The Irresistible $\mathcal{SRIQ}$

Ian Horrocks, Oliver Kutz, and Ulrike Sattler

School of Computer Science, The University of Manchester,
Kilburn Building, Oxford Road, Manchester, M13 9PL, UK,
{Horrocks, Kutz, Sattler}@cs.man.ac.uk

Abstract. Motivated primarily by medical terminology applications, the prominent DL $\mathcal{SHIQ}$ has already been extended to a DL with complex role inclusion axioms of the form $R \circ S \dot{\sqsubseteq} R$ or $S \circ R \dot{\sqsubseteq} R$, called $\mathcal{RIQ}$, and the $\mathcal{SHIQ}$ tableau algorithm has been extended to handle such inclusions.

This paper further extends $\mathcal{RIQ}$ and its tableau algorithm with important expressive means that are frequently requested in ontology applications, namely with reflexive, symmetric, transitive, and irreflexive roles, disjoint roles, and the construct $\exists R.\mathsf{Self}$, allowing, for instance, the definition of concepts such as a "narcist". Furthermore, we extend the algorithm to cover Abox reasoning extended with negated role assertions. The resulting logic is called $\mathcal{SRIQ}$.

1 Introduction

We describe an extension, called $\mathcal{SRIQ}$, of the description logic (DL) $\mathcal{SHIN}$ (11) underlying OWL lite and OWL DL (7). We believe that $\mathcal{SRIQ}$ enjoys some useful properties. Firstly, $\mathcal{SRIQ}$ extends $\mathcal{SHIN}$ with numerous expressive means which have been asked for by users, and which, we believe, will make modeling using DLs easier and more intuitive. Even though some of these expressive means can be viewed as minor syntactic sugar, their absence has proven quite harmful since developers of ontologies use work-arounds to compensate for this. As a consequence, ontologies become cluttered, complicated, and difficult to understand. In the worst case, the work-around only partially captures the intended semantics, thus leading to unintended or missing consequences, thereby destroying one of the main features of a logic-based formalism, namely its well-defined semantics and reasoning services. A well-known example of such an expressive means are *qualified* number restrictions. Their absence in OWL lite and OWL DL has caused problems in the past (14), and has lead to the development and use of questionable surrogates. Hence, $\mathcal{SRIQ}$ provides qualified number restrictions. Other, novel expressive means of $\mathcal{SRIQ}$ concern mostly roles and include:

– *disjoint roles.* E.g., the roles `sister` and `mother` could be declared as being disjoint. Most DLs can be said to be "lopsided" since they allow to express disjointness on concepts but not on roles, despite the fact that role disjointness is quite natural and can generate new subsumptions or inconsistencies in the presence of role hierarchies and number restrictions.

- *reflexive and irreflexive roles*. E.g., the role `knows` could be declared as being reflexive, and the role `sibling` could be declared as being irreflexive. Such statements can only cause new consequences in the presence of Aboxes, or in interplay with the new concept $\exists R.\mathsf{Self}$ described below, and could thus be said to be merely syntactic sugar.
- *negated role assertions*. Most Abox formalisms only allow for positive role assertions (with few exceptions (1; 5)), whereas $\mathcal{SRIQ}$ also allows for statements such as $(\texttt{John}, \texttt{Mary}) : \neg\texttt{likes}$. In the presence of complex role inclusions, negated role assertions can be quite useful and, like disjoint roles, they overcome a certain "lopsidedness" of DLs.
- Since $\mathcal{SRIQ}$ extends $\mathcal{SHIQ}$, we can also express that a role is transitive or symmetric, and can use role inclusion axioms $R \sqsubseteq S$.
- Since $\mathcal{SRIQ}$ extends $\mathcal{RIQ}$ (9), we can use complex role inclusion axioms of the form $R \circ S \dot{\sqsubseteq} R$ and $S \circ R \dot{\sqsubseteq} R$. For example, w.r.t. the axiom $\texttt{owns} \circ \texttt{hasPart} \dot{\sqsubseteq} \texttt{owns}$, and the fact that each car contains an engine $\texttt{Car} \dot{\sqsubseteq} \exists\texttt{hasPart}.\texttt{Engine}$, an owner of a car is also an owner of an engine, i.e., the following subsumption is implied: $\exists\texttt{owns}.\texttt{Car} \sqsubseteq \exists\texttt{owns}.\texttt{Engine}$.
- Finally, $\mathcal{SRIQ}$ allows for concepts of the form $\exists R.\mathsf{Self}$ which can be used to express "local reflexivity" of a role R, e.g., to define the concept "narcist" using $\exists\texttt{likes}.\mathsf{Self}$.

Besides a Tbox and an Abox, $\mathcal{SRIQ}$ provides a so-called *Rbox* to gather all statements concerning roles.

Secondly, $\mathcal{SRIQ}$ is designed to be of similar practicability as $\mathcal{SHIQ}$. The tableau algorithm for $\mathcal{SHIQ}$ and the one for $\mathcal{SRIQ}$ presented here are very similar, even though the additional expressive means of $\mathcal{SRIQ}$ require certain adjustments. However, these adjustments do not add new sources of non-determinism, and can thus be seen as "harmless". More precisely, we employ the same technique using finite automata as in (9) to handle role inclusions $R \circ S \dot{\sqsubseteq} R$ and $S \circ R \dot{\sqsubseteq} R$. This involves a pre-processing step which takes an Rbox and builds, for each role R, a finite automaton that accepts exactly those words $R_1 \dots R_n$ such that, in each model of the Rbox, $\langle x, y\rangle \in (R_1 \dots R_n)^{\mathcal{I}}$ implies $\langle x, y\rangle \in R^{\mathcal{I}}$. These automata are then used in the tableau expansion rules to check, for a node x with $\forall R.C \in \mathfrak{L}(x)$ and an $R_1 \dots R_n$-neighbour y of x, whether to add C to $\mathfrak{L}(y)$. Even though the pre-processing step might appear a little cumbersome, the usage of the automata in the algorithm makes it quite elegant and compact.

The current paper describes work in progress towards a description logic that overcomes certain shortcomings in expressiveness of other DLs. We have used $\mathcal{SHIN}$, $\mathcal{SHIQ}$, and $\mathcal{RIQ}$ as a starting point, extended them with some "useful-yet-harmless" expressive means, and also extended the tableau algorithm accordingly. Our selection of "harmless" language extensions certainly is not complete, and it has to be seen which further additions are particularly desirable. Moreover, we wish to generalise $\mathcal{SRIQ}$ also in another direction: currently, various new operators are restricted to *simple* roles, and we have yet to establish which of these restrictions are necessary in order to preserve decidability[1] or practica-

[1] See (11) for such a case.

bility. Moreover, we plan to extend $\mathcal{SRIQ}$ towards $\mathcal{SHOIQ}$ (10), i.e., to also include nominals.

2 The Logic $\mathcal{SRIQ}$

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

2.1 Roles, Role Hierarchies, and Role Assertions

Definition 1 (Interpretations). *Let* $\mathbf{C}$ *be a set of* ***concept names****,* $\mathbf{R}$ *a set of* ***role names****, and* $\mathbf{I} = \{a, b, c \ldots\}$ *a set of* ***individual names****. The set of* ***roles*** *is* $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$*, where a role* R^- *is called the* ***inverse role*** *of* R*.*

As usual, 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 associates, with each role name* R*, a binary relation* $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$*, with each concept name* C *a subset* $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$ *and, with each individual name* a *an element* $a^\mathcal{I} \in \Delta^\mathcal{I}$*. Inverse roles are interpreted as usual, i.e., for each role* $R \in \mathbf{R}$*, we have*

$$(R^-)^\mathcal{I} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R^\mathcal{I}\}.$$

Note that, unlike in the case of $\mathcal{SHIQ}$, we did not introduce *transitive role names*. This is so since, as will become apparent below, role box assertions can be used to force roles to be transitive.

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

Since we will often work with a string of roles, it is convenient to extend both $\cdot^\mathcal{I}$ and $\mathsf{Inv}(\cdot)$ to such strings: if $w = R_1 \ldots R_n$ for R_i roles, then we set $w^\mathcal{I} = R_1^\mathcal{I} \circ \ldots \circ R_n^\mathcal{I}$ and $\mathsf{Inv}(w) = \mathsf{Inv}(R_n) \ldots \mathsf{Inv}(R_1)$, where $\circ$ denotes composition of binary relations.

