2, $RS \in L(\rho_S)$, and thus $R^{\mathcal{I}} \circ S^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ by definition of the semantics. The case for $R \circ S \sqsubseteq R \in \mathcal{R}$ is analogous.

"$\Rightarrow$": Let $\mathcal{I}$ be a model of $\mathcal{R}$. First, it is easy to see that $\mathcal{I}$ satisfies $\rho_R \sqsubseteq R$ for each ρ_R after the second step. Next, we prove by induction on the number of substitutions carried out during the computation of $\exp(\mathcal{R})$ in the third step (which can be viewed as a refined induction on "affecting") that $\mathcal{I}$ is also a model of $\exp(\mathcal{R})$.

Thus, assume that $\mathcal{I}$ satisfies all axioms $\rho_S \sqsubseteq S$ already computed, and consider the computation of $\rho_R \sqsubseteq R$ in the third step of the syntactic transformation for R a role with $R \circ R \sqsubseteq R \notin \mathcal{R}$.

- In the first part of the third step, we substitute R with $R \cup \bigcup_{\substack{P \,\underline{\circledast}\, R \\ P \neq R}} \rho_P$ in ρ_R. Let $R_1 \ldots R_n \in L(\rho_R)$ and $\langle x, y \rangle \in (R_1 \ldots R_n)^{\mathcal{I}}$. If $R_1 \ldots R_n \in L(\rho_R)$ for ρ_R after the second step, then induction implies $\langle x, y \rangle \in R^{\mathcal{I}}$. Otherwise, we can split $R_1 \ldots R_n$ into uvw such that $uRw \in L(\rho_R)$ and $v \in L(\rho_P)$ for some $P \neq R$ with $P \,\underline{\circledast}\, R$. Since P affects R, ρ_P is already computed and, by induction, $\mathcal{I}$ satisfies $\rho_P \sqsubseteq P$, and thus $v^{\mathcal{I}} \subseteq P^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. Hence

$\langle x, y\rangle \in (uRw)^{\mathcal{I}}$ and, since $uRw \in L(\rho_R)$, this implies $\langle x, y\rangle \in R^{\mathcal{I}}$ by induction.

- Call the result of the first part of the third step $\tilde{\rho}_R$. In the second part, consider the substitution of some S in $\tilde{\rho}_R$ with $\bigcup_{P \, \underline{\sqsubseteq}^* S} \rho_P$, and call the result of this substitution ρ_R. Let $R_1 \dots R_n \in L(\rho_R)$ and $\langle x, y\rangle \in (R_1 \dots R_n)^{\mathcal{I}}$. Again, if $R_1 \dots R_n \in L(\tilde{\rho}_R)$, then induction implies $\langle x, y\rangle \in R^{\mathcal{I}}$. Otherwise, we can split $R_1 \dots R_n$ into uv_1xv_2w such that $uSxSw \in L(\rho_R)$ and $v_i \in L(\bigcup_{P \, \underline{\sqsubseteq}^* S} \rho_P)$. By induction, $v_i^{\mathcal{I}} \subseteq S^{\mathcal{I}}$, and thus $\langle x, y\rangle \in (uSxSw)^{\mathcal{I}}$ which, since $uSxSw \in L(\tilde{\rho}_R)$, implies by induction that $\langle x, y\rangle \in R^{\mathcal{I}}$.

The argumentation for roles R with $R \circ R \sqsubseteq R \in \mathcal{R}$ is analogous. ∎

4.2 A Tableau for $\mathcal{RIQ}$

In the following, if not stated otherwise, C, D (possibly with subscripts) denote $\mathcal{RIQ}$-concepts, R, S (possibly with subscripts) roles, and $\mathcal{R}$ an acyclic generalised role hierarchy.

We start by defining $\mathsf{fclos}(C_0, \mathcal{R})$, the *closure* of a concept C w.r.t. an acyclic generalised role hierarchy $\mathcal{R}$. Intuitively, this contains all relevant sub-concepts of C together with universal value restrictions over sets of roles paths described by nondeterministic finite automata (NFA). These NFAs are used to monitor the effect of $\forall R.C$ sub-concepts along paths in the tree model.

Let Σ be the alphabet of roles (role names and inverse role names) in $\exp(\mathcal{R})$. We use $L(\rho)$ to denote the (regular) language described by a regular expression ρ and ε to denote the empty word. For each (possibly inverse) role R occurring in C_0 or $\mathcal{R}$, we define $\mathcal{A}^R$ as follows:

- if R occurs in $\mathcal{R}$, then $\rho \sqsubseteq R \in \exp(\mathcal{R})$, and $\mathcal{A}^R$ is an NFA with $L(\mathcal{A}^R) = L(\rho)$. Due to the use of non-deterministic automata, $\mathcal{A}^R$ can be constructed in size linear in $|\rho_R|$.
- otherwise, $\mathcal{A}^R$ is a (two-state) automaton with $L(\mathcal{A}^R) = \{R\}$.

Next, for $\mathcal{A}$ an NFA and q a state in $\mathcal{A}$, $\mathcal{A}_q$ denotes the NFA obtained from $\mathcal{A}$ by making q the (only) initial state of $\mathcal{A}$, and we use $q \xrightarrow{S} q' \in \mathcal{A}$ to denote that $\mathcal{A}$ has a path labelled with S from q to q'.

Without loss of generality, we assume all concepts to be in NNF, that is, negation occurs in front of concept names only. Any $\mathcal{RIQ}$-concept can easily be transformed into an equivalent one in NNF by pushing negations in-wards using a combination of DeMorgan's laws and the following equivalences:

$$\begin{array}{rclrcl}
\neg(\exists R.C) & \equiv & (\forall R.\neg C) & \neg(\forall R.C) & \equiv & (\exists R.\neg C) \\
\neg(\leqslant nR.C) & \equiv & (\geqslant(n+1)R.C) & \neg(\geqslant(n+1)R.C) & \equiv & (\leqslant nR.C) \\
 & & & \neg(\geqslant 0R.C) & \equiv & A \sqcap \neg A \quad \text{for some } A \in \mathbf{C}
\end{array}$$

We use $\dot{\neg} C$ for the NNF of $\neg C$. Obviously, the length of $\dot{\neg} C$ is linear in the length of C.

