The Even More Irresistible $\mathcal{SROIQ}$

Ian Horrocks and **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

We describe an extension of the description logic underlying OWL-DL, $\mathcal{SHOIN}$, with a number of expressive means that we believe will make it more useful in practice. Roughly speaking, we extend $\mathcal{SHOIN}$ with all expressive means that were suggested to us by ontology developers as useful additions to OWL-DL, and which, additionally, do not affect its decidability and practicability. We consider complex role inclusion axioms of the form $R \circ S \dot{\sqsubseteq} R$ or $S \circ R \dot{\sqsubseteq} R$ to express propagation of one property along another one, which have proven useful in medical terminologies. Furthermore, we extend $\mathcal{SHOIN}$ with reflexive, antisymmetric, and irreflexive roles, disjoint roles, a universal role, and constructs $\exists R.\mathsf{Self}$, allowing, for instance, the definition of concepts such as a "narcist". Finally, we consider negated role assertions in Aboxes and qualified number restrictions. The resulting logic is called $\mathcal{SROIQ}$.

We present a rather elegant tableau-based reasoning algorithm: it combines the use of automata to keep track of universal value restrictions with the techniques developed for $\mathcal{SHOIQ}$. The logic $\mathcal{SROIQ}$ has been adopted as the logical basis for the next iteration of OWL, OWL 1.1.

Introduction

We describe an extension, called $\mathcal{SROIQ}$, of the description logics (DLs) $\mathcal{SHOIN}$ (Horrocks, Sattler, & Tobies, 1999a) underlying OWL-DL (Horrocks, Patel-Schneider, & van Harmelen, 2003)[1] and $\mathcal{RIQ}$ (Horrocks & Sattler, 2004). $\mathcal{SHOIN}$ provides most expressive means that one could reasonably expect from the description-logical basis of an ontology language, and was designed to constitute a good compromise between expressive power and computational complexity/practicability of reasoning. It lacks, however, e.g. qualified number restrictions which are present in the DL considered here since they are required in various applications (Wolstencroft *et al.*, 2005) and do not pose problems (Horrocks & Sattler, 2005). That is, we extend $\mathcal{SHOIQ}$—which is $\mathcal{SHOIN}$ with qualified number restrictions—and extend the work begun in Horrocks, Kutz, & Sattler (2005).

Since OWL-DL is becoming more widely used, it turns out that it lacks a number of expressive means which—when considered carefully—can be added without causing too much difficulties for automated reasoning. We extend $\mathcal{SHOIQ}$ with these expressive means and, although they are not completely independent in that some of them can be expressed using others, first present them together with some examples. Recall that, in $\mathcal{SHOIQ}$, we can already state that a role is transitive or the subrole or the inverse of another one (and therefore also that it is symmetric).

In addition, $\mathcal{SROIQ}$ allows for the following:

1. *disjoint roles*. Most DLs can be said to be "unbalanced" 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. E.g., the roles `sister` and `mother` should be declared as being disjoint.
2. *reflexive, irreflexive, and antisymmetric roles*. These features are of minor interest when considering only TBoxes not using nominals, yet they add some useful constraints if we also refer to individuals, either by using nominals or ABoxes, especially in the presence of number restrictions. E.g., the roles `knows`, `hasSibling`, and `properPartOf`, should be declared as, respectively, reflexive, irreflexive, and antisymmetric.
3. *negated role assertions*. Most Abox formalisms only allow for positive role assertions (with few exceptions (Areces *et al.*, 2003; Baader *et al.*, 2005)), whereas $\mathcal{SROIQ}$ also allows for statements like $(\mathtt{John}, \mathtt{Mary}) : \neg\mathtt{likes}$. In the presence of complex role inclusions, negated role assertions can be quite useful and, like disjoint roles, they overcome a certain asymmetry in expressivity.
4. $\mathcal{SROIQ}$ provides complex role inclusion axioms of the form $R \circ S \dot{\sqsubseteq} R$ and $S \circ R \dot{\sqsubseteq} R$ that were first introduced in $\mathcal{RIQ}$. For example, w.r.t. the axiom $\mathtt{owns} \circ \mathtt{hasPart} \dot{\sqsubseteq} \mathtt{owns}$, and the fact that each car contains an engine $\mathtt{Car} \dot{\sqsubseteq} \exists\mathtt{hasPart}.\mathtt{Engine}$, an owner of a car is also an owner of an engine, i.e., the following subsumption holds: $\exists\mathtt{owns}.\mathtt{Car} \sqsubseteq \exists\mathtt{owns}.\mathtt{Engine}$.

[1] OWL also includes *datatypes*, a simple form of *concrete domain* (Baader & Hanschke, 1991). These can, however, be treated exactly as in $\mathcal{SHOQ}(\mathbf{D})$/$\mathcal{SHOQ}(\mathbf{D}_n)$ (Horrocks & Sattler, 2001; Pan & Horrocks, 2003), so we will not complicate our presentation by considering them here.

5. $\mathcal{SROIQ}$ provides the *universal role* U. Together with nominals (which are also provided by $\mathcal{SHOIQ}$), this role is a prominent feature of hybrid logics (Blackburn & Seligman, 1995). Nominals can be viewed as a powerful generalisation of *ABox individuals* (Schaerf, 1994; Horrocks & Sattler, 2001), and they occur naturally in ontologies, e.g., when describing a class such as EUCountries by enumerating its members.
6. Finally, $\mathcal{SROIQ}$ 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" as $\exists \mathtt{likes}.\mathsf{Self}$.

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

$\mathcal{SROIQ}$ is designed to be of similar practicability as $\mathcal{SHOIQ}$. The tableau algorithm for $\mathcal{SROIQ}$ presented here is essentially a combination of the algorithms for $\mathcal{RIQ}$ and $\mathcal{SHOIQ}$. In particular, it employs the same technique using finite automata as in Horrocks & Sattler (2004) to handle role inclusions $R \circ S \dot{\sqsubseteq} R$ and $S \circ R \dot{\sqsubseteq} R$. Even though the additional expressive means require certain adjustments, these adjustments do not add new sources of non-determinism and, subject to empirical verification, are believed to be "harmless" in the sense of not significantly degrading typical performance as compared with the $\mathcal{SHOIQ}$ algorithm. Moreover, the algorithm for $\mathcal{SROIQ}$ has, similar to the one for $\mathcal{SHOIQ}$, excellent "pay as you go" characteristics. For instance, in case only expressive means of $\mathcal{SHIQ}$ are used, the new algorithm will behave just like the algorithm for $\mathcal{SHIQ}$.

We believe that the combination of properties described above makes $\mathcal{SROIQ}$ a very useful basis for future extensions of OWL DL.