A *role box* $\mathcal{R}$ consists of two components. The first component is a *role hierarchy* $\mathcal{R}_h$ which consists of (generalised) *role inclusion axioms*, i.e., statements of the form $R \dot\sqsubseteq S$, $RS \dot\sqsubseteq S$, and $SR \dot\sqsubseteq S$. The second component is a set $\mathcal{R}_a$ of *role assertions* stating, for instance, that a role R must be interpreted as a transitive, reflexive, irreflexive, symmetric, or transitive relation, or that two (possibly inverse) roles R and S are to be interpreted as *disjoint* binary relations.

We start with the definition of a role hierarchy, whose definition involves a strict partial order $\prec$ on roles, i.e., an irreflexive and transitive relation on $R \cup \{R^- \mid R \in \mathbf{R}\}$.

Definition 2 ((Regular) Role Inclusion Axioms).

Let $\prec$ *be a strict partial order on roles. A* ***role inclusion axiom*** *(RIA for short) is an expression of the form* $w \dot\sqsubseteq R$*, where* w *is a finite string of roles, and* R *is a role name. A* ***role hierarchy*** $\mathcal{R}_h$*, then, is a finite set of RIAs.*

An interpretation $\mathcal{I}$ ***satisfies*** *a role inclusion axiom* $S_1 \ldots S_n \dot\sqsubseteq R$*, if*

$$S_1^\mathcal{I} \circ \ldots \circ S_n^\mathcal{I} \subseteq R^\mathcal{I},$$

*where $\circ$ stands for the composition of binary relations. An interpretation is a **model** of a role hierarchy R_h, if it satisfies all RIAs in R_h, written $\mathcal{I} \models R_h$.*
*A RIA $w \dot{\sqsubseteq} R$ is $\prec$**-regular** if*

- *R is a role name,*
- *$w = RR$,*
- *$w = R^-$,*
- *$w = S_1 \dots S_n$ and $S_i \prec R$, for all $1 \leq i \leq n$,*
- *$w = RS_1 \dots S_n$ and $S_i \prec R$, for all $1 \leq i \leq n$, or*
- *$w = S_1 \dots S_n R$ and $S_i \prec R$, for all $1 \leq i \leq n$.*

*Finally, a role hierarchy $\mathcal{R}_h$ is said to be **regular** if there exists a strict partial order $\prec$ on roles such that each RIA in $\mathcal{R}_h$ is $\prec$-regular.*

Regularity prevents a role hierarchy from containing cyclic dependencies. For instance, the role hierarchy

$$\{RS \dot{\sqsubseteq} S, \quad RT \dot{\sqsubseteq} R, \quad UT \dot{\sqsubseteq} T, \quad US \dot{\sqsubseteq} U\}$$

is not regular because it would require $\prec$ to satisfy $S \prec U \prec T \prec R \prec S$, which would imply $S \prec S$, thus contradicting irreflexivity. Such cyclic dependencies are known to lead to undecidability (9).

From the definition of the semantics of inverse roles, it follows immediately that

$$\langle x, y\rangle \in w^{\mathcal{I}} \text{ iff } \langle y, x\rangle \in \mathsf{Inv}(w)^{\mathcal{I}}.$$

Hence, each model satisfying $w \dot{\sqsubseteq} S$ also satisfies $\mathsf{Inv}(w) \dot{\sqsubseteq} \mathsf{Inv}(S)$ (and vice versa), and thus the restriction to those RIAs with role *names* on their right hand side does not have any effect on expressivity.

Given a role hierarchy $\mathcal{R}_h$, we define the relation $\underline{\circledast}$ to be the transitive-reflexive closure of $\dot{\sqsubseteq}$ over $\{R \dot{\sqsubseteq} S, \mathsf{Inv}(R) \dot{\sqsubseteq} \mathsf{Inv}(S) \mid R \dot{\sqsubseteq} S \in \mathcal{R}_h\}$. A role R is called a **sub-role** (resp. **super-role**) of a role S if $R \underline{\circledast} S$ (resp. $S \underline{\circledast} R$). Two roles R and S are **equivalent** ($R \equiv S$) if $R \underline{\circledast} S$ and $S \underline{\circledast} R$.

Note that, due to the fourth restriction in the definition of $\prec$-regularity, we also restrict $\underline{\circledast}$ to be acyclic, and thus regular role hierarchies never contain two equivalent roles.[2]

Next, let us turn to the second component of Rboxes, the role assertions. For an interpretation $\mathcal{I}$, we define $Diag^{\mathcal{I}}$ to be the set $\{\langle x, x\rangle \mid x \in \Delta^{\mathcal{I}}\}$ and set $R^{\mathcal{I}} \downarrow := \{\langle x, x\rangle \mid \exists y \in \Delta^{\mathcal{I}}.\langle x, y\rangle \in R^{\mathcal{I}}\}$.

Definition 3 (Role Assertions). *For roles R and S, we call the assertions $\mathsf{Ref}(R)$, $\mathsf{Irr}(R)$, $\mathsf{Sym}(R)$, $\mathsf{Tra}(R)$, and $\mathsf{Dis}(R, S)$, **role assertions**, where, for each interpretation $\mathcal{I}$ and all $x,\ y,\ z \in \Delta^{\mathcal{I}}$, we have:*

[2] This is not a serious restriction for, if $\mathcal{R}$ contains $\underline{\circledast}$ cycles, we can simply choose one role R from each cycle and replace all other roles in this cycle with R in the input Rbox, Tbox and Abox (see below).

$\mathcal{I} \models \mathsf{Sym}(R)$ *if* $\langle x,y\rangle \in R^{\mathcal{I}}$ *implies* $\langle y,x\rangle \in R^{I}$;
$\mathcal{I} \models \mathsf{Tra}(R)$ *if* $\langle x,y\rangle \in R^{\mathcal{I}}$ *and* $\langle y,z\rangle \in R^{\mathcal{I}}$ *imply* $\langle x,z\rangle \in R^{I}$;
$\mathcal{I} \models \mathsf{Ref}(R)$ *if* $R^{\mathcal{I}} \downarrow\subseteq R^{\mathcal{I}}$;
$\mathcal{I} \models \mathsf{Irr}(R)$ *if* $R^{\mathcal{I}} \cap Diag^{\mathcal{I}} = \emptyset$;
$\mathcal{I} \models \mathsf{Dis}(R,S)$ *if* $R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$.

Adding symmetric and transitive role assertions is a trivial move since both of these expressive means can be replaced by complex role inclusion axioms as follows: for the role assertion $\mathsf{Sym}(R)$ we can add to the Rbox, equivalently, the role inclusion axiom $R^{-} \dot{\sqsubseteq} R$, and, for the role assertion $\mathsf{Tra}(R)$, we can add to the Rbox, equivalently, $RR \dot{\sqsubseteq} R$. The proof of this should be obvious. Thus, as far as expressivity is concerned, we can assume for convenience that no role assertions of the form $\mathsf{Tra}(R)$ or $\mathsf{Sym}(R)$ appear in $\mathcal{R}_a$, but that transitive and symmetric roles will be handled by the RIAs alone.

The situation is different, however, for the other Rbox assertions. Neither reflexivity nor irreflexivity nor disjointness of roles can be enforced by role inclusion axioms. However, as we shall see later, reflexivity and irreflexivity of roles are closely related to the new concept $\exists R.\mathsf{Self}$.

In $\mathcal{SHIQ}$, the application of qualified number restrictions has to be restricted to certain roles, called *simple roles*, to preserve decidability (11). In the context of $\mathcal{SRIQ}$, the definition of *simple role* has to be slightly modified, and simple roles figure not only in qualified number restrictions, but in several other constructs as well. Intuitively, non-simple roles are those that are implied by the composition of roles.

Given a role hierarchy $\mathcal{R}_h$ and a set of role assertions $\mathcal{R}_a$ (without transitivity or symmetry assertions), the set of roles that are **simple in** $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ is inductively defined as follows:

- a role name is simple if it does not occur on the right hand side of a RIA in $\mathcal{R}_h$,
- an inverse role R^{-} is simple if R is, and
- if R occurs on the right hand side of a RIA in $\mathcal{R}_h$, then R is simple if, for each $w \dot{\sqsubseteq} R \in \mathcal{R}_h$, $w = S$ for a simple role S.

A set of role assertions $\mathcal{R}_a$ is called **simple** if all roles R, S appearing in role assertions of the form $\mathsf{Ref}(R)$, $\mathsf{Irr}(R)$, or $\mathsf{Dis}(R,S)$ are simple in $\mathcal{R}$. If $\mathcal{R}$ is clear from the context, we often use "simple" instead of "simple in $\mathcal{R}$".