For a concept C_0, $\mathsf{clos}(C_0)$ is the smallest set that contains C and that is closed under sub-concepts and $\dot\neg$. The set $\mathsf{fclos}(C_0, \mathcal{R})$ is then defined as follows:

$$\begin{aligned}\mathsf{fclos}(C_0, \mathcal{R}) := \ & \mathsf{clos}(C_0) \cup \\ & \{\forall \mathcal{A}_q^S.D \mid S \text{ occurs in } \mathcal{R} \text{ or } C_0,\ q \text{ is a state in } \mathcal{A}^S, \text{ and} \\ & \qquad \forall S.D \in \mathsf{clos}(C_0)\}\end{aligned}$$

It is not hard to show and well-known that the size of $\mathsf{clos}(C_0)$ is polynomial in the size of C_0. The size of $\mathsf{fclos}(\mathcal{A})$ is more involved: each ρ with $\rho \sqsubseteq R \in \exp(\mathcal{R})$ is a regular expression whose size is at most exponential in the size of $\mathcal{R}$ (see Lemma 2), and the number of such expressions in $\exp(\mathcal{R})$ is linear in $|\mathcal{R}|$. The construction of a *non-deterministic* automaton $\mathcal{A}_\rho$ from a regular expression ρ yields an NFA linear in the size of ρ, and thus we have an exponential bound for the cardinality of $\mathsf{fclos}(C_0, \mathcal{R})$ in the size of C_0 and $\mathcal{R}$. Investigating the size of $\mathsf{fclos}(C_0, \mathcal{R})$ more closely and deciding whether this exponential blow-up can be avoided will be a part of future work. So far, we only define in the Section 5 a further syntactic restriction which avoids this exponential blow-up.

We are now ready to define tableaux as a useful abstraction of models.

Definition 5 *$T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ is a* tableau *for D w.r.t. $\mathcal{R}$ iff*

- $\mathbf{S}$ *is a non-empty set,*
- $\mathcal{L} : \mathbf{S} \to 2^{\mathsf{fclos}(\mathcal{A})}$ *maps each element in $\mathbf{S}$ to a set of concepts and*
- $\mathcal{E} : \mathbf{R}_\mathcal{A} \to 2^{\mathbf{S} \times \mathbf{S}}$ *maps each role to a set of pairs of elements in $\mathbf{S}$.*

Furthermore, for all $s, t \in \mathbf{S}$, $C, C_1, C_2 \in \mathsf{fclos}(\mathcal{A})$, and $R, S \in \mathbf{R}_\mathcal{A}$, T satisfies:

(P0) there is some $s \in \mathbf{S}$ with $D \in \mathcal{L}(s)$,

(P1) if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$,

(P2) if $C_1 \sqcap C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ and $C_2 \in \mathcal{L}(s)$,

(P3) if $C_1 \sqcup C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ or $C_2 \in \mathcal{L}(s)$,

(P4a) if $\forall \mathcal{A}_p.C \in \mathcal{L}(s)$, $\langle s, t\rangle \in \mathcal{E}(S)$, and $p \xrightarrow{S} q \in \mathcal{A}_p$, then $\forall \mathcal{A}_q.C \in \mathcal{L}(t)$,

(P4b) if $\forall \mathcal{A}_p.C \in \mathcal{L}(s)$ and $\varepsilon \in L(A_p)$, then $C \in \mathcal{L}(s)$,

(P5) if $\exists S.C \in \mathcal{L}(s)$, then there is some t with $\langle s, t\rangle \in \mathcal{E}(S)$ and $C \in \mathcal{L}(t)$,

(P6) if $\forall S.C \in \mathcal{L}(s)$, then $\forall \mathcal{A}^S.C \in \mathcal{L}(s)$,

(P7) $\langle x, y\rangle \in \mathcal{E}(R)$ iff $\langle y, x\rangle \in \mathcal{E}(\mathsf{Inv}(R))$,

(P8) if $({\leqslant} nS.C) \in \mathcal{L}(s)$, then $\sharp S^T(s, C) \leqslant n$,

(P9) if $({\geqslant} nS.C) \in \mathcal{L}(s)$, then $\sharp S^T(s, C) \geqslant n$,

(P10) if $({\leqslant} nS.C) \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(S)$ then $C \in \mathcal{L}(t)$ or $\dot\neg C \in \mathcal{L}(t)$,

where $\bowtie$ is a place-holder for either $\leqslant$ or $\geqslant$, and

$$S^T(s, C) := \{t \in \mathbf{S} \mid \langle s, t\rangle \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\}.$$

Lemma 4 *A $\mathcal{RIQ}$-concept D is satisfiable w.r.t. $\mathcal{R}$ iff there exists a tableau for D w.r.t. $\mathcal{R}$.*

Proof: For the *if* direction, let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ be a tableau for D w.r.t. $\mathcal{R}$. We extend the relational structure of T and then prove that this indeed gives a model. More precisely, a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of D and $\mathcal{R}$ can be defined as follows:

$$
\begin{array}{lll}
\Delta^{\mathcal{I}} & := & \mathbf{S} \\
\multicolumn{3}{l}{\text{for concept names A in } \mathsf{clos}(\mathcal{A}):} \\
A^{\mathcal{I}} & := & \{s \mid A \in \mathcal{L}(s)\} \\
\multicolumn{3}{l}{\text{for roles names } R:} \\
R^{\mathcal{I}} & := & \{\langle s,t\rangle \in \mathcal{E}(R)\} \cup \\
 & & \{\langle s_0, s_n\rangle \in (\Delta^{\mathcal{I}})^2 \mid \begin{array}[t]{l} \rho \sqsubseteq R \in \exp(\mathcal{R}) \text{ and there is} \\ s_1, \ldots, s_{n-1} \text{ with } \langle s_i, s_{i+1}\rangle \in \mathcal{E}(S_{i+1}) \text{ for} \\ 0 \leq i \leq n-1 \text{ and } S_1 \cdots S_n \in L(\rho)\} \end{array}
\end{array}
$$

The definition of inverse roles and complex concepts is given through the definition of the $\mathcal{SH}^{+}\mathcal{IQ}$ semantics. Due to (P7) and the fact that we only consider acyclic generalised role hierarchies containing both directions of axioms of the form $R \circ S \sqsubseteq S$ (see the remark immediately above Definition 2), we can indeed treat role names and inverse roles symmetrically.

We have to show that $\mathcal{I}$ is a model of $\mathcal{R}$ and D. Due to Lemma 3, it suffices to prove that $\mathcal{I}$ satisfies $\exp(\mathcal{R})$.

$\mathcal{I}$ is a model of $\exp(\mathcal{R})$: Let $\rho \sqsubseteq R \in \exp(\mathcal{R})$ and $\langle s,t\rangle \in (R_1 \ldots R_n)^{\mathcal{I}}$ for some $R_1 \ldots R_n \in L(\rho)$. Then, by definition of $R^{\mathcal{I}}$, we have $\langle s,t\rangle \in R^{\mathcal{I}}$, and thus $\mathcal{I}$ satisfies $\exp(\mathcal{R})$.