The Logic $\mathcal{SROIQ}$

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

Roles, Role Hierarchies, and Role Assertions

Definition 1 *Let $\mathbf{C}$ be a set of **concept names** including a subset $\mathbf{N}$ of **nominals**, $\mathbf{R}$ a set of **role names** including the universal role U, and $\mathbf{I} = \{a, b, c \dots\}$ 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 the universal role U the universal relation $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$, with each concept name C a subset $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$, where $C^\mathcal{I}$ is a singleton set if $C \in \mathbf{N}$, 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*

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

Obviously, $(U^-)^\mathcal{I} = (U)^\mathcal{I}$. Note that, unlike in the cases of $\mathcal{SHIQ}$ or $\mathcal{SHOIQ}$, we did not introduce *transitive role names*. This is because, 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 finite strings of roles it is convenient to extend both $\cdot^\mathcal{I}$ and $\mathsf{Inv}(\cdot)$ to such strings: if $w = R_1 \dots R_n$ is a string of roles R_i $(1 \leq i \leq n)$, we set $\mathsf{Inv}(w) = \mathsf{Inv}(R_n) \dots \mathsf{Inv}(R_1)$ and $w^\mathcal{I} = R_1^\mathcal{I} \circ \dots \circ R_n^\mathcal{I}$, 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*. The second component is a set $\mathcal{R}_a$ of *role assertions* stating, for instance, that a role R must be interpreted as an irreflexive relation.

We start with the definition of a (regular) role hierarchy whose definition involves a certain ordering on roles, called *regular*. A strict partial order $\prec$ on a set A is an irreflexive and transitive relation on A. A strict partial order $\prec$ on the set of roles $R \cup \{R^- \mid R \in \mathbf{R}\}$ is called a **regular order** if $\prec$ satisfies, additionally, $S \prec R \iff S^- \prec R$, for all roles R and S. Note, in particular, that the irreflexivity of $\prec$ ensures that neither $S^- \prec S$ nor $S \prec S^-$ hold.

Definition 2 ((Regular) Role Inclusion Axioms) *Let $\prec$ be a regular 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 not including the universal role U, and $R \neq U$ is a role name. A **role hierarchy** $\mathcal{R}_h$ is a finite set of RIAs. An interpretation $\mathcal{I}$ **satisfies** a role inclusion axiom $w \dot{\sqsubseteq} R$, written $\mathcal{I} \models w \dot{\sqsubseteq} R$, if $w^\mathcal{I} \subseteq R^\mathcal{I}$. 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, and*

1. *$w = RR$, or*
2. *$w = R^-$, or*
3. *$w = S_1 \dots S_n$ and $S_i \prec R$, for all $1 \leq i \leq n$, or*
4. *$w = RS_1 \dots S_n$ and $S_i \prec R$, for all $1 \leq i \leq n$, or*
5. *$w = S_1 \dots S_n R$ and $S_i \prec R$, for all $1 \leq i \leq n$.*

*Finally, a role hierarchy $\mathcal{R}_h$ is **regular** if there exists a regular order $\prec$ 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

$$\{RS \dot{\sqsubseteq} S, \quad RT \dot{\sqsubseteq} R, \quad VT \dot{\sqsubseteq} T, \quad VS \dot{\sqsubseteq} V\}$$

is not regular because it would require $\prec$ to satisfy $S \prec V \prec T \prec R \prec S$, which would imply $S \prec S$, thus contradicting the irreflexivity of $\prec$. Such cyclic dependencies are known to lead to undecidability (Horrocks & Sattler, 2004).

Also, note that RIAs of the form $RR^- \dot{\sqsubseteq} R$, which would imply (a weak form of) reflexivity of R, are not regular according to the definition of regular orderings. However, the same condition on R can be imposed by using the GCI $\exists R.\top \dot{\sqsubseteq} \exists R.\mathsf{Self}$; see below.

From the definition of the semantics of inverse roles, it follows immediately that $\langle x, y\rangle \in w^\mathcal{I}$ 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 only 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** (**super-role**) of a role S if $R \underline{\circledast} S$ ($S \underline{\circledast} R$). Two roles R and S are **equivalent** ($R \equiv S$) if $R \underline{\circledast} S$ and $S \underline{\circledast} R$.

Note that, due to restriction (3) 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 $\{\langle x, x\rangle \mid x \in \Delta^{\mathcal{I}}\}$. Note that, since the interpretation of the universal role U is fixed in any given model (as the universal relation on $\Delta^I \times \Delta^I$ which is, by definition, reflexive, symmetric, and transitive), we disallow the universal role to appear in role assertions.

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

$$
\begin{array}{lll}
\mathcal{I} \models \mathsf{Sym}(R) & \textit{if} & \langle x,y\rangle \in R^{\mathcal{I}} \textit{ implies } \langle y,x\rangle \in R^{I};\\
\mathcal{I} \models \mathsf{Asy}(R) & \textit{if} & \langle x,y\rangle \in R^{\mathcal{I}} \textit{ implies } \langle y,x\rangle \notin R^{\mathcal{I}}\\
\mathcal{I} \models \mathsf{Tra}(R) & \textit{if} & \langle x,y\rangle \in R^{\mathcal{I}} \textit{ and } \langle y,z\rangle \in R^{\mathcal{I}}\\
 & & \textit{imply } \langle x,z\rangle \in R^{I};\\
\mathcal{I} \models \mathsf{Ref}(R) & \textit{if} & Diag^{\mathcal{I}} \subseteq R^{\mathcal{I}};\\
\mathcal{I} \models \mathsf{Irr}(R) & \textit{if} & R^{\mathcal{I}} \cap Diag^{\mathcal{I}} = \emptyset;\\
\mathcal{I} \models \mathsf{Dis}(R,S) & \textit{if} & R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset.
\end{array}
$$

Adding symmetric and transitive role assertions is a trivial move since both of these expressive means can be replaced with complex role inclusion axioms as follows: $\mathsf{Sym}(R)$ is equivalent to $R^- \dot\sqsubseteq R$ and $\mathsf{Tra}(R)$ is equivalent to $RR \dot\sqsubseteq R$. 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/or symmetric roles will be handled by the RIAs alone. In particular, notice that regularity of a role hierarchy is preserved when replacing such role assertions with the corresponding RIAs.

The situation is different, however, for the other Rbox assertions. None of reflexivity, irreflexivity, antisymmetry or 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 $\exists R.\mathsf{Self}$.

Note that the version of antisymmetry introduced above is *strict* in the sense that it also implies irreflexivity as opposed to the more widely used notion of antisymmetry which allows for reflexive points. For instance, in *mereology*, the relation `PartOf` is usually assumed to be 'reflexive antisymmetric' (i.e., reflexivity, plus xRy and yRx implies $x = y$), while the relation `properPartOf` is assumed to be 'irreflexive antisymmetric' (defined just as antisymmetry above) (Simons, 1987; Casati & Varzi, 1999). The more general version of antisymmetry is more difficult to handle algorithmically, and we leave this to future work.

[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.