Definition 4 (Role Box). *A $\mathcal{SRIQ}$-**role box** (Rbox for short) is a set $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$, where $\mathcal{R}_h$ is a regular role hierarchy and $\mathcal{R}_a$ is a finite, simple set of role assertions.*

*An interpretation **satisfies a role box** $\mathcal{R}$ (written $\mathcal{I} \models \mathcal{R}$) if $\mathcal{I} \models R_h$ and $\mathcal{I} \models \phi$ for all role assertions $\phi \in R_a$. Such an interpretation is called a model of $\mathcal{R}$.*

2.2 Concepts and Inference Problems for $\mathcal{SRIQ}$

We are now ready to define the syntax and semantics of $\mathcal{SRIQ}$-concepts.

Definition 5 ($\mathcal{SRIQ}$ Concepts, Tboxes, and Aboxes). *The set of $\mathcal{SRIQ}$-**concepts** is the smallest set such that*

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

*A **general concept inclusion axiom** (GCI) is an expression of the form $C \mathrel{\dot\sqsubseteq} D$ for two $\mathcal{SRIQ}$-concepts C and D. A **Tbox** $\mathcal{T}$ is a finite set of GCIs.*

*An **individual assertion** is of one of the following forms: $a : C$, $(a, b) : R$, $(a, b) : \neg S$, or $a \not\doteq b$, for $a, b \in \mathbf{I}$ (the set of individual names), a (possibly inverse) role R, a (possibly inverse) simple role S, and a $\mathcal{SRIQ}$-concept C. A $\mathcal{SRIQ}$-**Abox** $\mathcal{A}$ is a finite set of individual assertions.*

Note that number restrictions $(\geqslant nS.C)$ and $(\leqslant nS.C)$, the concept $\exists S.\mathsf{Self}$, and negated role assertions $(a, b) : \neg S$, are all restricted to *simple* roles. The reason for this restriction is simple: without it, the satisfiability problem of $\mathcal{SHIQ}$-concepts is already undecidable (11), even for a logic without inverse roles and with only *unqualifying* number restrictions (these are number restrictions of the form $(\geqslant nR.\top)$ and $(\leqslant nR.\top)$). For $\mathcal{SRIQ}$, it is part of future work to determine which of the other restrictions to simple roles are necessary in order to preserve decidability. For example, it should be possible to also allow non-simple roles in negated role assertions $(a, b) : \neg R$ without losing decidability.

Note also that, in the definition of $\mathcal{SRIQ}$-Aboxes, we do not assume the unique name assumption (UNA) (which is commonly assumed in DLs (4)). Rather, by allowing inequalities between individuals in the Abox to be explicitly stated, we increase flexibility while, obviously, the UNA can be regained by explicitly stating $a \not\doteq b$ for every pair $a, b \in \mathbf{I}$ of individuals. Moreover, notice that, in contrast to standard Aboxes, $\mathcal{SRIQ}$-Aboxes can also contain negated role assertions of the form $(a, b) : \neg R$.

Definition 6 (Semantics and Inference Problems).
*Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, concepts C, D, roles R, S, and non-negative integers n, the **extension of complex concepts** is defined inductively by the following equations, where $\sharp M$ denotes the cardinality of a set M:*

$$\begin{array}{rll}
\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \quad \bot^{\mathcal{I}} = \emptyset, & (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & \textit{(top, bottom, negation)} \\
(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, & (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} & \textit{(conjunction, disjunction)} \\
(\exists R.C)^{\mathcal{I}} = & \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} & \textit{(exists restriction)} \\
(\exists R.\mathsf{Self})^{\mathcal{I}} = & \{x \mid \langle x, x\rangle \in R^{\mathcal{I}}\} & \textit{(}\exists R.\mathsf{Self}\textit{-concepts)} \\
(\forall R.C)^{\mathcal{I}} = & \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ implies } y \in C^{\mathcal{I}}\} & \textit{(value restriction)} \\
(\geqslant nR.C)^{\mathcal{I}} = & \{x \mid \sharp\{y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} \geqslant n\} & \textit{(atleast restriction)} \\
(\leqslant nR.C)^{\mathcal{I}} = & \{x \mid \sharp\{y.\langle x, y\rangle \in R^{\mathcal{I}} \textit{ and } y \in C^{\mathcal{I}}\} \leqslant n\} & \textit{(atmost restriction)}
\end{array}$$

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

A concept C is called ***satisfiable*** *if there is an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$. A concept D* ***subsumes*** *a concept C (written $C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for each interpretation. Two concepts are* ***equivalent*** *(written $C \equiv D$) if they are mutually subsuming. The above inference problems can be defined w.r.t. a general role box $\mathcal{R}$ and/or a Tbox $\mathcal{T}$ 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 if $x \in C^{\mathcal{I}}$.*

An interpretation $\mathcal{I}$ ***satisfies*** *(is a model of)* ***an Abox*** *$\mathcal{A}$ ($\mathcal{I} \models \mathcal{A}$) if for all individual assertions $\phi \in \mathcal{A}$ we have $\mathcal{I} \models \phi$, where*

$$\begin{array}{lll} \mathcal{I} \models a{:}C & \textit{if} & a^{\mathcal{I}} \in C^{\mathcal{I}}; \\ \mathcal{I} \models a \neq b & \textit{if} & a^{\mathcal{I}} \neq b^{\mathcal{I}}; \\ \mathcal{I} \models (a,b){:}R & \textit{if} & \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}; \\ \mathcal{I} \models (a,b){:}\neg R & \textit{if} & \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}. \end{array}$$

An Abox $\mathcal{A}$ is ***consistent*** *with respect to an Rbox $\mathcal{R}$ and a Tbox $\mathcal{T}$ if there is a model $\mathcal{I}$ for $\mathcal{R}$ and $\mathcal{T}$ such that $\mathcal{I} \models \mathcal{A}$.*

For DLs that are closed under negation, subsumption and (un)satisfiability of concepts can be mutually reduced: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable, and C is unsatisfiable iff $C \sqsubseteq \bot$. Furthermore, a concept C is satisfiable iff the Abox $\{a{:}C\}$ is consistent.

It is straightforward to extend these reductions to Rboxes and Tboxes. In contrast, the reduction of inference problems w.r.t. a Tbox to pure concept inference problems (possibly w.r.t. a role hierarchy), deserves special care: in (2; 13; 3), the *internalisation* of GCIs is introduced, a technique that realises exactly this reduction. For $\mathcal{SRIQ}$, 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 $\mathcal{T}$, $\mathcal{A}$, or $\mathcal{R}$ and their respective inverses.

Lemma 1. *Let C and D be concepts, $\mathcal{A}$ an Abox, $\mathcal{T}$ a Tbox, and $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ an Rbox. We define*

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

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

$$\mathcal{R}_h^U := \mathcal{R}_h \cup \{R \dot{\sqsubseteq} U, \mathsf{Inv}(R) \dot{\sqsubseteq} U \mid R \textit{ occurs in } C,\ D,\ \mathcal{T},\ \mathcal{A}, \textit{ or } \mathcal{R}\},$$

$$\mathcal{R}_a^U := \mathcal{R}_a \cup \{\mathsf{Tra}(U)\}, \textit{ and } \mathcal{R}^U := \mathcal{R}_h^U \cup \mathcal{R}_a^U.$$

- *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. $\mathcal{R}^U$.*

– *$\mathcal{A}$ is satisfiable w.r.t. $\mathcal{R}$ and $\mathcal{T}$ iff $\mathcal{A} \cup \{a : C_\mathcal{T} \sqcap \forall U.C_\mathcal{T} \mid$ a occurs in $\mathcal{A}\}$ is satisfiable w.r.t. $\mathcal{R}^U$.*

The proof of Lemma 1 is similar to the ones that can be found in (13; 2). Most importantly, it must be shown that (a): if a $\mathcal{SRIQ}$-concept C is satisfiable with respect to a Tbox $\mathcal{T}$ and an Rbox $\mathcal{R}$, 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.

Now, note also that, instead of having a role assertion $\mathsf{Ref}(R) \in \mathcal{R}_a$, we can add, equivalently, the GCI $\exists R.\top \mathrel{\dot\sqsubseteq} \exists R.\mathsf{Self}$ to $\mathcal{T}$, which can in turn be internalised. Likewise, instead of asserting $\mathsf{Irr}(R)$, we can, equivalently, add the GCI $\top \mathrel{\dot\sqsubseteq} \neg\exists R.\mathsf{Self}$. Thus, we arrive at the following theorem:

Theorem 1.