$\mathcal{I}$ is a model of D: To prove that $\mathcal{I}$ is a model of D, we show that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$ for any $s \in \mathbf{S}$. Together with (P0), this implies that $\mathcal{I}$ is a model of D. This proof can be given by induction on the length $\|C\|$ of a concept $C \in \mathsf{clos}(D)$ in NNF, where we count neither negation nor integers in number restrictions. The only interesting cases are $C = (\leqslant nr.E)$ and $C = \forall S.E$ (for the others, see [Horrocks *et al.*, 2000; Horrocks and Sattler, 2002]):

- If $(\leqslant nS.E) \in \mathcal{L}(s)$, then S is by definition a simple role, i.e., neither S nor any of its sub-roles (or their inverse) occur on the right hand of an axiom of the form $R_1 \circ R_2 \sqsubseteq R_3$. Hence $S^{\mathcal{I}} = \mathcal{E}(S)$, and thus, by (P8) and induction, we have $s \in (\leqslant nS.E)^{\mathcal{I}}$.
- Let $\forall S.E \in \mathcal{L}(s)$ and $\langle s,t\rangle \in S^{\mathcal{I}}$. From (P6) we have that $\forall \mathcal{A}^S.E \in \mathcal{L}(s)$.
  - If $\langle s,t\rangle \in \mathcal{E}(S)$, then (P4a) and $S \in L(\mathcal{A}^S)$ (either since $L(\mathcal{A}^S) = \{S\}$ or by Lemma 2.(i)) imply that $\forall \mathcal{A}^S_q.E \in \mathcal{L}(t)$ with $\varepsilon \in L(\mathcal{A}^S_q)$, and thus (P4b) implies that $E \in \mathcal{L}(t)$.
  - Otherwise, there is some $\rho \sqsubseteq S \in \exp(\mathcal{R})$ and $s_0, \ldots, s_n$ with $s_0 = s$, $s_n = t$, $\langle s_i, s_{i+1}\rangle \in S^{\mathcal{I}}_{i+1}$ for each $0 \leq i \leq n-1$, and $S_1 \cdots S_n \in L(\rho)$. By definition, $L(\rho) = L(\mathcal{A}^S)$, hence (P4a) ($n$ times) implies that $\forall \mathcal{A}.E \in \mathcal{L}(t)$ with $\varepsilon \in L(\mathcal{A})$, and thus (P4b) implies $E \in \mathcal{L}(t)$.

By induction, $t \in E^{\mathcal{I}}$, and thus $s \in (\forall S.E)^{\mathcal{I}}$.

For the converse, for $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model of D w.r.t. $\mathcal{R}$, we define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ for $\mathcal{A}$ and $\mathcal{R}$ as follows:

$$
\begin{array}{rcl}
\mathbf{S} & := & \Delta^{\mathcal{I}}, \\
\mathcal{E}(R) & := & R^{\mathcal{I}}, \text{ and} \\
\mathcal{L}(s) & := & \{C \in \mathsf{clos}(D) \mid s \in C^{\mathcal{I}}\} \cup \\
 & & \{\forall \mathcal{A}^S.C \mid \forall S.C \in \mathsf{clos}(D) \text{ and } s \in (\forall S.C)^{\mathcal{I}}\} \cup \\
 & & \{\forall \mathcal{A}^R_q.C \in \mathsf{fclos}(D, \mathcal{R}) \mid \text{ for all } S_1 \cdots S_n \in L(\mathcal{A}^R_q), \\
 & & \qquad\qquad\qquad\qquad s \in (\forall S_1.\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}\}
\end{array}
$$

We have to show that T satisfies each ($\mathsf{P}i$). We restrict our attention to the only new cases are ($\mathsf{P4}$) and ($\mathsf{P6}$).

For ($\mathsf{P6}$), if $\forall S.C \in \mathcal{L}(s)$, then $s \in (\forall S.C)^{\mathcal{I}}$, and thus the definition of T implies that $\forall \mathcal{A}^S. \in \mathcal{L}(s)$.

For ($\mathsf{P4a}$), let $\forall \mathcal{A}_p.C \in \mathcal{L}(s)$. Hence $s \in (\forall S_1.\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}$ for each $S_1 \cdots S_n \in L(\mathcal{A}_p)$. Now let $\langle s, t\rangle \in \mathcal{E}(S) \subseteq S^{\mathcal{I}}$. Then $t \in (\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}$ if $s \in (\forall S.\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}$. Moreover, if $p \xrightarrow{S} q$ in $\mathcal{A}_p$, then $SS_2 \ldots S_n \in L(\mathcal{A}_p)$ implies $S_2 \ldots S_n \in L(\mathcal{A}_q)$. Hence $t \in (\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}$ if $SS_2 \cdots S_n \in \mathcal{L}(\mathcal{A}_p)$, and thus $p \xrightarrow{S} q$ in $\mathcal{A}_p$ implies that $\forall \mathcal{A}_q.C \in \mathcal{L}(t)$.

For ($\mathsf{P4b}$), if $\forall \mathcal{A}_p.C \in \mathcal{L}(s)$ with $\varepsilon \in L(\mathcal{A}_p)$, then $s \in (\forall \varepsilon.C)^{\mathcal{I}} = C^{\mathcal{I}}$, which implies $C \in \mathcal{L}(s)$ by definition of $\mathcal{L}$.

4.3 The Tableau Algorithm

In this section, we present a completion algorithm that tries to construct, for an input $\mathcal{RIQ}$-concept D and an acyclic generalised role hierarchy $\mathcal{R}$, a tableau for D w.r.t. $\mathcal{R}$. We prove that this algorithm constructs a tableau for D and $\mathcal{R}$ iff there exists a tableau for D and $\mathcal{R}$, and thus decides satisfiability of $\mathcal{RIQ}$ concepts w.r.t. acyclic generalised role hierarchies (and a general terminology).

This algorithm generates a *completion tree*, a structure that will be unravelled to an (infinite) tableau for the input concept. As usual, in the presence of transitive roles, *blocking* is employed to ensure termination of the algorithm. In the additional presence of inverse roles, blocking is *dynamic*, i.e., blocked nodes (and their sub-branches) can be un-blocked and blocked again later. In the further, additional presence of number restrictions, *pairs* of nodes are blocked rather than single nodes [Horrocks *et al.*, 2000; Horrocks and Sattler, 2002].

Definition 6 *A* completion tree $\mathbf{T}$ *for a $\mathcal{RIQ}$ concept D and a acyclic generalised role hierarchy $\mathcal{R}$ is a tree, where each node x is labelled with a set $\mathcal{L}(x) \subseteq \mathsf{fclos}(D, \mathcal{R})$ and each edge $\langle x, y\rangle$ from a node x to its successor y is labelled with a non-empty set $\mathcal{L}(\langle x, y\rangle) \subseteq \mathcal{R}$ of (possibly inverse) roles occurring in D and $\mathcal{R}$. Finally, completion trees come with an explicit inequality relation $\not\doteq$ on nodes which is implicitly assumed to be symmetric.*

*If $R \in \mathcal{L}(\langle x, y\rangle)$ for a node x and its successor y and $R \sqsubseteq^* S$, then y is called an S-*successor *of x and x is called an $\mathsf{Inv}(S)$-*predecessor *of y. If y is an S-successor or an $\mathsf{Inv}(S)$-predecessor of x, then y is called an S-neighbour of x. Finally,* ancestor *is the transitive closure of* predecessor.