In $\mathcal{SHIQ}$ (and $\mathcal{SHOIQ}$), the application of qualified number restrictions has to be restricted to certain roles, called *simple roles*, to preserve decidability (Horrocks, Sattler, & Tobies, 1999a). In the context of $\mathcal{SROIQ}$, 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 $\mathsf{Irr}(R)$, $\mathsf{Asy}(R)$, 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* ***$\mathcal{SROIQ}$-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 $\mathcal{R}$.*

Concepts and Inference Problems for $\mathcal{SROIQ}$

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

- *every concept name (including nominals) 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 $C \dot\sqsubseteq D$ for two $\mathcal{SROIQ}$-concepts C and D. A* ***Tbox*** *$\mathcal{T}$ is a finite set of GCIs.*

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

It is part of future work to determine which of the restrictions to simple roles in role assertions $\mathsf{Dis}(R, S)$, $\mathsf{Asy}(R)$, and $\mathsf{Irr}(R)$, as well as the concept expression $\exists S.\mathsf{Self}$, are strictly necessary in order to preserve decidability or practicability. These restrictions, however, allow a rather smooth integration of the new constructs into existing algorithms.

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, and concept names, roles, and nominals are interpreted as in Definition 1:*

$$\begin{array}{l}
\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \quad \bot^{\mathcal{I}} = \emptyset, \quad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, \quad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \\
(\exists R.\mathsf{Self})^{\mathcal{I}} = \{x \mid \langle x,x\rangle \in R^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\} \\
(\geqslant nR.C)^{\mathcal{I}} = \{x \mid \sharp\{y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geqslant n\} \\
(\leqslant nR.C)^{\mathcal{I}} = \{x \mid \sharp\{y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leqslant n\}
\end{array}$$

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

*$\mathcal{I}$ **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*

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

*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 $\mathcal{I} \models \mathcal{A}$.*

The above inference problems can be defined w.r.t. a 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 $\mathcal{T}$.

Reduction of Inference Problems

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 $\{a : C\}$ (a a 'new' individual name) 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 Baader (1991); Schild (1991); Baader *et al.* (1993), the *internalisation* of GCIs is introduced, a technique that realises exactly this reduction. For $\mathcal{SROIQ}$, this technique only needs to be slightly modified. We will show in a series of steps that, in $\mathcal{SROIQ}$, satisfiability of a concept C with respect to a triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{T}\rangle$ of, respectively, a $\mathcal{SROIQ}$ Abox, Rbox, and Tbox, can be reduced to concept satisfiability of a concept C' with respect to an Rbox $\mathcal{R}'$, where the Rbox $\mathcal{R}'$ only contains role assertions of the form $\mathsf{Dis}(R,S)$, $\mathsf{Ref}(R)$, or $\mathsf{Asy}(R)$, and the universal role U does not appear in C'.

While nominals can be used to 'internalise' the Abox, in order to eliminate the universal role, we use a 'simulated' universal role U', i.e., a reflexive, symmetric, and transitive super-role of all roles and their inverses appearing in $\langle \mathcal{A}, \mathcal{R}, \mathcal{T}\rangle$, and which, additionally, connects all nominals appearing in the input.

Thus, let C and $\langle \mathcal{A}, \mathcal{R}, \mathcal{T}\rangle$ be a $\mathcal{SROIQ}$ concept and Abox, Rbox, and Tbox, respectively. In a first step, we replace the Abox $\mathcal{A}$ with an Abox $\mathcal{A}'$ such that $\mathcal{A}'$ only contains individual assertions of the form $a : C$. To this purpose, we associate with every individual $a \in \mathbf{I}$ appearing in $\mathcal{A}$ a new nominal o_a not appearing in $\mathcal{T}$ or C. Next, $\mathcal{A}'$ is the result of replacing every individual assertion in $\mathcal{A}$ of the form $(a,b) : R$ with $a : \exists R.o_b$, every $(a,b) : \neg R$ with $a : \forall R.\neg o_b$, and every $a \neq b$ with $a : \neg o_b$. Now, given C and $\mathcal{A}'$, define C' as follows:

$$C' := C \sqcap \bigsqcap_{a:D \in \mathcal{A}'} \exists U.(o_a \sqcap D),$$

where U is the universal role. It should be clear that $\mathbf{C}$ is satisfiable with respect to $\langle \mathcal{A}, \mathcal{R}, \mathcal{T}\rangle$ if and only if C' is satisfiable with respect to $\langle \mathcal{R}, \mathcal{T}\rangle$.

Lemma 7 (Abox Elimination) *$\mathcal{SROIQ}$ concept satisfiability with respect to Aboxes, Rboxes, and Tboxes is polynomially reducible to $\mathcal{SROIQ}$ concept satisfiability with respect to Rboxes and Tboxes only.*

Hence, in the following we will assume that Aboxes have been eliminated. Next, although we have the 'real' universal role U present in the language, the following lemma shows how general concept inclusion axioms can be *internalised* while at the same time eliminating occurrences of the universal role U, using a simulated "universal" role U', that is, a transitive super-role of all roles (except U) occurring in $\mathcal{T}$ or $\mathcal{R}$ and their respective inverses. Recall that the universal role U is not allowed to appear in Rboxes.

Lemma 8 (Tbox and Universal Role Elimination) *Let C, D be concepts, $\mathcal{T}$ a Tbox, and $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ an Rbox. Let $U' \neq U$ be a role that does not occur in C, D, $\mathcal{T}$, or $\mathcal{R}$, and, for X a Tbox or a concept, let X' result from X by replacing every occurrence of U with U'. We define*

$$C_{\mathcal{T}'} := \forall U'.\big(\bigsqcap_{C'_i \dot\sqsubseteq D'_i \in \mathcal{T}'} \neg C'_i \sqcup D'_i\big) \sqcap \big(\bigsqcap_{\mathbf{N} \ni o \in \mathcal{T} \cup C \cup D} \exists U'.o\big),$$

$$\mathcal{R}_h^{U'} := \mathcal{R}_h \cup \{R \dot\sqsubseteq U' \mid R \text{ occurs in } C', D', \mathcal{T}', \text{ or } \mathcal{R}\},$$

$$\mathcal{R}_a^{U'} := \mathcal{R}_a \cup \{\mathsf{Tra}(U'), \mathsf{Sym}(U'), \mathsf{Ref}(U')\}, \text{ and}$$

$$\mathcal{R}_{U'} := \mathcal{R}_h^{U'} \cup \mathcal{R}_a^{U'}. \text{ Then}$$

- *C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{R}$ iff $C' \sqcap C_{\mathcal{T}'}$ is satisfiable w.r.t. $\mathcal{R}_{U'}$.*
- *D subsumes C with respect to $\mathcal{T}$ and $\mathcal{R}$ iff $C' \sqcap \neg D' \sqcap C_{\mathcal{T}'}$ is unsatisfiable w.r.t. $\mathcal{R}_{U'}$.*

The proof of Lemma 8 is similar to the ones that can be found in Schild (1991) and Baader (1991). Most importantly, it must be shown that (a): if a $\mathcal{SROIQ}$-concept C is satisfiable with respect to a Tbox $\mathcal{T}$ and an Rbox $\mathcal{R}$, then $C, \mathcal{T}, \mathcal{R}$ have a **nominal connected** model, i.e., a model which is a union of connected components, where each such component contains a nominal, and where any two elements of a connected component are connected by a role path over those roles occurring in C, $\mathcal{T}$ or $\mathcal{R}$, 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' and $C_{\mathcal{T}'}$, which guarantees that all nominals are connected by U' links.