1. *Satisfiability and subsumption of $\mathcal{SRIQ}$-concepts w.r.t. Tboxes and Rboxes are polynomially reducible to (un)satisfiability of $\mathcal{SRIQ}$-concepts w.r.t. Rboxes.*
2. *Consistency of $\mathcal{SRIQ}$-Aboxes w.r.t. Tboxes and Rboxes is polynomially reducible to consistency of $\mathcal{SRIQ}$-Aboxes w.r.t. Rboxes.*
3. *W.l.o.g., we can assume that Rboxes do not contain role assertions of the form $\mathsf{Irr}(R)$, $\mathsf{Ref}(R)$, $\mathsf{Tra}(R)$, or $\mathsf{Sym}(R)$.*

With Theorem 1, all standard inference problems for $\mathcal{SRIQ}$-concepts and Aboxes can be reduced to the problem of determining the consistency of a $\mathcal{SRIQ}$-Abox w.r.t. to an Rbox, where we can assume w.l.o.g. that all role assertions in the Rbox are of the form $\mathsf{Dis}(R, S)$—we call such an Rbox **reduced**.

3 $\mathcal{SRIQ}$ is Decidable

In this section, we show that $\mathcal{SRIQ}$ is decidable. We present a tableau-based algorithm that decides the consistency of a $\mathcal{SRIQ}$-Abox w.r.t. a reduced Rbox, and therefore also all standard inference problems as discussed above, see Theorem 1. Therefore, in the following, by Rbox we always mean *reduced* Rbox.

The algorithm tries to construct, given a $\mathcal{SRIQ}$-Abox $\mathcal{A}$, a *tableau* for $\mathcal{A}$, that is, an abstraction of a model of $\mathcal{A}$. Given the appropriate notion of a tableau, it is then quite straightforward to prove that the algorithm is a decision procedure for $\mathcal{SRIQ}$-satisfiability.

Before specifying this algorithm, we translate a role hierarchy $\mathcal{R}_h$ into non-deterministic automata which are used both in the definition of a tableau and in the tableau algorithm. Intuitively, an automaton is used to memorise the path between an object x that has to satisfy a concept of the form $\forall R.C$ and other objects, and then to determine which of these objects must satisfy C.[3]

[3] This technique together with the relationship between automata and regular languages is the reason why we called these role hierarchies "regular".

For the following considerations, it is worthwhile to recall that, for a string $w = R_1 \dots R_m$ and R_i roles, $\mathsf{Inv}(w) = \mathsf{Inv}(R_m) \dots \mathsf{Inv}(R_1)$. The following Lemma is a direct consequence of the definition of the semantics.

Lemma 2. *If $\mathcal{I}$ is a model of $\mathcal{R}_h$ with $S^- \dot{\sqsubseteq} S \in \mathcal{R}_h$ and $w \dot{\sqsubseteq} S \in \mathcal{R}_h$, then $\mathsf{Inv}(w)^\mathcal{I} \subseteq S^\mathcal{I}$.*

3.1 Translating RIAs into Automata

The technique used in this chapter is identical to the one presented in (9), and repeated here only to make this paper self-contained. First, we will define, for a regular role hierarchy $\mathcal{R}_h$ and a (possibly inverse) role S occurring in $\mathcal{R}_h$, a non-deterministic finite automaton (NFA) $\mathcal{B}_S$ which captures all implications between (paths of) roles and S that are consequences of $\mathcal{R}_h$. To make this clear, before we define $\mathcal{B}_S$, we formulate the lemma which we are going to prove for it.

Proposition 1. *$\mathcal{I}$ is a model of $\mathcal{R}_h$ if and only if, for each (possibly inverse) role S occurring in $\mathcal{R}_h$, each word $w \in L(\mathcal{B}_S)$, and each $\langle x, y\rangle \in w^\mathcal{I}$, we have $\langle x, y\rangle \in S^\mathcal{I}$.*

In the following, we use NFAs with ε-transitions in a rather informal way (see, e.g., (6) for more details), e.g., we use $p \overset{R}{\to} q$ to denote that there is a transition from a state p to a state q with the letter R instead of introducing transition relations formally. The automata $\mathcal{B}_S$ are defined in three steps.

Definition 7. *Let $\mathcal{A}$ be a $\mathcal{SRIQ}$-Abox and $\mathcal{R}$ a reduced Rbox which is $\prec$-regular. For each role name R occurring in $\mathcal{R}$ or $\mathcal{A}$, we first define the NFA $\mathcal{A}_R$ as follows: $\mathcal{A}_R$ contains a state i_R and a state f_R with the transition $i_R \overset{R}{\to} f_R$. The state i_R is the only initial state and f_R is the only final state. Moreover, for each $w \dot{\sqsubseteq} R \in \mathcal{R}$, $\mathcal{A}_R$ contains the following states and transitions:*

1. *if $w = RR$, then $\mathcal{A}_R$ contains $f_R \overset{\varepsilon}{\to} i_R$, and*
2. *if $w = R_1 \cdots R_n$ and $R_1 \neq R \neq R_n$, then $\mathcal{A}_R$ contains*

$$i_R \overset{\varepsilon}{\to} i_w \overset{R_1}{\to} f_w^1 \overset{R_2}{\to} f_w^2 \overset{R_3}{\to} \dots \overset{R_n}{\to} f_w^n \overset{\varepsilon}{\to} f_R,$$

3. *if $w = RR_2 \cdots R_n$, then $\mathcal{A}_R$ contains*

$$f_R \overset{\varepsilon}{\to} i_w \overset{R_2}{\to} f_w^2 \overset{R_3}{\to} f_w^3 \overset{R_4}{\to} \dots \overset{R_n}{\to} f_w^n \overset{\varepsilon}{\to} f_R,$$

4. *if $w = R_1 \cdots R_{n-1}R$, then $\mathcal{A}_R$ contains*

$$i_R \overset{\varepsilon}{\to} i_w \overset{R_1}{\to} f_w^1 \overset{R_2}{\to} f_w^2 \overset{R_3}{\to} \dots \overset{R_{n-1}}{\to} f_w^{n-1} \overset{\varepsilon}{\to} i_R,$$

where all f_w^i, i_w are assumed to be distinct.

In the next step, we use a mirrored copy *of NFAs: this is a copy of an NFA in which we have carried out the following modifications: we*

- *make final states to non-final but initial states,*
- *make initial states to non-initial but final states,*
- *replace each transition* $p \xrightarrow{S} q$ *for* S *a (possibly inverse) role* S *with* $q \xrightarrow{\mathsf{Inv}(S)} p$*, and*
- *replace each transition* $p \xrightarrow{\varepsilon} q$ *with* $q \xrightarrow{\varepsilon} p$*.*

Secondly, we define the NFAs $\hat{\mathcal{A}}_R$ *as follows:*

- *if* $R^- \dot{\sqsubseteq} R \notin \mathcal{R}$*, then* $\hat{\mathcal{A}}_R := \mathcal{A}_R$*,*
- *if* $R^- \dot{\sqsubseteq} R \in \mathcal{R}$*, then* $\hat{\mathcal{A}}_R$ *is obtained as follows: first, take the disjoint union*[4] *of* $\mathcal{A}_S$ *with a mirrored copy of* $\mathcal{A}_S$*. Secondly, make* i_R *the only initial state,* f_R *the only final state. Finally, for* f'_R *the copy of* f_R *and* i'_R *the copy of* i_R*, add transitions* $i_R \xrightarrow{\varepsilon} f'_R$*,* $f'_R \xrightarrow{\varepsilon} i_R$*,* $i'_R \xrightarrow{\varepsilon} f_R$*, and* $f_R \xrightarrow{\varepsilon} i'_R$*.*

Thirdly, the NFAs $\mathcal{B}_R$ *are defined inductively over* $\prec$*:*

- *if* R *is minimal w.r.t.* $\prec$ *(i.e., there is no* R' *with* $R' \prec R$*), we set* $\mathcal{B}_R := \hat{\mathcal{A}}_R$*.*
- *otherwise,* $\mathcal{B}_R$ *is the disjoint union of* $\hat{\mathcal{A}}_R$ *with a copy* $\mathcal{B}'_S$ *of* $\mathcal{B}_S$ *for each transition* $p \xrightarrow{S} q$ *in* $\hat{\mathcal{A}}_R$ *with* $S \neq R$*. Moreover, for each such transition, we add* ε*-transitions from* p *to the initial state in* $\mathcal{B}'_S$ *and from the final state in* $\mathcal{B}'_S$ *to* q*, and we make* i_R *the only initial state and* f_R *the only final state in* $\mathcal{B}_R$*.*

Finally, the automaton $\mathcal{B}_{R^-}$ *is a mirrored copy of* $\mathcal{B}_R$*.*

Please note that the inductive definition of $\mathcal{B}_R$ is well-defined since the acyclic relation $\prec$ is used to restrict the dependencies between roles.