For a role S, a concept C and a node x in $\mathbf{T}$ we define $S^{\mathbf{T}}(x, C)$ by

$$S^{\mathbf{T}}(x, C) := \{y \mid y \text{ is an } S\text{-neighbour of } x \text{ and } C \in \mathcal{L}(y)\}.$$

A node is blocked *iff it is either directly or indirectly blocked. A node x is* directly blocked *iff none of its ancestors are blocked, and it has ancestors x', y and y' such that*

1. *y is not the root node* and
2. *x is a successor of x' and y is a successor of y'* and
3. *$\mathcal{L}(x) = \mathcal{L}(y)$ and $\mathcal{L}(x') = \mathcal{L}(y')$* and
4. *$\mathcal{L}(\langle x', x\rangle) = \mathcal{L}(\langle y', y\rangle)$.*

If there are no successors x'', y'' of x' and y' with these properties, then we say that y blocks *x.*

A node y is indirectly blocked *if one of its ancestors is blocked.*

Given a $\mathcal{RIQ}$-concept D (in NNF) and an acyclic generalised role hierarchy $\mathcal{R}$, the algorithm initialises a completion tree $\mathbf{T}_{D,\mathcal{R}}$ consisting only of a root node x_0 labelled with $\{D\}$. Then $\mathbf{T}_{D,\mathcal{R}}$ is expanded by repeatedly applying the rules from Figure 3.

For a node x, $\mathcal{L}(x)$ is said to contain a clash *if, for some concept name A, $\{A, \neg A\} \subseteq \mathcal{L}(x)$, or if there is some concept $(\leqslant nS.C) \in \mathcal{L}(x)$ and x has $n+1$ S-neighbours $y_0, \ldots, y_n$ with $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for all $0 \leq i < j \leq n$. A completion tree is* clash-free *if none of its nodes contains a clash, and it is* complete *if no rule from Figure 3 can be applied to it.*

For a $\mathcal{RIQ}$-concept D, the algorithm starts with the completion tree $\mathbf{T}_{D,\mathcal{R}}$. It applies the expansion rules in Figure 3, stopping when a clash occurs, and answers "D is satisfiable w.r.t. $\mathcal{R}$" iff the completion rules can be applied in such a way that they yield a complete and clash-free completion tree, and "D is unsatisfiable w.r.t. $\mathcal{R}$" otherwise.

All but the $\forall_i$-rules have been used before for fragments of $\mathcal{RIQ}$, and the three $\forall_i$-rules are the obvious counterparts to the tableau conditions (P4) and (P6).

As usual, we prove termination, soundness, and completeness of the tableau algorithm to show that it indeed decides satisfiability of $\mathcal{RIQ}$-concepts w.r.t. acyclic generalised role hierarchies.

Lemma 5 *Let D be a $\mathcal{RIQ}$-concept and $\mathcal{R}$ an acyclic generalised role hierarchy. The completion algorithm terminates when started for D and $\mathcal{R}$.*

$\sqcap$-rule:	if 1.	$C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	$\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
	then	$\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$
$\sqcup$-rule:	if 1.	$C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	$\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$
	then	$\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{C_1, C_2\}$
$\exists$-rule:	if 1.	$\exists S.C \in \mathcal{L}(x)$, x is not blocked, and
	2.	x has no S-neighbour y with $C \in \mathcal{L}(y)$
	then	create a new node y with
		$\mathcal{L}(\langle x, y\rangle) := \{S\}$ and $\mathcal{L}(y) := \{C\}$
$\forall_1$-rule:	if 1.	$\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and
		$\forall \mathcal{A}^S.C \notin \mathcal{L}(x)$
	then	$\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{\forall \mathcal{A}^S.C\}$
$\forall_2$-rule:	if 1.	$\forall \mathcal{A}_p.C \in \mathcal{L}(x)$, x is not indirectly blocked, $p \xrightarrow{S} q$ in $\mathcal{A}_p$,
	2.	and there is an S-neighbour y of x with $\forall \mathcal{A}_q.C \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{\forall \mathcal{A}_q.C\}$
$\forall_3$-rule:	if 1.	$\forall \mathcal{A}.C \in \mathcal{L}(x)$, x is not indirectly blocked, $\varepsilon \in L(\mathcal{A})$,
	2.	and $C \notin \mathcal{L}(x)$
	then	$\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C\}$
choose-rule:	if 1.	$(\leqslant n S.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	there is an S-neighbour y of x with $\{C, \dot{\neg} C\} \cap \mathcal{L}(y) = \emptyset$
	then	$\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \dot{\neg} C\}$
$\geqslant$-rule:	if 1.	$(\geqslant n S.C) \in \mathcal{L}(x)$, x is not blocked, and
	2.	there are no $y_1, \ldots, y_n \in S^{\mathbf{T}}(x, C)$
		with $y_i \neq y_j$ for each $1 \leq i < j \leq n$
	then	create n new nodes $y_1, \ldots, y_n$ with $\mathcal{L}(\langle x, y_i\rangle) = \{S\}$,
		$\mathcal{L}(y_i) = \{C\}$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$.
$\leqslant$-rule:	if 1.	$(\leqslant n S.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	$\# S^{\mathbf{T}}(x, C) > n$, there are $y, z \in S^{\mathbf{T}}(x, C)$ with
		not $y \neq z$ and y is not an ancestor of z,
	then	1. $\mathcal{L}(z) \longrightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and
		2. if z is an ancestor of x
		then $\mathcal{L}(\langle z, x\rangle) \longrightarrow \mathcal{L}(\langle z, x\rangle) \cup \mathsf{Inv}(\mathcal{L}(\langle x, y\rangle))$
		else $\mathcal{L}(\langle x, z\rangle) \longrightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$
		3. remove y and the sub-tree below y

Figure 3: The Expansion Rules for the $\mathcal{RIQ}$ Tableau Algorithm.

Proof: Let $m = \sharp\mathsf{fclos}(D, \mathcal{R})$, n the number of roles occurring in D and $\mathcal{R}$, and $n_{\max} := \max\{n \mid ({\geqslant}nR.C) \in \mathsf{clos}(D)\}$. Termination is a consequence of the following properties of the expansion rules:

1. Nodes are labelled with subsets of $\mathsf{fclos}(D, \mathcal{R})$ and edges with sets of roles occurring in D and $\mathcal{R}$, so there are at most 2^{2mn} different possible labellings for a pair of nodes and an edge. Therefore, if a path p is of length at least 2^{2mn}, the pair-wise blocking condition implies the existence of a node x on p such that x is blocked. Since a path on which nodes are blocked cannot become longer, paths are of length at most 2^{2mn}.