Now, note also that, instead of having a role assertion $\mathsf{Irr}(R) \in \mathcal{R}_a$, we can add, equivalently, the GCI $\top \dot\sqsubseteq \neg\exists R.\mathsf{Self}$ to $\mathcal{T}$, which can in turn be internalised. Likewise, instead of asserting $\mathsf{Ref}(R)$, we can, equivalently, add the GCI $\top \dot\sqsubseteq \exists R.\mathsf{Self}$ to $\mathcal{T}$. However, in the case of $\mathsf{Ref}(R)$ this replacement is only admissible for *simple* roles R and thus not possible (syntactically) in general.

Thus, using these equivalences (including the replacement of Rbox assertions of the form $\mathsf{Sym}(R)$ and $\mathsf{Tra}(R)$) and Lemmas 7 and 8, we arrive at the following theorem:

Theorem 9 (Reduction)

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

With Theorem 9, all standard inference problems for $\mathcal{SROIQ}$-concepts and Aboxes can be reduced to the problem of determining the consistency of a $\mathcal{SROIQ}$-concept w.r.t. to an Rbox (both not containing the universal role), where we can assume w.l.o.g. that all role assertions in the Rbox are of the form $\mathsf{Ref}(R)$, $\mathsf{Asy}(R)$, or $\mathsf{Dis}(R, S)$—we call such an Rbox **reduced**.

$\mathcal{SROIQ}$ is Decidable

In this section, we show that $\mathcal{SROIQ}$ is decidable. We present a tableau-based algorithm that decides the consistency of a $\mathcal{SROIQ}$ concept w.r.t. a reduced Rbox, and therefore also all standard inference problems as discussed above, see Theorem 9. Therefore, in the following, by Rbox we always mean *reduced* Rbox.

The algorithm tries to construct, for a $\mathcal{SROIQ}$-concept C and an Rbox $\mathcal{R}$, a *tableau* for C and $\mathcal{R}$, that is, an abstraction of a model of C and $\mathcal{R}$.

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) $\mathcal{B}_S$ is defined. The construction of these automata is identical to the one presented in Horrocks & Sattler (2004), and we therefore refer to this paper for detailed definitions and proofs of the automata related results below.

The following proposition states that $\mathcal{B}_S$ indeed captures all implications between (paths of) roles and S that are consequences of the role hierarchy $\mathcal{R}_h$, where $L(\mathcal{B}_S)$ denotes the language (a set of strings of roles) accepted by $\mathcal{B}_S$.

Proposition 10 *$\mathcal{I}$ 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 $\langle x, y\rangle \in S^{\mathcal{I}}$.*

Unfortunately, as shown in Horrocks & Sattler (2004), the size of $\mathcal{B}_R$ can be exponential in the size of $\mathcal{R}$. Horrocks & Sattler (2004) consider certain further syntactic restrictions of role hierarchies (there called *simple* role hierarchies) that avoid this exponential blow-up. We conjecture that, without some such further restriction, this blow-up is unavoidable. The following technical Lemma from Horrocks & Sattler (2004) will be needed later on.

Lemma 11

1. *$S \in L(\mathcal{B}_S)$ and, if $w \dot\sqsubseteq S \in \mathcal{R}$, then $w \in L(\mathcal{B}_S)$.*
2. *If S is a simple role, then $L(\mathcal{B}_S) = \{R \mid R \,\underline{\circledast}\, S\}$.*
3. *$L(\mathcal{B}_{\mathsf{Inv}(S)}) = \{\mathsf{Inv}(w) \mid w \in L(\mathcal{B}_S)\}$.*

A Tableau for $\mathcal{SROIQ}$

In the following, if not stated otherwise, C, D (possibly with subscripts) denote $\mathcal{SROIQ}$-concepts (not using the universal role), R, S (possibly with subscripts) roles, $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ an Rbox, and $\mathbf{R}_C$ the set of roles occurring in C and $\mathcal{R}$ together with their inverses. Furthermore, as noted in Theorem 9, we can (and will from now on) assume w.l.o.g. that all role assertions appearing in $\mathcal{R}_a$ are of the form $\mathsf{Dis}(R, S)$, $\mathsf{Asy}(R)$, or $\mathsf{Ref}(R)$.

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 all paths while the NFA gets "updated" with the path taken so far. For this "update", we use the following definition.

Definition 12 *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 **negation normal form** (NNF), that is, negation occurs only in front of concept names or in front of $\exists R.\mathsf{Self}$. Any $\mathcal{SROIQ}$-concept can easily be transformed into an equivalent one in NNF by pushing negations inwards using a combination of De Morgan's laws and equivalences such as $\neg(\exists R.C) \equiv (\forall R.\neg C)$, $\neg({\leqslant} nR.C) \equiv ({\geqslant}(n+1)R.C)$, etc. 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 mentioned above 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.

Definition 13 (Tableau) *$T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ is a* tableau *for C_0 w.r.t. $\mathcal{R}$ if*

- *$\mathbf{S}$ is a non-empty set;*
- *$\mathcal{L} : \mathbf{S} \to 2^{\mathsf{fclos}(C_0,\mathcal{R})}$ maps each element in $\mathbf{S}$ to a set of concepts;*
- *$\mathcal{E} : \mathbf{R}_{C_0} \to 2^{\mathbf{S}\times\mathbf{S}}$ maps each role to a set of pairs of elements in $\mathbf{S}$;*
- *$C_0 \in \mathcal{L}(s)$ for some $s \in \mathbf{S}$.*

Furthermore, for all $s, t \in \mathbf{S}$, $C, C_1, C_2 \in \mathsf{fclos}(C_0, \mathcal{R})$, $o \in \mathbf{N} \cap \mathsf{fclos}(C_0, \mathcal{R})$, $R, S \in \mathbf{R}_{C_0}$, and

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

the tableau T satisfies:

(P1) $C \in \mathcal{L}(s) \implies \neg C \notin \mathcal{L}(s)$, *($C$ atomic or $\exists R.\mathsf{Self}$);*
(P2) $\top \in \mathcal{L}(s)$, *and* $\bot \notin \mathcal{L}(s)$;
(P3) $\exists R.\mathsf{Self} \in \mathcal{L}(s) \implies \langle s, s\rangle \in \mathcal{E}(R)$;
(P4) $\neg\exists R.\mathsf{Self} \in \mathcal{L}(s) \implies \langle s, s\rangle \notin \mathcal{E}(R)$;
(P5) $C_1 \sqcap C_2 \in \mathcal{L}(s) \implies C_1, C_2 \in \mathcal{L}(s)$;
(P6) $C_1 \sqcup C_2 \in \mathcal{L}(s) \implies C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$;
(P7) $\forall\mathcal{B}(p).C \in \mathcal{L}(s)$, $\langle s, t\rangle \in \mathcal{E}(S)$, *and* $p \xrightarrow{S} q \in \mathcal{B}(p) \implies \forall\mathcal{B}(q).C \in \mathcal{L}(t)$;
(P8) $\forall\mathcal{B}.C \in \mathcal{L}(s)$ *and* $\varepsilon \in L(\mathcal{B}) \implies C \in \mathcal{L}(s)$;
(P9) $\forall S.C \in \mathcal{L}(s) \implies \forall\mathcal{B}_S.C \in \mathcal{L}(s)$;
(P10) $\exists S.C \in \mathcal{L}(s) \implies$ *there is some* $r \in \mathbf{S}$ *with* $\langle s, r\rangle \in \mathcal{E}(S)$ *and* $C \in \mathcal{L}(r)$;
(P11) $\langle s, t\rangle \in \mathcal{E}(R) \iff \langle t, s\rangle \in \mathcal{E}(\mathsf{Inv}(R))$;
(P12) $\langle s, t\rangle \in \mathcal{E}(R)$ *and* $R \,\underline{\circledast}\, S \implies \langle s, t\rangle \in \mathcal{E}(S)$;
(P13) $(\leqslant nS.C) \in \mathcal{L}(s) \implies \sharp S^T(s, C) \leqslant n$;
(P14) $(\geqslant nS.C) \in \mathcal{L}(s) \implies \sharp S^T(s, C) \geqslant n$;
(P15) $(\leqslant nS.C) \in \mathcal{L}(s)$ *and* $\langle s, t\rangle \in \mathcal{E}(S) \implies C \in \mathcal{L}(t)$ *or* $\dot\neg C \in \mathcal{L}(t)$;
(P16) $\mathsf{Dis}(R, S) \in \mathcal{R}_a \implies \mathcal{E}(R) \cap \mathcal{E}(S) = \emptyset$;
(P17) $\mathsf{Ref}(R) \in \mathcal{R}_a \implies \langle s, s\rangle \in \mathcal{E}(R)$;
(P18) $\mathsf{Asy}(R) \in \mathcal{R}_a \implies \langle s, t\rangle \in \mathcal{E}(R)$ *implies* $\langle t, s\rangle \notin \mathcal{E}(R)$;
(P19) $o \in \mathcal{L}(r)$ *for some* $r \in \mathbf{S}$;
(P20) $o \in \mathcal{L}(s) \cap \mathcal{L}(t) \implies s = t$.

Theorem 14 (Tableau) *A $\mathcal{SROIQ}$-concept C_0 is satisfiable w.r.t. a reduced Rbox $\mathcal{R}$ iff there exists a tableau for C_0 w.r.t. $\mathcal{R}$.*

Proof: For the *if* direction, let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ be a tableau for C_0 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 C_0 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 $\mathsf{fclos}(C_0, \mathcal{R})$, where (P19) and (P20) guarantee that nominals are indeed interpreted as singleton sets, and, for roles names $R \in \mathbf{R}_{C_0}$, we set

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

The semantics of complex concepts is given through the definition of the $\mathcal{SROIQ}$-semantics. Due to Lemma 11.3 and (P11), the semantics of inverse roles can either be given directly as for role names, or by setting $(R^-)^{\mathcal{I}} := \{\langle y, x\rangle \mid \langle x, y\rangle \in R^{\mathcal{I}}\}$. Moreover, we have, by definition of $\mathcal{I}$, Lemma 11.2, (P11), and (P12) that, for T a simple role, $T^{\mathcal{I}} = \mathcal{E}(T)$.

We have to show that $\mathcal{I}$ is a model of $\mathcal{R}$ and C_0. We begin by showing that $\mathcal{I} \models \mathcal{R}$. Since $\mathcal{R}$ is reduced, we only have to deal with role assertions of the form $\mathsf{Dis}(R, S)$, $\mathsf{Ref}(R)$, and $\mathsf{Asy}(R)$.

Consider an assertion $\mathsf{Dis}(R, S) \in \mathcal{R}$. By definition of $\mathcal{SROIQ}$-Rboxes, both R and S are simple roles, and thus $R^{\mathcal{I}} = \mathcal{E}(R)$ and $S^{\mathcal{I}} = \mathcal{E}(S)$. Moreover, (P16) implies $\mathcal{E}(R)\cap\mathcal{E}(S) = \emptyset$, and thus $R^{\mathcal{I}}\cap S^{\mathcal{I}} = \emptyset$. Next, if $\mathsf{Ref}(R) \in \mathcal{R}_a$, (P17) and $R \in L(\mathcal{B}_R)$ (Lemma 11.1) imply $Diag^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. Finally, if $\mathsf{Asy}(R) \in \mathcal{R}_a$ then $R^{\mathcal{I}} = \mathcal{E}(R)$ since R is simple, and so $\langle s, t\rangle \in R^{\mathcal{I}}$ implies $\langle s, t\rangle \in \mathcal{E}(R)$ and so $\langle t, s\rangle \notin \mathcal{E}(R)$ by (P18), whence $\langle t, s\rangle \notin R^{\mathcal{I}}$. Thus $\mathcal{I}$ satisfies each role assertion in $\mathcal{R}_a$.

Next, we have to show that $\mathcal{I} \models \mathcal{R}_h$. Due to Proposition 10, 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 $\langle x, y\rangle \in S^{\mathcal{I}}$. The proof of this is identical to the case of $\mathcal{RIQ}$ and can be found in Horrocks & Sattler (2004).

Secondly, we prove that $\mathcal{I}$ is a model of C_0. We show that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$ for each $s \in \mathbf{S}$ and each $C \in \mathsf{fclos}(\mathcal{A}, \mathcal{R})$. 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 $C = (\neg)\exists R.\mathsf{Self}$ (for the other cases, see Horrocks, Sattler, & Tobies (2000); Horrocks & Sattler (2002)):

- If $(\leqslant nS.E) \in \mathcal{L}(s)$, then (P13) implies that $\# S^T(s, E) \leq n$. Moreover, since S is simple, Lemma 11.2 implies that $L(\mathcal{B}_S) = \{S' \mid S' \,\underline{\circledast}\, S\}$, and (P12) implies that $S^{\mathcal{I}} = \mathcal{E}(S)$. Hence (P15) 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 (P9) we have that $\forall\mathcal{B}_S.E \in \mathcal{L}(s)$. By definition of $S^{\mathcal{I}}$, there are $S_1 \ldots S_n \in L(\mathcal{B}_S)$ and s_i with $s = s_0$, $t = s_n$, and $\langle s_{i-1}, s_i\rangle \in \mathcal{E}(S_i)$. Applying (P7) n times, this yields $\forall\mathcal{B}_S(q).E \in \mathcal{L}(t)$ for q a final state of $\mathcal{B}_S$. Thus (P8) 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 (P3), $\langle s, s\rangle \in \mathcal{E}(R)$ 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 $s \in (\exists R.\mathsf{Self})^{\mathcal{I}}$.