We have kept the construction of $\mathcal{B}_S$ as simple as possible. If one wants to construct an equivalent NFA without ε-transitions or which is deterministic, then there are well-known techniques to do this (6). Recall that elimination of ε-transitions can be carried out without increasing the number of an automaton's states, whereas determinisation might yield an exponential blow-up. However, as we will see later, this determinisation will happen anyway "on-the-fly" in the tableau algorithm, and thus has no influence on the complexity, see (9) for a discussion.

Lemma 3. *For* R *a role, the size of* $\mathcal{B}_R$ *is bounded exponentially in the depth*

$$d_{\mathcal{R}} := \max\{n \mid \textit{there are } S_1 \prec \ldots \prec S_n, u_i, v_i \textit{ with } u_i S_{i-1} v_i \dot{\sqsubseteq} S_i \in \mathcal{R}\}$$

and thus in the size of $\mathcal{R}$*. Moreover, there are* $\mathcal{R}$ *and* R *such that the number of states in* $\mathcal{B}_R$ *is* $2^{d_{\mathcal{R}}}$*.*

In (9), certain further syntactic restrictions of role hierarchies were considered (there called *simple* role hierarchies) that avoid this exponential blow-up. We conjecture that without some such further restriction, this blow-up is unavoidable. Next, we will repeat a technical Lemma from (9) which we will use later, and refer the reader to (9) for its proof and the proof of Proposition 1.

[4] A disjoint union of two automata is the disjoint union of their states, transition relations, etc.

Lemma 4. *1.* $S \in L(\mathcal{B}_S)$ *and, if* $w \mathrel{\dot{\sqsubseteq}} S \in \mathcal{R}$*, then* $w \in L(\mathcal{B}_S)$*.*
2. If S *is a simple role, then* $L(\mathcal{B}_S) = \{R \mid R \mathrel{\underline{\ast}} S\}$*.*
3. If $\overleftarrow{\mathcal{A}}$ *is a mirrored copy of an NFA* $\mathcal{A}$*, then* $L(\overleftarrow{\mathcal{A}}) = \{\mathsf{Inv}(w) \mid w \in L(\mathcal{A})\}$*.*

3.2 A Tableau for $\mathcal{SRIQ}$

In the following, if not stated otherwise, C, D (possibly with subscripts) denote $\mathcal{SRIQ}$-concepts, R, S (possibly with subscripts) roles, $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ an Rbox, $\mathcal{A}$ an Abox, $\mathbf{R}_{\mathcal{A}}$ the set of roles occurring in $\mathcal{A}$ and $\mathcal{R}$ together with their inverses, and $\mathbf{I}_{\mathcal{A}}$ the set of individuals occurring in $\mathcal{A}$. Furthermore, as noted in Theorem 8, we can (and will from now on) assume w.l.o.g. that no role assertions of the form $\mathsf{Irr}(R)$, $\mathsf{Ref}(R)$, $\mathsf{Tra}(R)$, or $\mathsf{Sym}(R)$ appear in $\mathcal{R}_a$.

We start by defining $\mathsf{fclos}(C_0, \mathcal{R})$, the *closure* of a concept C_0 w.r.t. a regular role hierarchy $\mathcal{R}$. Intuitively, this contains all relevant sub-concepts of C_0 together with universal value restrictions over sets of role paths described by an NFA. We use NFAs in universal value restrictions to memorise the path between an object that has to satisfy a value restriction and other objects. To do this, we "push" this NFA-value restriction along this path while the NFA gets "updated" with the path taken so far. For this "update", we use the following definition.

Definition 8. *For* $\mathcal{B}$ *an NFA and* q *a state of* $\mathcal{B}$*,* $\mathcal{B}(q)$ *denotes the NFA obtained from* $\mathcal{B}$ *by making* q *the (only) initial state of* $\mathcal{B}$*, and we use* $q \xrightarrow{S} q' \in \mathcal{B}$ *to denote that* $\mathcal{B}$ *has a transition* $q \xrightarrow{S} q'$*.*

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

$$
\begin{array}{rlrl}
\neg(\exists R.C) & \equiv (\forall R.\neg C) & \neg(\forall R.C) & \equiv (\exists R.\neg C) \\
\neg(\leqslant nR.C) & \equiv (\geqslant (n+1)R.C) & \neg(\geqslant (n+1)R.C) & \equiv (\leqslant nR.C) \\
 & & \neg(\geqslant 0R.C) & \equiv \bot
\end{array}
$$

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_0 and that is closed under sub-concepts and $\dot{\neg}$. The set $\mathsf{fclos}(C_0, \mathcal{R})$ is then defined as follows:

$$\mathsf{fclos}(C_0, \mathcal{R}) := \mathsf{clos}(C_0) \cup \{\forall \mathcal{B}_S(q).D \mid \forall S.D \in \mathsf{clos}(C_0) \text{ and } \mathcal{B}_S \text{ has a state } q\}.$$

It is not hard to show and well-known that the size of $\mathsf{clos}(C_0)$ is linear in the size of C_0. For the size of $\mathsf{fclos}(C_0, \mathcal{R})$, we have seen in Lemma 3 that, for a role S, the size of $\mathcal{B}_S$ can be exponential in the depth of $\mathcal{R}$. Since there are at most linearly many concepts $\forall S.D$, this yields a bound for the cardinality of $\mathsf{fclos}(C_0, \mathcal{R})$ that is exponential in the depth of $\mathcal{R}$ and linear in the size of C_0.

Next, we define

$$\mathsf{fclos}(\mathcal{A},\mathcal{R}) := \bigcup_{a:C\in\mathcal{A}} \mathsf{fclos}(C,\mathcal{R}).$$

The size of $\mathsf{fclos}(\mathcal{A},\mathcal{R})$ is linear in $\mathcal{A}$ and exponential in the depth of $\mathcal{R}$.

Definition 9. *$T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ is a* tableau *for $\mathcal{A}$ w.r.t. $\mathcal{R}$ iff*

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

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

(P1a) if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$ (C atomic or $\exists R.\mathsf{Self}$),
(P1b) $\top \in \mathcal{L}(s)$, and $\bot \notin \mathcal{L}(s)$, for all s,
(P1c) if $\exists R.\mathsf{Self} \in \mathcal{L}(s)$, then $\langle s, s\rangle \in \mathcal{E}(R)$,
(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 $\forall\mathcal{B}(p).C \in \mathcal{L}(s)$, $\langle s, t\rangle \in \mathcal{E}(S)$, and $p \xrightarrow{S} q \in \mathcal{B}(p)$, then $\forall\mathcal{B}(q).C \in \mathcal{L}(t)$,
(P4b) if $\forall\mathcal{B}.C \in \mathcal{L}(s)$ and $\varepsilon \in L(\mathcal{B})$, 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 $\forall\mathcal{B}_S.C \in \mathcal{L}(s)$,
(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 $\dot\neg C \in \mathcal{L}(t)$,
(P11) if $a : C \in \mathcal{A}$, then $C \in \mathcal{L}(\mathcal{I}(a))$,
(P12) if $(a, b) : R \in \mathcal{A}$, then $\langle\mathcal{I}(a), \mathcal{I}(b)\rangle \in \mathcal{E}(R)$,
(P13) if $(a, b) : \neg R \in \mathcal{A}$, then $\langle\mathcal{I}(a), \mathcal{I}(b)\rangle \notin \mathcal{E}(R)$,
(P14) if $a \neq b \in \mathcal{A}$, then $\mathcal{I}(a) \neq \mathcal{I}(b)$,
(P15) if $\mathsf{Dis}(R, S) \in \mathcal{R}$, then $\mathcal{E}(R) \cap \mathcal{E}(S) = \emptyset$,
(P16) if $\langle s, t\rangle \in \mathcal{E}(R)$ and $R \mathrel{\underline{\circledast}} S$, then $\langle s, t\rangle \in \mathcal{E}(S)$,

where, in (P8) and (P9),

$$S^T(s, C) := \{t \in \mathbf{S} \mid \langle s, t\rangle \in \mathcal{E}(S') \text{ for some } S' \in L(\mathcal{B}_S) \text{ and } C \in \mathcal{L}(t)\}.$$