2. The expansion rules never remove labels from nodes in the tree, and the only rule that removes a node from the tree is the $\leqslant$-rule.

3. Only the $\exists$- or the $\geqslant$-rule generate new nodes, and each generation is triggered by a concept of the form $\exists R.C$ or $({\geqslant}nR.C)$ in the label of a node x. Each of these concepts triggers at most once the generation of at most $n_{\max}$ R-successors y_i of x: note that if the $\leqslant$-rule subsequently causes an R-successor y_i of x to be removed, then x will have some R-neighbour z with $\mathcal{L}(z) \supseteq \mathcal{L}(y_i)$. This, together with the definition of a clash, implies that the rule application which led to the generation of y_i will not be repeated. Since $\mathsf{fclos}(D, \mathcal{R})$ contains a total of at most m $\exists R.C$, the out-degree of the tree is bounded by $mn_{\max}$. ■

Lemma 6 *Let D be a $\mathcal{RIQ}$-concept and $\mathcal{R}$ an acyclic generalised role hierarchy. If the expansion rules can be applied to D and $\mathcal{R}$ such that they yield a complete and clash-free completion tree, then D has a tableau w.r.t. $\mathcal{R}$.*

Proof: Let $\mathbf{T}$ be a complete and clash-free completion tree. We can "unravel" $\mathbf{T}$ to a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ as follows: Intuitively, an individual in $\mathbf{S}$ corresponds to a *path* in $\mathbf{T}$ from the root node to some node that is not blocked, where we "jump" up to a blocking node instead of going to a blocked one. Doing this naively, one "looses" R-successors blocked by the same node. To ensure that we still have enough successors to satisfy at-least restrictions, we adorn nodes with the nodes they are blocking.

More precisely, a *path* is a sequence of pairs of nodes of $\mathbf{T}$ of the form $p = [\frac{x_0}{x'_0}, \dots, \frac{x_n}{x'_n}]$. For such a path we define $\mathsf{Tail}(p) := x_n$ and $\mathsf{Tail}'(p) := x'_n$. With $[p|\frac{x_{n+1}}{x'_{n+1}}]$, we denote the path $[\frac{x_0}{x'_0}, \dots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}]$. The set $\mathsf{Paths}(\mathbf{T})$ is defined inductively as follows:

- For the root node x_0 of $\mathbf{T}$, $[\frac{x_0}{x_0}] \in \mathsf{Paths}(\mathbf{T})$, and
- For each path $p \in \mathsf{Paths}(\mathbf{T})$ and each successor z of $\mathsf{Tail}(p)$ in $\mathbf{T}$:
  - if z is not blocked, then $[p|\frac{z}{z}] \in \mathsf{Paths}(\mathbf{T})$, and
  - if z is blocked by a node y, then $[p|\frac{y}{z}] \in \mathsf{Paths}(\mathbf{T})$.

Please note that, by construction of $\mathsf{Paths}(\mathbf{T})$ and the blocking condition,

1. if $p \in \mathsf{Paths}(\mathbf{T})$, then $\mathsf{Tail}(p)$ is not blocked,
2. $\mathsf{Tail}(p) = \mathsf{Tail}'(p)$ iff $\mathsf{Tail}'(p)$ is not blocked, and
3. $\mathcal{L}(\mathsf{Tail}(p)) = \mathcal{L}(\mathsf{Tail}'(p))$.

We are now ready to define the tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ as follows:

$$\begin{aligned}
\mathbf{S} &:= \mathsf{Paths}(\mathbf{T}) \\
\mathcal{L}(p) &:= \mathcal{L}(\mathsf{Tail}(p)) \\
\mathcal{E}(R) &:= \{\langle p, [p|\tfrac{x}{x'}]\rangle \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is an } R\text{-successor of } \mathsf{Tail}(p)\} \cup \\
&\quad\ \{\langle [q|\tfrac{x}{x'}], q\rangle \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is an } \mathsf{Inv}(R)\text{-successor of } \mathsf{Tail}(q)\} \cup
\end{aligned}$$

We show that T is a tableau for D.

- T satisfies (P0) because D is in the label of the root node which cannot be blocked.
- T satisfies (P1) because $\mathbf{T}$ is clash-free.
- (P2), (P3), and (P10) are satisfied by T because $\mathbf{T}$ is complete and thus neither the $\sqcap$-, the $\sqcup$-, nor the *choose*-rule is applicable to $\mathbf{T}$.
- For (P4a), let $s, t \in \mathbf{S}$ with $\forall \mathcal{A}_p.C \in \mathcal{L}(s)$, $\langle s, t\rangle \in \mathcal{E}(S)$, and $p \xrightarrow{S} q$ in $\mathcal{A}_p$.

  If $t = [s|\frac{x}{x'}]$, then x' is an S-successor of $\mathsf{Tail}(s)$ and completeness of $\mathbf{T}$ implies that the $\forall_2$-rule cannot be applied to s, hence $\forall \mathcal{A}_q.C \in \mathcal{L}(x') = \mathcal{L}(x) = \mathcal{L}(t)$.

  If $s = [t|\frac{x}{x'}]$, then x' is an $\mathsf{Inv}(S)$-successor of $\mathsf{Tail}(t)$ and $\forall \mathcal{A}_p.C \in \mathcal{L}(x')$. Again, completeness of $\mathbf{T}$ implies $\forall \mathcal{A}_q.C \in \mathcal{L}(\mathsf{Tail}(t)) = \mathcal{L}(t)$.
- For (P4b), let $s \in \mathbf{S}$ with $\forall \mathcal{A}.C \in \mathcal{L}(s) = \mathcal{L}(\mathsf{Tail}(s))$ and $\varepsilon \in L(\mathcal{A})$. Since the $\forall_3$-rule cannot be applied, $C \in \mathcal{L}(\mathsf{Tail}(s))$, and hence $C \in \mathcal{L}(s)$.
- For (P5), let $\exists R.C \in \mathcal{L}(s)$ and $\mathsf{Tail}(s) = x$. Since x is not blocked and $\mathbf{T}$ complete, x has some R-neighbour y with $C \in \mathcal{L}(y)$.
  - If y is a successor of x and y is
    * not blocked, then $t := [s|\frac{y}{y}] \in \mathbf{S}$.
    * blocked, then y is directly blocked, say by z. Hence $t := [s|\frac{z}{y}] \in \mathbf{S}$, and the blocking condition implies $C \in \mathcal{L}(z)$.
  - x is an $\mathsf{Inv}(R)$-*successor* of y, then either
    * $s = [t|\frac{x}{x'}]$ with $\mathsf{Tail}(t) = y$.
    * $s = [t|\frac{x}{x'}]$ with $\mathsf{Tail}(t) = y' \neq y$. Since x only has one predecessor, y' is not the predecessor of x. This implies $x \neq x'$, x blocks x', and y' is the predecessor of x' due to the construction of Paths. The definition of the blocking condition implies $\mathcal{L}(x') = \mathcal{L}(x)$ $\mathcal{L}(\langle y', x'\rangle) = \mathcal{L}(\langle y, x\rangle)$, and $\mathcal{L}(y') = \mathcal{L}(y)$.