- Let $\neg\exists R.\mathsf{Self} \in \mathcal{L}(s)$. Then, by (P4), $\langle s, s\rangle \notin \mathcal{E}(R)$. Since R is a simple role, $R^{\mathcal{I}} = \mathcal{E}(R)$. Hence $\langle s, s\rangle \notin R^{\mathcal{I}}$, and so $s \in (\neg\exists R.\mathsf{Self})^{\mathcal{I}}$.

For the converse, suppose $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a model of C_0 w.r.t. $\mathcal{R}$. We define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ for C_0 and $\mathcal{R}$ as follows:

$$\mathbf{S} := \Delta^{\mathcal{I}};$$

$$\mathcal{E}(R) := R^{\mathcal{I}};\text{ and}$$

$$\begin{aligned}\mathcal{L}(s) := &\{C \in \mathsf{fclos}(C_0, \mathcal{R}) \mid s \in C^{\mathcal{I}}\}\\ &\cup \{\forall\mathcal{B}_S.C \mid \forall S.C \in \mathsf{fclos}(C_0,\mathcal{R}) \text{ and } s \in (\forall S.C)^{\mathcal{I}}\}\\ &\cup \{\forall\mathcal{B}_R(q).C \in \mathsf{fclos}(C_0,\mathcal{R}) \mid S_1 \cdots S_n \in L(\mathcal{B}_R(q)) \Rightarrow\\ &\quad s \in (\forall S_1.\forall S_2.\cdots\forall S_n.C)^{\mathcal{I}}, \varepsilon \in L(\mathcal{B}_R(q)) \Rightarrow s \in C^{\mathcal{I}}\}\end{aligned}$$

We have to show that T satisfies (P1)–(P20), and restrict our attention to the only new cases.

For (P9), 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 (P7), let $\forall\mathcal{B}(p).C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(S)$. 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:

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

For (P8), $\varepsilon \in L(\mathcal{B}(p))$ implies $s \in C^{\mathcal{I}}$ by definition of T, and thus $C \in \mathcal{L}(s)$.

Finally, (P16)–(P20) follow immediately from the definition of the semantics.

□

The Tableau Algorithm

In this section, we present a terminating, sound, and complete tableau algorithm that decides consistency of $\mathcal{SROIQ}$-concepts not using the universal role w.r.t. reduced Rboxes, and thus, using Theorem 9, also concept satisfiability w.r.t. Rboxes, Tboxes and Aboxes.

We first define the underlying data structures and corresponding operations. For more detailed explanations concerning the intuitions underlying these definitions, consult Horrocks & Sattler (2005).

The algorithm generates a *completion graph*, a structure that, if complete and clash-free, can be unravelled to a (possibly infinite) tableau for the input concept and Rbox. Moreover, it is shown that the algorithm returns a complete and clash-free completion graph for C_0 and $\mathcal{R}$ if and only if there exists a tableau for C_0 and $\mathcal{R}$, and thus with Lemma 14, if and only if the concept C_0 is satisfiable w.r.t. $\mathcal{R}$.

As usual, in the presence of transitive roles, *blocking* is employed to ensure termination of the algorithm (Horrocks, Sattler, & Tobies, 2000).

Definition 15 (Completion Graph) *Let $\mathcal{R}$ be a reduced Rbox, let C_0 be a $\mathcal{SROIQ}$-concept in NNF not using the universal role, and let $\mathbf{N}$ be the set of nominals. A* ***completion graph*** *for C_0 with respect to $\mathcal{R}$ is a directed graph $\mathbf{G} = (V, E, \mathcal{L}, \neq)$ where each node $x \in V$ is labelled with a set*

$$\mathcal{L}(x) \subseteq \mathsf{fclos}(C_0,\mathcal{R}) \cup \mathbf{N} \cup \{(\leqslant mR.C) \mid (\leqslant nR.C) \in \mathsf{fclos}(C_0,\mathcal{R}) \text{ and } m \leq n\}$$

and each edge $\langle x, y\rangle \in E$ is labelled with a set of role names $\mathcal{L}(\langle x, y\rangle)$ containing (possibly inverse) roles occurring in C_0 or $\mathcal{R}$. Additionally, we keep track of inequalities between nodes of the graph with a symmetric binary relation $\neq$ between the nodes of $\mathbf{G}$.

In the following, we often use $R \in \mathcal{L}(\langle x, y\rangle)$ as an abbreviation for $\langle x, y\rangle \in E$ and $R \in \mathcal{L}(\langle x, y\rangle)$.

If $\langle x, y\rangle \in E$, then y is called a ***successor*** *of x and x is called a* ***predecessor*** *of y.* ***Ancestor*** *is the transitive closure of predecessor, and* ***descendant*** *is the transitive closure of successor. A node y is called an R-****successor*** *of a node x if, for some R' with $R' \underline{\circledast} R$, $R' \in \mathcal{L}(\langle x, y\rangle)$. A node y is called a* ***neighbour*** *(R-****neighbour****) of a node x if y is a successor (R-successor) of x or if x is a successor ($\mathsf{Inv}(R)$-successor) of y.*

For a role S and a node x in $\mathbf{G}$, we define the set of x's S-neighbours with C in their label, $S^{\mathbf{G}}(x, C)$, as follows:

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

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

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

If $o_1, \ldots, o_\ell$ are all the nominals occurring in C_0 then the tableau algorithm is initialised with the completion graph $\mathbf{G} = (\{r_0, r_1 \ldots, r_\ell\}, \emptyset, \mathcal{L}, \emptyset)$ with $\mathcal{L}(r_0) = \{C_0\}$ and $\mathcal{L}(r_i) = \{o_i\}$ for $1 \leq i \leq \ell$. $\mathbf{G}$ is then expanded by repeatedly applying the expansion rules given in Figure 1, stopping if a clash occurs.

Before describing the tableau algorithm in more detail, we define some terms and operations used in the (application of the) expansion rules:

Nominal Nodes and Blockable Nodes A node x is a **nominal node** if $\mathcal{L}(x)$ contains a nominal. A node that is not a nominal node is a **blockable** node. A nominal $o \in \mathbf{N}$ is said to be **new in G** if no node in $\mathbf{G}$ has o in its label.

Blocking A node x is **label blocked** if it has ancestors x', y and y' such that

1. x is a successor of x' and y is a successor of y',
2. y, x and all nodes on the path from y to x are blockable,
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 is **blocked** if either it is label blocked or it is blockable and its predecessor is blocked; if the predecessor of a blockable node x is blocked, then we say that x is **indirectly blocked**.

Generating and Shrinking Rules and Safe Neighbours The $\geqslant$-, $\exists$- and *NN*-rules are called **generating rules**, and the $\leqslant$- and the o-rule are called **shrinking rules**. An R-neighbour y of a node x is **safe** if (i) x is blockable or if (ii) x is a nominal node and y is not blocked.