Lemma 5. *A $\mathcal{SRIQ}$-Abox $\mathcal{A}$ is consistent w.r.t. $\mathcal{R}$ iff there exists a tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$.*

Proof: For the *if* direction, let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ be a tableau for $\mathcal{A}$ 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 $\mathcal{A}$ and $\mathcal{R}$ can be defined as follows:

we set $\Delta^{\mathcal{I}} := \mathbf{S}$, $C^{\mathcal{I}} := \{s \mid C \in \mathcal{L}(s)\}$ for concept names C in $\mathsf{fclos}(\mathcal{A}, \mathcal{R})$, $a^{\mathcal{I}} := \mathcal{J}(a)$ for individual names $a \in \mathbf{I}_{\mathcal{A}}$, and for roles names $R \in \mathbf{R}_{\mathcal{A}}$, we set

$$\begin{aligned} R^{\mathcal{I}} := \{\langle s_0, s_n\rangle \in (\Delta^{\mathcal{I}})^2 \mid & \text{ there are } s_1, \ldots, s_{n-1} \text{ with } \langle s_i, s_{i+1}\rangle \in \mathcal{E}(S_{i+1}) \\ & \text{ for } 0 \leq i \leq n-1 \text{ and } S_1 \cdots S_n \in L(\mathcal{B}_R)\} \end{aligned}$$

The semantics of complex concepts is given through the definition of the $\mathcal{SRIQ}$-semantics. Due to Lemma 4.3 and (P7), the semantics of inverse roles can either be given directly as for role names, or by setting $(R^-)^{\mathcal{I}} := \{\langle y, x\rangle \mid \langle x, y\rangle \in R^{\mathcal{I}}\}$.

We have to show that $\mathcal{I}$ is a model of $\mathcal{R}$ and $\mathcal{A}$. We begin by showing that $\mathcal{I} \models \mathcal{R}$. First, we look at role assertions. Remember that we assumed that $\mathcal{R}$ is reduced, and thus we only have to deal with role disjointness assertions of the form $\mathsf{Dis}(R, S)$. Consider an assertion $\mathsf{Dis}(R, S) \in \mathcal{R}$. By definition of $\mathcal{SRIQ}$-Rboxes, both R and S are simple roles, and (P15) implies $\mathcal{E}(R) \cap \mathcal{E}(S) = \emptyset$. Moreover, we have, by definition of $\mathcal{I}$, Lemma 4.2, (P7), and (P16) that, for T a simple role, $T^{\mathcal{I}} = \mathcal{E}(T)$. Hence $R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$, and thus $\mathcal{I}$ satisfies each role assertion in $\mathcal{R}_a$.

Next, we show that $\mathcal{I} \models \mathcal{R}_h$. Due to Proposition 1, it suffices to prove that, for each (possibly inverse) role S, each word $w \in L(\mathcal{B}_S)$, and each $\langle x, y\rangle \in w^{\mathcal{I}}$, we have $\langle x, y\rangle \in S^{\mathcal{I}}$.

Let $w \in L(\mathcal{B}_S)$ and $\langle x, y\rangle \in w^{\mathcal{I}}$. For $w = S_1 \ldots S_n$, this implies the existence of y_i such that $y_0 = x$, $y_n = y$, and $\langle y_{i-1}, y_i\rangle \in S_i^{\mathcal{I}}$ for each $1 \leq i \leq n$. For each i, we define a word w_i as follows:

- if $\langle y_{i-1}, y_i\rangle \in \mathcal{E}(S_i)$, then set $w_i := S_i$.
- otherwise, there is some $v_i = T_1^{(i)} \ldots T_{n_i}^{(i)} \in L(\mathcal{B}_{S_i})$ and there are $y_j^{(i)}$ such that $y_{i-1} = y_0^{(i)}$, $y_i = y_{n_i}^{(i)}$, and $\langle y_{j-1}^{(i)}, y_j^{(i)}\rangle \in \mathcal{E}(T_j^{(i)})$ for each $1 \leq j \leq n_i$. In this case, we set $w_i := v_i$.

Let $\hat{w} := w_1 \ldots w_n$. By construction of $\mathcal{B}_S$ from $\hat{\mathcal{A}}_S$, $w \in L(\mathcal{B}_S)$ implies that $\hat{w} \in L(\mathcal{B}_S)$. For $\hat{w} = U_1 \ldots U_{n'}$, we can thus re-name the y_i and $y_j^{(i)}$ to z_i such that we have $z_0 = x$, $z_n = y$, and $\langle z_{i-1}, z_i\rangle \in \mathcal{E}(U_i)$. Hence, by definition of $\cdot^{\mathcal{I}}$, we have $\langle x, y\rangle \in S^{\mathcal{I}}$.

Secondly, we prove that $\mathcal{I}$ is a model of $\mathcal{A}$. We show that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$ for each $s \in \mathbf{S}$ and each $C \in \mathsf{fclos}(\mathcal{A}, \mathcal{R})$. Together with (P11)–(P14), this implies that $\mathcal{I}$ is a model of $\mathcal{A}$. This proof can be given by induction on the length of concepts, where we count neither negation nor integers in number restrictions. The only interesting cases are $C = ({\leqslant}nS.E)$, $C = \forall S.E$, and $C = \exists R.\mathsf{Self}$ (for the other cases, see (12; 8)):

- If $({\leqslant}nS.E) \in \mathcal{L}(s)$, then (P8) implies that $\#S^T(s, E) \leq n$. Moreover, since S is simple, Lemma 4.2 implies that $L(\mathcal{B}_S) = \{S' \mid S' \mathrel{\underline{\circledast}} S\}$, and (P16) implies that $S^{\mathcal{I}} = \mathcal{E}(S)$. Hence (P10) implies that, for all t, if $\langle s, t\rangle \in S^{\mathcal{I}}$, then $E \in \mathcal{L}(t)$ or $\dot{\neg}E \in \mathcal{L}(t)$. By induction $E^{\mathcal{I}} = \{t \mid E \in \mathcal{L}(t)\}$, and thus $s \in ({\leqslant}nS.E)^{\mathcal{I}}$.

- Let $\forall S.E \in \mathcal{L}(s)$ and $\langle s,t\rangle \in S^{\mathcal{I}}$. From (P6) we have that $\forall \mathcal{B}_S.E \in \mathcal{L}(s)$. By definition of $S^{\mathcal{I}}$, there are $S_1 \dots S_n \in L(\mathcal{B}_S)$ and s_i with $s = s_0$, $t = s_n$, and $\langle s_{i-1}, s_i\rangle \in \mathcal{E}(S_i)$. Applying (P4a) n times, this yields $\forall \mathcal{B}_S(q).E \in \mathcal{L}(t)$ for q a final state of $\mathcal{B}_S$. Thus (P4b) implies that $E \in \mathcal{L}(t)$. By induction, $t \in E^{\mathcal{I}}$, and thus $s \in (\forall S.E)^{\mathcal{I}}$.
- Let $\exists R.\mathsf{Self} \in \mathcal{L}(s)$. Then, by (P1c), $\langle s,s\rangle \in \mathcal{E}(R)$ and, since $R \in L(B_R)$ and by definition of $\mathcal{I}$, we have $\langle s,s\rangle \in R^{\mathcal{I}}$. It follows that $s \in (\exists R.\mathsf{Self})^{\mathcal{I}}$.

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

$$
\begin{aligned}
\mathbf{S} &:= \Delta^{\mathcal{I}},\\
\mathcal{J}(a) &:= a^{\mathcal{I}},\\
\mathcal{E}(R) &:= R^{\mathcal{I}}, \text{ and}\\
\mathcal{L}(s) &:= \{C \in \mathsf{clos}(\mathcal{A}) \mid s \in C^{\mathcal{I}}\} \cup\\
&\quad\ \{\forall \mathcal{B}_S.C \mid \forall S.C \in \mathsf{clos}(\mathcal{A}) \text{ and } s \in (\forall S.C)^{\mathcal{I}}\} \cup\\
&\quad\ \{\forall \mathcal{B}_R(q).C \in \mathsf{fclos}(\mathcal{A},\mathcal{R}) \mid \text{ for all } S_1 \cdots S_n \in L(\mathcal{B}_R(q)),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad s \in (\forall S_1.\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}} \text{ and}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if } \varepsilon \in L(\mathcal{B}_R(q)), \text{ then } s \in C^{\mathcal{I}}\}
\end{aligned}
$$

We have to show that T satisfies each (Pi). We restrict our attention to the only new cases.