In all four cases, $\langle s,t\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(t)$.

- For (P6), let $\forall S.C \in \mathcal{L}(s) = \mathcal{L}(\mathsf{Tail}(s))$. Then completeness implies that the $\forall_1$-rule cannot be applied, and thus $\forall \mathcal{A}^S.C \in \mathcal{L}(\mathsf{Tail}(s))$.
- (P7) holds because of the symmetric definition of the mapping $\mathcal{E}$.
- For (P8) and (P9), let $s \in \mathbf{S}$ with $\mathsf{Tail}(s) = x$ and $(\bowtie nS.C) \in \mathcal{L}(s) = \mathcal{L}(x)$ for $\bowtie \in \{\leq, \geq\}$. Then $\mathbf{T}$ being complete and clash-free and $(\bowtie nS.C) \in \mathcal{L}(x)$ implies $\#S^{\mathbf{T}}(x,C) \bowtie n$. We prove that T satisfies (P8) and (P9) by showing that
$$\#S^T(s,C) = \#S^{\mathbf{T}}(x,C).$$
The $\leq$ part is implied by the definition of $\mathcal{E}(\cdot)$ which yields at most one t with $\langle s,t\rangle \in \mathcal{E}(S)$ and $C \in \mathcal{L}(t)$ per S-neighbour of x with C in its label: $\mathcal{E}(\cdot)$ yields exactly one such t per S-successor of x with C in its label, and there is at most one t in $\mathbf{S}$ with $s = [t|\frac{x}{x'}]$. The $\geq$ part is due to the fact that (a) if x is an $\mathsf{Inv}(S)$-successor of its predecessor y with $C \in \mathcal{L}(y)$, then $\langle s,t\rangle \in \mathcal{E}(S)$ for the "prefix" t of s with $s = [t|\frac{x}{x'}]$, and (b) if x has two S-successors $y_1 \neq y_2$ with $C \in \mathcal{L}(y_i)$, then $\langle s,t_1\rangle, \langle s,t_2\rangle \in \mathcal{E}(S)$ for $t_1 = [s|\frac{y_1'}{y_1}] \neq [s|\frac{y_2'}{y_2}] = t_2$. ■

Lemma 7 *Let D be a $\mathcal{RIQ}$-concept and $\mathcal{R}$ an acyclic generalised role hierarchy. If D has a tableau w.r.t. $\mathcal{R}$, then the expansion rules can be applied to D and $\mathcal{R}$ such that they yield a complete and clash-free completion tree.*

Proof: Let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ be a tableau for D and $\mathcal{R}$. We use T to trigger the application of the expansion rules such that they yield a completion tree $\mathbf{T}$ that is both complete and clash-free. To this purpose, we use a function π which maps the nodes of $\mathbf{T}$ to elements of $\mathbf{S}$. The mapping π is defined as follows:

- For the root node x_0 of $\mathbf{T}$, we define $\pi(x_0) = s_0$ for some $s_0 \in \mathbf{S}$ with $D \in \mathcal{L}(s_0)$ (such an s_0 exists because of (P0)).
- If $\pi(x)$ is already defined, and a successor y of x is generated for $\exists R.C \in \mathcal{L}(x)$, then $\pi(y) = t$ for some $t \in \mathbf{S}$ with $C \in \mathcal{L}(t)$ and $\langle \pi(x), t\rangle \in \mathcal{E}(R)$.
- If $\pi(x)$ is already defined, and successors y_i of x are generated for $(\geqslant nR.C) \in \mathcal{L}(x)$, then $\pi(y_i) = t_i$ for n distinct $t_i \in \mathbf{S}$ with $C \in \mathcal{L}(t_i)$ and $\langle \pi(x), t_i\rangle \in \mathcal{E}(R)$.

Due to the following observations, π is well-defined:

Firstly, the mapping for the initial completion tree for D and $\mathcal{R}$ obviously satisfies the following three conditions:

$$\left.\begin{array}{l} \mathcal{L}(x) \subseteq \mathcal{L}(\pi(x)), \\ \text{if } y \text{ is an } S\text{-neighbour of } x, \text{ then } \langle \pi(x), \pi(y)\rangle \in \mathcal{E}(S), \text{ and} \\ x \neq y \text{ implies } \pi(x) \neq \pi(y). \end{array}\right\} \qquad (*)$$

Secondly, we show that the following claim holds:
CLAIM: Let $\mathbf{T}$ be generated by the completion algorithm for D and $\mathcal{R}$ and let π satisfy $(*)$. If an expansion rule is applicable to $\mathbf{T}$, then this rule can be applied such that it yields a completion tree $\mathbf{T}'$ and a (possibly extended) π that satisfy $(*)$.

As a consequence of this claim, (P1), and (P8) we have that if D and $\mathcal{R}$ have a tableau, then the expansion rules can be applied to D and $\mathcal{R}$ such that they yield a clash-free completion tree, which is eventually also complete due to Lemma 5.

The claim can be proved by a case distinction on the completion rules: Let $\mathbf{T}$ be generated by the completion algorithm for D and $\mathcal{R}$ and let π satisfy $(*)$.