Pruning When a node y is **merged** into a node x, we "prune" the completion graph by removing y and, recursively, all blockable successors of y. More precisely, pruning a node y (written $\mathsf{Prune}(y)$) in $\mathbf{G} = (V, E, \mathcal{L}, \neq)$ yields a graph that is obtained from $\mathbf{G}$ as follows:

1. for all successors z of y, remove $\langle y, z\rangle$ from E and, if z is blockable, $\mathsf{Prune}(z)$;
2. remove y from V.

Merging In some rules, we "merge" one node into another node. Intuitively, when we merge a node y into a node x, we add $\mathcal{L}(y)$ to $\mathcal{L}(x)$, "move" all the edges leading *to* y so that they lead to x and "move" all the edges leading from y to nominal nodes so that they lead from x to the same nominal nodes; we then remove y (and blockable sub-trees below y) from the completion graph. More precisely, merging a node y into a node x (written $\mathsf{Merge}(y, x)$) in $\mathbf{G} = (V, E, \mathcal{L}, \neq)$ yields a graph that is obtained from $\mathbf{G}$ as follows:

1. for all nodes z such that $\langle z, y\rangle \in E$
   (a) if $\{\langle x, z\rangle, \langle z, x\rangle\} \cap E = \emptyset$, then add $\langle z, x\rangle$ to E and set $\mathcal{L}(\langle z, x\rangle) = \mathcal{L}(\langle z, y\rangle)$,
   (b) if $\langle z, x\rangle \in E$, then set $\mathcal{L}(\langle z, x\rangle) = \mathcal{L}(\langle z, x\rangle) \cup \mathcal{L}(\langle z, y\rangle)$,
   (c) if $\langle x, z\rangle \in E$, then set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle x, z\rangle) \cup \{\mathsf{Inv}(S) \mid S \in \mathcal{L}(\langle z, y\rangle)\}$, and
   (d) remove $\langle z, y\rangle$ from E;
2. for all nominal nodes z such that $\langle y, z\rangle \in E$
   (a) if $\{\langle x, z\rangle, \langle z, x\rangle\} \cap E = \emptyset$, then add $\langle x, z\rangle$ to E and set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle y, z\rangle)$,
   (b) if $\langle x, z\rangle \in E$, then set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle y, z\rangle)$,
   (c) if $\langle z, x\rangle \in E$, then set $\mathcal{L}(\langle z, x\rangle) = \mathcal{L}(\langle z, x\rangle) \cup \{\mathsf{Inv}(S) \mid S \in \mathcal{L}(\langle y, z\rangle)\}$, and
   (d) remove $\langle y, z\rangle$ from E;
3. set $\mathcal{L}(x) = \mathcal{L}(x) \cup \mathcal{L}(y)$;
4. add $x \neq z$ for all z such that $y \neq z$; and
5. $\mathsf{Prune}(y)$.

If y was merged into x, we call x a **direct heir** of y, and we use being an **heir** of another node for the transitive closure of being a "direct heir".

Level (of Nominal Nodes) Let $o_1, \ldots, o_\ell$ be all the nominals occurring in the input concept D. We define the *level* of a node inductively as follows:

- each (nominal) node x with an $o_i \in \mathcal{L}(x)$, $1 \leq i \leq \ell$, is of level 0, and
- a nominal node x is of level i if x is not of some level $j < i$ and x has a neighbour that is of level $i - 1$.

Strategy (of Rule Application) The expansion rules in Figure 1 are applied according to the following strategy:

1. the o-rule is applied with highest priority,
2. next, the $\leqslant$- and the *NN*-rule are applied, and they are applied first to nominal nodes with lower levels (before they are applied to nodes with higher levels). In case they are both applicable to the same node, the *NN*-rule is applied first.
3. all other rules are applied with a lower priority.

We are now ready to finish the description of the tableau algorithm. A completion graph is **complete** if it contains a clash, or when none of the rules is applicable. If the expansion rules can be applied to C_0 and $\mathcal{R}$ in such a way that they yield a complete, clash-free completion graph, then the algorithm returns "C_0 is *satisfiable* w.r.t. $\mathcal{R}$", and "C_0 is *unsatisfiable* w.r.t. $\mathcal{R}$" otherwise.

Termination, Soundness, and Completeness

All but the Self–Ref-rule have been used before for fragments of $\mathcal{SROIQ}$, see Horrocks, Sattler, & Tobies (1999a); Horrocks & Sattler (2002, 2004), and the three $\forall_i$-rules are the obvious counterparts to the tableau conditions (P7)–(P9).

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

Theorem 16 (Termination, Soundness, and Completeness)

Let C_0 be a $\mathcal{SROIQ}$-concept in NNF and $\mathcal{R}$ a reduced Rbox.

1. *The tableau algorithm terminates when started with C_0 and $\mathcal{R}$.*
2. *The expansion rules can be applied to C_0 and $\mathcal{R}$ such that they yield a complete and clash-free completion graph if and only if there is a tableau for C_0 w.r.t. $\mathcal{R}$.*

Proof: (1): The algorithm constructs a graph that consists of a set of arbitrarily interconnected nominal nodes, and "trees" of blockable nodes with each tree rooted in r_0 or in a nominal node, and where branches of these trees might end in an edge leading to a nominal node.

Termination is a consequence of the usual $\mathcal{SHIQ}$ conditions with respect to the blockable tree parts of the graph, plus the fact that there is a bound on the number of new nominal nodes that can be added to $\mathbf{G}$ by the *NN*-rule.

The termination proof for the $\mathcal{SROIQ}$ tableaux is virtually identical to the one for $\mathcal{SHOIQ}$, whence we omit the

$\sqcap$-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–Ref-rule:	if $\exists S.\mathsf{Self} \in \mathcal{L}(x)$ or $\mathsf{Ref}(S) \in \mathcal{R}_a$, 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 $(\leqslant nS.C) \in \mathcal{L}(x)$, 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 1. $(\geqslant nS.C) \in \mathcal{L}(x)$, x is not blocked 2. there are not n safe S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$ 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. $(\leqslant nS.C) \in \mathcal{L}(z)$, z is not indirectly blocked 2. $\sharp S^{\mathbf{G}}(z, C) > n$ and there are two S-neighbours x, y of z with $C \in \mathcal{L}(x) \cap \mathcal{L}(y)$, and not $x \neq y$ then 1. if x is a nominal node then $\mathsf{Merge}(y, x)$ 2. else, if y is a nominal node or an ancestor of x then $\mathsf{Merge}(x, y)$ 3. else $\mathsf{Merge}(y, x)$
o-rule:	if for some $o \in N_I$ there are 2 nodes x, y with $o \in \mathcal{L}(x) \cap \mathcal{L}(y)$ and not $x \neq y$ then $\mathsf{Merge}(x, y)$
NN-rule:	if 1. $(\leqslant nS.C) \in \mathcal{L}(x)$, x is a nominal node, and there is a blockable S-neighbour y of x such that $C \in \mathcal{L}(y)$ and x is a successor of y, 2. there is no m such that $1 \leqslant m \leqslant n$, $(\leqslant mS.C) \in \mathcal{L}(x)$, and there exist m nominal S-neighbours $z_1, \ldots, z_m$ of x with $C \in \mathcal{L}(z_i)$ and $z_i \neq z_j$ for all $1 \leq i < j \leq m$. then 1. guess m with $1 \leqslant m \leqslant n$, and set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{(\leqslant mS.C)\}$ 2. create m new nodes $y_1, \ldots, y_m$ with $\mathcal{L}(\langle x, y_i\rangle) = \{S\}$, $\mathcal{L}(y_i) = \{C, o_i\}$, for each $o_i \in N_I$ new in $\mathbf{G}$, and $y_i \neq y_j$ for $1 \leq i < j \leq m$.