For (P6), if $\forall S.C \in \mathcal{L}(s)$, then $s \in (\forall S.C)^{\mathcal{I}}$ and thus $\forall \mathcal{B}_S.C \in \mathcal{L}(s)$ by definition of T.

For (P4a), let $\forall \mathcal{B}(p).C \in \mathcal{L}(s)$ and $\langle s,t\rangle \in \mathcal{E}(S) = S^{\mathcal{I}}$. Assume that there is a transition $p \xrightarrow{S} q$ in $\mathcal{B}(p)$ and $\forall \mathcal{B}(q).C \notin \mathcal{L}(t)$. By definition of T, this can have two reasons:

- there is a word $S_2 \dots S_n \in L(\mathcal{B}(q))$ and $t \notin (\forall S_2.\dots\forall S_n.C)^{\mathcal{I}}$. However, this implies that $SS_2 \dots S_n \in L(\mathcal{B}(p))$ and thus that $s \in (\forall S.\forall S_2.\dots\forall S_n.C)^{\mathcal{I}}$, which contradicts, together with $\langle s,t\rangle \in S^{\mathcal{I}}$, the definition of the semantics of $\mathcal{SRIQ}$ concepts.
- $\varepsilon \in L(\mathcal{B}(q))$ and $t \notin C^{\mathcal{I}}$. This implies that $S \in L(\mathcal{B}(p))$ and thus contradicts $s \in (\forall S.C)^{\mathcal{I}}$.

Hence $\forall \mathcal{B}(q).C \notin \mathcal{L}(t)$.

For (P4b), $\varepsilon \in L(\mathcal{B}(p))$ implies $s \in C^{\mathcal{I}}$ by definition of T, and thus $C \in \mathcal{L}(s)$.

Finally, (P11) to (P16) follow immediately from the definition of the semantics. □

3.3 The Tableau Algorithm

In this section, we present a tableau algorithm that decides consistency of $\mathcal{SRIQ}$-Aboxes w.r.t. Rboxes, and thus, using Lemma 1, also w.r.t. Tboxes.

Since Aboxes usually involve several individuals with arbitrary role relationships between them, the completion algorithm presented works on *forests* rather than on *trees* (which suffice for deciding concept satisfiability). A forest is a collection of trees whose root nodes correspond to the individuals appearing in the input Abox and which form an arbitrarily connected graph according to the role assertions stated in the Abox.

The algorithm generates a *completion forest*, a structure that, if complete and clash-free, can be unravelled to an (infinite) tableau for the input Abox and Rbox. Moreover, it is shown that the algorithm returns a complete and clash-free completion forest for $\mathcal{A}$ and $\mathcal{R}$ if and only if there exists a tableau for $\mathcal{A}$ and $\mathcal{R}$, and thus with Lemma 5, if and only if the Abox $\mathcal{A}$ is consistent w.r.t. $\mathcal{R}$.

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 (12). The blocking conditions as they are presented here are, clearly, too strict. As a consequence, blocking may occur later than necessary, and thus we end up with a search space that is larger than necessary. In (8), we have shown how to loosen the blocking condition for $\mathcal{SHIQ}$ while retaining correctness of the algorithm. Here, we focus on the decidability of $\mathcal{SRIQ}$, and defer a similar loosening for $\mathcal{SRIQ}$ to future work.

Definition 10. *A* ***completion forest*** $\mathbf{F}$ *for a* $\mathcal{SRIQ}$*-Abox* $\mathcal{A}$ *and an Rbox* $\mathcal{R}$ *is a collection of trees whose distinguished* ***root nodes*** *can be connected arbitrarily. Moreover, each node* x *is labelled with a set* $\mathcal{L}(x) \subseteq \mathsf{fclos}(\mathcal{A}, \mathcal{R})$ *and each edge* $\langle x, y\rangle$ *from a node* x *to its* ***successor*** y *is labelled with a non-empty set* $\mathcal{L}(\langle x, y\rangle)$ *of (possibly inverse and possibly negated) roles occurring in* $\mathcal{A}$ *and* $\mathcal{R}$*. Finally, completion forests come with an explicit* ***inequality relation*** $\neq$ *on nodes which is implicitly assumed to be symmetric.*

Let x *and* y *be nodes in* $\mathbf{F}$ *and* R *a role. If* $R' \mathrel{\underline{\sqsubseteq}_*} R$ *and* $R' \in \mathcal{L}(\langle x, y\rangle)$*, then* y *is called an* R*-****successor*** *of* x*.*

If y *is an* R*-successor of* x *or* x *is an* $\mathsf{Inv}(R)$*-successor of* y*, then* y *is called an* R*-****neighbour*** *of* x*. Moreover, a node* x *is a* ***neighbour*** *of* y*, if it is an* R*-neighbour for some role* R*.* ***Successors****,* ***predecessors****,* ***ancestors****, and* ***descendants*** *are defined as usual.*

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

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

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

1. *none of* x'*,* y *and* y' *is a root node,*
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. $\mathcal{L}(\langle x', x\rangle) = \mathcal{L}(\langle y', y\rangle)$*.*

In this case, we say that y ***blocks*** x*.*

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

Given a non-empty $\mathcal{SRIQ}$-Abox $\mathcal{A}$ and a reduced Rbox $\mathcal{R}$, the tableau algorithm is initialised with the completion forest $\mathbf{F}_{\mathcal{A},\mathcal{R}}$ defined as follows:

- *for each individual a occurring in $\mathcal{A}$, $\mathbf{F}_{\mathcal{A},\mathcal{R}}$ contains a **root node** x_a,*
- *if $(a,b){:}R \in \mathcal{A}$ or $(a,b){:}\neg R \in \mathcal{A}$, then $\mathbf{F}_{\mathcal{A},\mathcal{R}}$ contains an edge $\langle x_a, x_b\rangle$,*
- *if $a \neq b \in \mathcal{A}$, then $x_a \neq x_b$ is in $\mathbf{F}_{\mathcal{A},\mathcal{R}}$,*
- *$\mathcal{L}(x_a) := \{C \mid a{:}C \in \mathcal{A}\}$, and*
- *$\mathcal{L}(\langle x_a, x_b\rangle) := \{R \mid (a,b){:}R \in \mathcal{A}\} \cup \{\neg R \mid (a,b){:}\neg R \in \mathcal{A}\}$*

*A completion forest $\mathbf{F}$ is said to **contain a clash** if there are nodes x and y such that*

1. *$\perp \in \mathcal{L}(x)$, or*
2. *for some concept name A, $\{A, \neg A\} \subseteq \mathcal{L}(x)$, or*
3. *x is an S-neighbour of x and $\neg\exists S.\mathsf{Self} \in \mathcal{L}(x)$,*
4. *x and y are root nodes, y is an R-neighbour of x, and $\neg R \in \mathcal{L}(\langle x, y\rangle)$, or*
5. *there is some $\mathsf{Dis}(R,S) \in \mathcal{R}_a$ and y is an R- and an S-neighbour of x, or*
6. *there is some concept $(\leqslant nS.C) \in \mathcal{L}(x)$ and $\{y_0, \ldots, y_n\} \subseteq S^{\mathbf{F}}(x,C)$ with $y_i \neq y_j$ for all $0 \leq i < j \leq n$.*

*A completion forest that does not contain a clash is called **clash-free**. A completion forest is **complete** if none of the rules from Figure 1 can be applied to it.*

When started with a non-empty Abox $\mathcal{A}$ and a reduced Rbox $\mathcal{R}$, the tableau algorithm initialises $\mathbf{F}_{\mathcal{A},\mathcal{R}}$ and repeatedly applies the expansion rules from Figure 1 to it, stopping when a clash occurs, and applying the shrinking rules eagerly, i.e., the $\leq$- and the $\leq_r$-rule are applied with highest priority. The algorithm answers "$\mathcal{A}$ is satisfiable w.r.t. $\mathcal{R}$" if and only if the expansion rules can be applied in such a way that they yield a complete and clash-free completion forest, and "$\mathcal{A}$ is unsatisfiable w.r.t. $\mathcal{R}$" otherwise.

All but the Self-rule have been used before for fragments of $\mathcal{SRIQ}$, see (11; 8; 9), and the three $\forall_i$-rules are the obvious counterparts to the tableau conditions (P4a), (P4b), and (P6).

As usual, we prove termination, soundness, and completeness of the tableau algorithm to show that it indeed decides consistency of $\mathcal{SRIQ}$-Aboxes w.r.t. Rboxes.