- If the $\sqcap$-, the $\sqcup$-, the $\forall_1$-, or the $\forall_3$-rule are applicable to $\mathbf{T}$, then the first line of $(*)$ together with (P2), (P3), (P6), or (P4b) imply that each of these rules can be applied in such a way[2] that $(*)$ also holds after its application.
- If the $\forall_2$- or the *choose*-rule is applicable to $\mathbf{T}$, then the first two lines of $(*)$ together with (P4a) or (P10) imply that each of these rules can be applied in such a way[2] that $(*)$ also holds after its application.
- If the $\exists$-rule is applicable to x with $\exists S.C \in \mathcal{L}(x)$, then $(*)$ implies that $\exists S.C \in \mathcal{L}(\pi(x))$, and thus (P5) implies the existence of $b \in \mathbf{S}$ with $C \in \mathcal{L}(b)$ and $\langle \pi(x), b\rangle \in \mathcal{E}(S)$. Hence applying the $\exists$-rule and extending π with $\pi(y) := b$ for y the new node y generated preserves $(*)$.
- If the $\geqslant$-rule is applicable to x with $(\geqslant nS.C) \in \mathcal{L}(x)$, then $(*)$ implies that $(\geqslant nS.C) \in \mathcal{L}(\pi(x))$, and thus (P9) implies the existence of $b_1, \dots, b_n \in \mathbf{S}$ with $C \in \mathcal{L}(b_i)$, $\langle \pi(x), b_i\rangle \in \mathcal{E}(S)$, and $b_i \neq b_j$ for all $i \neq j$. Hence π can be extended with $\pi(y_i) := b_i$ for y_i the newly generated nodes, thus preserving $(*)$.
- If the $\leqslant$-rule is applicable to x with $(\leqslant nS.C) \in \mathcal{L}(x)$, then $(*)$ implies that $(\leqslant nS.C) \in \mathcal{L}(\pi(x))$, and thus (P8) implies that there are at most n $b_i \in \mathbf{S}$ with $C \in \mathcal{L}(b_i)$ and $\langle \pi(x), b_i\rangle \in \mathcal{E}(S)$. Thus the second line of $(*)$ implies that $\pi(y_i) = \pi(y_j)$ for y_i, y_j two S-neighbours of x with $C \in \mathcal{L}(y_i) \cap \mathcal{L}(y_j)$, and the (contra-position of the) last line of $(*)$ ensures that *not* $y_i \neq y_j$. Hence the $\leqslant$-rule can be applied while preserving $(*)$. ■

From Theorem 1, Lemma 4, 5, 6, and 7, we thus have the following theorem:

Theorem 3 *The completion algorithm decides satisfiability and subsumption of $\mathcal{RIQ}$-concepts with respect to acyclic generalised role hierarchies and terminologies.*

[2] This non-deterministic formulation is due to the non-determinism of the $\sqcup$- resp. the *choose*-rule.

5 Avoiding the blow-up

In the previous section, we have presented an algorithm that decides satisfiability and subsumption of $\mathcal{RIQ}$-concepts with respect to acyclic generalised role hierarchies and terminologies. Unfortunately, compared to similar algorithms that are implemented in state-of-the-art description logic reasoners [Horrocks, 1998; Patel-Schneider and Horrocks, 1999; Haarslev and Möller, 2001] and behave well in many cases, we have here an exponential blow-up: the closure $\mathsf{fclos}(D, \mathcal{R})$ is exponential in $\mathcal{R}$ since we have "unfolded" the acyclic generalised role hierarchy $\mathcal{R}$ into the possibly exponentially large $\exp(\mathcal{R})$. While investigating whether and how this exponential blow-up can be avoided, we observe that a further restriction of the syntax of acyclic generalised role hierarchies avoids this blow-up:

An acyclic generalised role hierarchy $\mathcal{R}$ is called *simple* if, whenever $R_1 \circ S \sqsubseteq S$ and $S \circ R_2 \sqsubseteq S$ are in $\mathcal{R}$, then R_1 and R_2 do not have a common subrole R' that occurs on the right hand side of an axiom $R' \circ S' \sqsubseteq R'$ or $S' \circ R' \sqsubseteq R'$.

For a *simple* acyclic generalised role hierarchy $\mathcal{R}$, $\exp(\mathcal{R})$ is only polynomial in the size of $\mathcal{R}$ since each term used in the substitution step of the construction of $\exp(\mathcal{R})$ from $\mathcal{R}$ is at most used once in each other axiom.

Lemma 8 *For a $\mathcal{RIQ}$-concept D and a simple acyclic generalised role hierarchy $\mathcal{R}$, the size of $\mathsf{fclos}(D, \mathcal{R})$ is polynomial in the size of D and $\mathcal{R}$.*

Thus, for simple role hierarchies, the tableau algorithm presented here is of the same worst case complexity as for $\mathcal{SHIQ}$, namely 2NExpTime. A detailed investigation of the exact complexity will be part of future work.

6 Evaluation of the $\mathcal{RIQ}$ algorithm in FaCT

In order to evaluate the practicability of the above algorithm, we have extended the DL system FaCT [Horrocks, 1998] to deal with $\mathcal{RIQ}$, and we have carried out a preliminary empirical evaluation.

From a practical point of view, one potential problem with the $\mathcal{RIQ}$ algorithm is that the number of different automata, and hence the number of different $\forall\mathcal{A}.C$ concepts, could be very large. Moreover, many of these automata could be equivalent (i.e., accept the same languages). As blocking depends on finding ancestor nodes labeled with the same set of concepts, the discovery of blocks could be unnecessarily delayed, and this can lead to a serious degradation in performance [Horrocks and Sattler, 2002].

The FaCT implementation addresses these possible problems by transforming all of the initial NFAs into minimal deterministic NFAs (using the AT&T FSM Library™ [Mohri *et al.*, 1998]). Only one finite state automata is constructed for each role, the states in each automaton are uniquely numbered, and the implementation uses concepts of the form $\forall\mathcal{A}.C$, where $\mathcal{A}$ is the number of a state in one of the automata. Because the automata are deterministic, for each concept of the form $\forall\mathcal{A}.C$ in the label of a node, the $\forall_2$-rule can add at most one concept to the label of a given neighbouring node. Moreover, because the automata are minimal, if $\forall\mathcal{A}.C$ leads to the presence of $\forall\mathcal{A}'.C$ in some successor node (as a result of repeated applications of the $\forall_2$-rule), then $\forall\mathcal{A}.C$ is equivalent to $\forall\mathcal{A}'.C$ iff $\mathcal{A} = \mathcal{A}'$. As $\mathcal{A}$ and $\mathcal{A}'$ are numbers, such comparisons are very easy, and minimisation of automata avoids unnecessary blocking delays.

The implementation is still at the "beta" stage, but it has been possible to carry out some preliminary tests using the well known Galen medical terminology KB [Rector and Horrocks, 1997; Horrocks, 1998]. This KB contains 2,740 named concepts and 413 roles, 26 of which are transitive. The roles are arranged in a relatively complex hierarchy with a maximum depth of 10. Classifying this KB using FaCT's $\mathcal{SHIQ}$ reasoner takes 116s on an 800 MHz Pentium III equipped Linux PC. Classifying the same KB using the new $\mathcal{RIQ}$ reasoner took a total of 275s, but this includes 135s to compute the minimal deterministic NFAs for the role box (it should be noted that this is an unusually large and complex role box, and that computing the NFAs is a preprocessing step that will not need to be repeated when the remainder of the KB is extended, modified, or queried). This result is encouraging as it shows that, in the case of the Galen KB at least, using automata in $\forall\mathcal{A}.C$ concepts does not lead to a significant degradation in performance. Moreover, the time taken by the $\mathcal{RIQ}$ reasoner includes approximately 100s to compute the minimal deterministic automata for the role box. This overhead could become significant if optimisations of the $\mathcal{RIQ}$ reasoner result in even better performance, but it should be noted that (a) this is a preprocessing step that will not need to be repeated when the remainder of the KB is extended, modified or queried, and (b) this is an unusually large and complex role box.