Lemma 6. *Let $\mathcal{A}$ be a $\mathcal{SRIQ}$-Abox and $\mathcal{R}$ a reduced Rbox. The tableau algorithm terminates when started for $\mathcal{A}$ and $\mathcal{R}$.*

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

1. Nodes are labelled with subsets of $\mathsf{fclos}(\mathcal{A},\mathcal{R})$ and edges with sets of roles occurring in $\mathcal{A}$ and $\mathcal{R}$ plus possibly negated roles for roles between root nodes. Hence, there are at most 2^{2mn} different possible labellings for a pair

$\sqcap$-rule: if $C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$,
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C_1, C_2\}$

$\sqcup$-rule: if $C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and
$\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{C_1, C_2\}$

$\exists$-rule: if $\exists S.C \in \mathcal{L}(x)$, x is not blocked, and
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 $\mathcal{L}(y) := \{C\}$

Self-rule: if $\exists S.\mathsf{Self} \in \mathcal{L}(x)$, x is not blocked, and $S \notin \mathcal{L}(\langle x, x\rangle)$
then add an edge $\langle x, x\rangle$ if it does not yet exist, and
set $\mathcal{L}(\langle x, x\rangle) \longrightarrow \mathcal{L}(\langle x, x\rangle) \cup \{S\}$

$\forall_1$-rule: if $\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and
$\forall \mathcal{B}_S.C \notin \mathcal{L}(x)$
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{\forall \mathcal{B}_S.C\}$

$\forall_2$-rule: if $\forall \mathcal{B}(p).C \in \mathcal{L}(x)$, x is not indirectly blocked, $p \xrightarrow{S} q$ in $\mathcal{B}(p)$,
and there is an S-neighbour y of x with $\forall \mathcal{B}(q).C \notin \mathcal{L}(y)$,
then $\mathcal{L}(y) \longrightarrow \mathcal{L}(y) \cup \{\forall \mathcal{B}(q).C\}$

$\forall_3$-rule: if $\forall \mathcal{B}.C \in \mathcal{L}(x)$, x is not indirectly blocked, $\varepsilon \in L(\mathcal{B})$, and $C \notin \mathcal{L}(x)$
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{C\}$

---

choose-rule: if $(\leqslant n S.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and
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 $(\geqslant n S.C) \in \mathcal{L}(x)$, x is not blocked, and
there are no $y_1, \ldots, y_n \in S^{\mathbf{F}}(x, C)$
with $y_i \neq y_j$ for each $1 \leq i < j \leq n$
then create n new successors $y_1, \ldots, y_n$ of x 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 $(\leqslant n S.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and
$\# S^{\mathbf{F}}(x, C) > n$, there are $y, z \in S^{\mathbf{F}}(x, C)$ with
not $y \neq z$ and y is not a root node nor 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. Set $u \neq z$ for all u with $u \neq y$.
4. remove y and the sub-tree below y from $\mathbf{F}$.

$\leqslant_r$-rule: if $(\leqslant n S.C) \in \mathcal{L}(x)$, $\# S^{\mathbf{F}}(x, C) > n$,
and there are two root nodes $y, z \in S^{\mathbf{F}}(x, C)$ with *not* $y \neq z$,
then 1. $\mathcal{L}(z) \longrightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ and
2. For all edges $\langle y, w\rangle$:
i. if the edge $\langle z, w\rangle$ does not exist, create it with $\mathcal{L}(\langle z, w\rangle) := \mathcal{L}(\langle y, w\rangle)$;
ii. else $\mathcal{L}(\langle z, w\rangle) \longrightarrow \mathcal{L}(\langle z, w\rangle) \cup \mathcal{L}(\langle y, w\rangle)$.
3. For all edges $\langle w, y\rangle$:
i. if the edge $\langle w, z\rangle$ does not exist, create it with $\mathcal{L}(\langle w, z\rangle) := \mathcal{L}(\langle w, y\rangle)$;
ii. else $\mathcal{L}(\langle w, z\rangle) \longrightarrow \mathcal{L}(\langle w, z\rangle) \cup \mathcal{L}(\langle w, y\rangle)$.
4. Set $u \neq z$ for all u with $u \neq y$.
5. Remove y and all incoming and outgoing from y from $\mathbf{F}$.

Fig. 1. The Expansion Rules for the $\mathcal{SRIQ}$ Tableau Algorithm.

of non-root nodes and an edge. Therefore, if a path p of non-root nodes 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 forest, and the only rules that removes nodes from the forest are the $\leqslant$- and the $\leqslant_r$-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 $\mathsf{fclos}(\mathcal{A}, \mathcal{R})$ contains a total of at most m $\exists R.C$, the out-degree of the forest is bounded by $mn_{\max}$. □

Lemma 7. *Let $\mathcal{A}$ be a $\mathcal{SRIQ}$-Abox and $\mathcal{R}$ an Rbox. The expansion rules can be applied to $\mathcal{A}$ and $\mathcal{R}$ such that they yield a complete and clash-free completion forest if and only if there is a tableau for $\mathcal{A}$ w.r.t. $\mathcal{R}$.*

For the if direction, we can unravel a complete and clash-free completion forest $\mathbf{F}$ in a standard way into a tableau T, where the same technique as for $\mathcal{SHIQ}$ is used to make sure that (P9) is satisfied even if two "sibling" nodes are blocked by the same node. It is easily seen that the $\forall_i$ expansion rules make sure that the resulting structure indeed satisfies the new tableau condition (P4a), (P4b), and (P6).

For the only-if direction, we take a tableau $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{R}$ and use it to steer the application of the non-deterministic rules, i.e., the $\sqcup$-, the X- and the $\leqslant$ -rule. To do this, while building the completion forest, we define a mapping π from the nodes of the completion forest into the tableau which satisfies the following three conditions:

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

We start with π mapping the root node to some tableau element s_0 with C_0 in its label, and prove that, if an expansion rule is applicable to $\mathbf{F}$, then this rule can be applied in such a way that $(*)$ is preserved. As a consequence of this claim, (P1), (P8), and Lemma 6, we thus end with a complete and clash-free completion forest.

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

Theorem 2. *The tableau algorithm decides satisfiability and subsumption of $\mathcal{SRIQ}$-concepts with respect to Aboxes, Rboxes, and Tboxes.*

Bibliography

[1] Areces, C., Blackburn, P., Hernandez, B., and Marx, M. Handling Boolean Aboxes. In *Proc. of the 2003 Description Logic Workshop (DL 2003)* (2003), CEUR (`http://ceur-ws.org/`).

[2] Baader, F. 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)* (Sydney, 1991).

[3] Baader, F., Bürckert, H.-J., Nebel, B., Nutt, W., and Smolka, G. On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations. *Journal of Logic, Language and Information 2* (1993), 1–18.

[4] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., Eds. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[5] Baader, F., Lutz, C., Milicic, M., Sattler, U., and Wolter, F. Integrating Description Logics and Action Formalisms: First Results. In *Proc. of the 20th National Conference on Artificial Intelligence (AAAI-05)* (2005), A. Press, Ed.

[6] Hopcroft, J. E., and Ullman, J. D. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley Publ. Co., Reading, Massachussetts, 1997.

[7] Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *J. of Web Semantics 1*, 1 (2003), 7–26.

[8] Horrocks, I., and Sattler, U. Optimised Reasoning for $\mathcal{SHIQ}$. In *Proc. of the 15th European Conf. on Artificial Intelligence (ECAI 2002)* (2002).

[9] Horrocks, I., and Sattler, U. Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. *Artificial Intelligence 160* (2004), 79–104.

[10] Horrocks, I., and Sattler, U. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. In *Proc. of 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)* (2005), Morgan Kaufmann, Los Altos.

[11] Horrocks, I., Sattler, U., and Tobies, S. Practical Reasoning for Expressive Description Logics. In *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)* (1999), H. Ganzinger, D. McAllester, and A. Voronkov, Eds., vol. 1705 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 161–180.

[12] Horrocks, I., Sattler, U., and Tobies, S. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In *Proc. of the 17th Conf. on Automated Deduction (CADE-17)* (Germany, 2000), D. MacAllester, Ed., vol. 1831 of *Lecture Notes in Computer Science*, Springer-Verlag.

[13] SCHILD, K. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)* (Sydney, 1991), pp. 466–471.

[14] WOLSTENCROFT, K., BRASS, A., HORROCKS, I., LORD, P., SATTLER, U., TURI, D., AND STEVENS, R. A Little Semantic Web Goes a Long Way in Biology. In *Proc. of the 4th International Semantic Web Conference* (2005), LNCS, SV. To appear.