The KB was then extended with several role inclusion axioms that express the propagation of location across various partonomic roles. These included

$$\mathtt{hasLocation} \circ \mathtt{isSolidDivisionOf} \sqsubseteq \mathtt{hasLocation}$$

and

$$\mathtt{hasLocation} \circ \mathtt{isLayerOf} \sqsubseteq \mathtt{hasLocation}.$$

Classifying the extended KB took 280s, an increase of only 2% (3.5% if we exclude the NFA computation time). Subsumption queries w.r.t. this KB revealed that, e.g.,

$$\mathtt{Fracture} \sqcap \exists\mathtt{hasLocation}.\mathtt{NeckOfFemur}$$

was implicitly a kind of

$$\mathtt{Fracture} \sqcap \exists\mathtt{hasLocation}.\mathtt{Femur}$$

(`NeckOfFemur` is a solid division of `Femur`), and

$$\mathtt{Ulcer} \sqcap \exists\mathtt{hasLocation}.\mathtt{GastricMucosa}$$

was implicitly a kind of

$$\mathtt{Ulcer} \sqcap \exists\mathtt{hasLocation}.\mathtt{Stomach}$$

(`GastricMucosa` is a layer of `Stomach`). None of these subsumption relationships held w.r.t. the original KB. The times taken to compute these relationships w.r.t. the classified KB could not be measured accurately as they were of the same order as a system clock tick (10ms).

7 Discussion

Motivated (primarily) by medical terminology applications, we have investigated the decidability of the well known expressive DL, $\mathcal{SHIQ}$, extended with RIAs of the form $R \circ S \sqsubseteq P$. We have shown that this extension is undecidable even when RIAs are restricted to the forms $R \circ S \sqsubseteq R$ or $S \circ R \sqsubseteq R$, but that decidability can be regained by further restricting RIAs to be acyclic. We have presented a tableau algorithm for this DL and reported on its implementation in the FaCT system. A preliminary evaluation suggests that the algorithm will perform well in realistic applications and demonstrates that it can provide important additional functionality in a medical terminology application.

References

[Baader *et al.*, 1993] F. Baader, H.-J. Bürckert, B. Nebel, W. Nutt, and G. Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. *Journal of Logic, Language and Information*, 2:1–18, 1993.

[Baader, 1991] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, 1991.

[Baldoni *et al.*, 1998] M. Baldoni, L. Giordano, and A. Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In *Proc. of the Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX-98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1998.

[Baldoni, 1998] M. Baldoni. *Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension*. PhD thesis, Dipartimento di Informatica, Università degli Studi di Torino, Italy, 1998.

[Berger, 1966] R. Berger. The undecidability of the dominoe problem. *Mem. Amer. Math. Soc.*, 66, 1966.

[Brachman and Schmolze, 1985] R. J. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Demri, 2001] S. Demri. The complexity of regularity in grammar logics and related modal logics. *Journal of Logic and Computation*, 11(6), 2001.

[Fariñas del Cerro and Penttonen, 1988] L. Fariñas del Cerro and M. Penttonen. Grammar logics. *Logique et Analyse*, 121-122:123–134, 1988.

[Haarslev and Möller, 2001] V. Haarslev and R. Möller. RACER system description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[Hopcroft and Ullman, 1997] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison Wesley Publ. Co., Reading, Massachussetts, 1997.

[Horrocks and Sattler, 2002] I. Horrocks and U. Sattler. Optimised reasoning for shiq. In *Proc. of the 15th European Conf. on Artificial Intelligence (ECAI 2002)*, 2002.

[Horrocks *et al.*, 1999] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer-Verlag, 1999.

[Horrocks *et al.*, 2000] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic shiq. In D. MacAllester, editor, *Proc. of the 17th Conf. on Automated Deduction (CADE-17)*, number 1831 in Lecture Notes in Computer Science, Germany, 2000. Springer-Verlag.

[Horrocks, 1998] I. Horrocks. Using an Expressive Description Logic: FaCT or Fiction? In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-98)*, 1998.

[Kupferman *et al.*, 2002] O. Kupferman, U. Sattler, and M. Y. Vardi. The complexity of the graded mu-calculus. In *Proc. of the 18th Conf. on Automated Deduction (CADE-19)*, volume 2392 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2002.

[Lenat and Guha, 1989] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems*. Addison Wesley Publ. Co., Reading, Massachussetts, 1989.

[Mohri *et al.*, 1998] M. Mohri, F. C. N. Pereira, and M. Riley. *A Rational Design for a Weighted Finite-State Transducer Library*. Number 1436 in Lecture Notes in Computer Science. Springer-Verlag, 1998.

[Padgham and Lambrix, 1994] L. Padgham and P. Lambrix. A framework for part-of hierarchies in terminological logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 485–496, 1994.

[Patel-Schneider and Horrocks, 1999] P. F. Patel-Schneider and I. Horrocks. DLP and FaCT. In *Proc. of the Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX-99)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 19–23. Springer-Verlag, 1999.

[Rector and Horrocks, 1997] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. of the WS on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*. AAAI Press, 1997.

[Rector *et al.*, 1997] A. Rector, S. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon. The GRAIL concept modelling language for medical terminology. *AI in Medicine*, 9:139–171, 1997.

[Rector, 2002] A. Rector. Analysis of propagation along transitive roles: Formalisation of the galen experience with medical ontologies. In *Proc. of the 2001 Description Logic Workshop (DL 2002)*, 2002.

[Sattler, 2000] 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.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 466–471, Sydney, 1991.

[Schmidt-Schauß and Smolka, 1991] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[Schmidt-Schauss, 1989] M. Schmidt-Schauss. Subsumption in KL-ONE is undecidable. In *Proc. of the 1st Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-89)*, pages 421–431, Boston (USA), 1989.

[Schulz and Hahn, 2001] S. Schulz and U. Hahn. Parts, locations, and holes - formal reasoning about anatomical structures. In *Proc. of AIME 2001*, volume 2101 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[Spackman, 2000] K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Informatics Association*, 2000. Fall Symposium Special Issue.

[Tobies, 2001] S. Tobies. PSPACE reasoning for graded modal logics. *Journal of Logic and Computation*, 11(1):85–106, 2001.

[Wessel, 2001] M. Wessel. Obstacles on the way to qualitative spatial reasoning with description logics: Some undecidability results. In *Proc. of the 2001 Description Logic Workshop (DL 2001)*, 2001.