Figure 1: The Expansion Rules for the $\mathcal{SROIQ}$ Tableau Algorithm.

details and refer the reader to Horrocks & Sattler (2005). To see this, note first that the blocking technique employed for $\mathcal{SROIQ}$ is identical to the one for $\mathcal{SHOIQ}$. Next, the closure $\mathsf{fclos}(C_0, \mathcal{R})$ is defined differently, comprising concepts of the form $\forall \mathcal{B}_S(q).C$, generally yielding a size of $\mathsf{fclos}(C_0, \mathcal{R})$ that can be exponential in the depth of the role hierarchy. However, the construction of the automata can also be considered a pre-processing step and part of the input, in that case keeping the polynomial bound on the size of the closure relative to the input. Furthermore, it should be clear that the new Self–Ref-rule (only adding new reflexive edges) as well as the new clash conditions do not affect the termination of the algorithm.

(2): For the "if" direction, we can obtain a tableau $T = (\mathbf{S}, \mathcal{L}', \mathcal{E})$ from a complete and clash-free completion graph $\mathbf{G}$ by *unravelling* blockable "tree" parts of the graph as usual (these are the only parts where blocking can apply).

More precisely, paths are defined as follows. For a label blocked node x, let $b(x)$ denote a **node that blocks** x.

A **path** is a sequence of pairs of blockable nodes of $\mathbf{G}$ of the form $p = \langle (x_0, x'_0), \ldots, (x_n, x'_n)\rangle$. For such a path, we define $\mathsf{Tail}(p) := x_n$ and $\mathsf{Tail}'(p) := x'_n$. With $\langle p|(x_{n+1}, x'_{n+1})\rangle$ we denote the path

$$\langle (x_0, x'_0), \ldots, (x_n, x'_n), (x_{n+1}, x'_{n+1})\rangle.$$

The set $\mathsf{Paths}(\mathbf{G})$ is defined inductively as follows:

- For each blockable node x of $\mathbf{G}$ that is a successor of a nominal node or a root node, $\langle (x, x)\rangle \in \mathsf{Paths}(\mathbf{G})$, and
- For a path $p \in \mathsf{Paths}(\mathbf{G})$ and a blockable node y in $\mathbf{G}$:
  - if y is a successor of $\mathsf{Tail}(p)$ and y is not blocked, then $\langle p|(y, y)\rangle \in \mathsf{Paths}(\mathbf{G})$, and
  - if y is a successor of $\mathsf{Tail}(p)$ and y is blocked, then $\langle p|(b(y), y)\rangle \in \mathsf{Paths}(\mathbf{G})$.

Please note that, due to the construction of Paths, all nodes occurring in a path are blockable and, for $p \in \mathsf{Paths}(\mathbf{G})$ with $p = \langle p'|(x, x')\rangle$, x is not blocked, x' is blocked iff $x \neq x'$, and x' is never indirectly blocked. Furthermore, the blocking condition implies $\mathcal{L}(x) = \mathcal{L}(x')$.

Next, we use $\mathsf{Nom}(\mathbf{G})$ for the set of nominal nodes in $\mathbf{G}$, and define a tableau $T = (\mathbf{S}, \mathcal{L}', \mathcal{E})$ from $\mathbf{G}$ as follows.

$$\mathbf{S} = \mathsf{Nom}(\mathbf{G}) \cup \mathsf{Paths}(\mathbf{G})$$

$$\mathcal{L}'(p) = \begin{cases} \mathcal{L}(\mathsf{Tail}(p)) & \text{if } p \in \mathsf{Paths}(\mathbf{G}) \\ \mathcal{L}(p) & \text{if } p \in \mathsf{Nom}(\mathbf{G}) \end{cases}$$

$$\mathcal{E}(R) = E_1 \cup E_2 \cup E_3 \cup E_4, \text{ where}$$

$$E_1 = \{\langle p, q\rangle \in \mathsf{Paths}(\mathbf{G}) \times \mathsf{Paths}(\mathbf{G}) \mid q = \langle p|(x, x')\rangle \text{ and } x' \text{ is an } R\text{-successor of } \mathsf{Tail}(p), \text{ or } p = \langle q|(x, x')\rangle \text{ and } x' \text{ is an } \mathsf{Inv}(R)\text{-successor of } \mathsf{Tail}(q)\}$$

$$E_2 = \{\langle p, x\rangle \in \mathsf{Paths}(\mathbf{G}) \times \mathsf{Nom}(\mathbf{G}) \mid x \text{ is an } R\text{-neighbour of } \mathsf{Tail}(p)\}$$

$$E_3 = \{\langle x, p\rangle \in \mathsf{Nom}(\mathbf{G}) \times \mathsf{Paths}(\mathbf{G}) \mid \mathsf{Tail}(p) \text{ is an } R\text{-neighbour of } x\}$$

$$E_4 = \{\langle x, y\rangle \in \mathsf{Nom}(\mathbf{G}) \times \mathsf{Nom}(\mathbf{G}) \mid y \text{ is an } R\text{-neighbour of } x\}$$

We already commented above on $\mathbf{S}$, and $\mathcal{L}'$ is straightforward. Unfortunately, $\mathcal{E}$ is slightly cumbersome because we must distinguish between blockable and nominal nodes.

CLAIM: T is a tableau for C_0 with respect to $\mathcal{R}$.

Firstly, by definition of the algorithm, there is an heir x_0 of r_0 with $C_0 \in \mathcal{L}(x_0)$. By the $\leqslant$-rule, x_0 is either a root node or a nominal node, and thus cannot be blocked. Hence there is some $s \in \mathbf{S}$ with $C_0 \in \mathcal{L}'(s)$. Next, we prove that T satisfies each (Pi).

- (P1), (P2), (P5) and (P6) are trivially implied by the definition of $\mathcal{L}'$ and completeness of $\mathbf{G}$.
- (P3) and (P17) follow from the construction of $\mathcal{E}$ and completeness of $\mathbf{G}$, and (P4) follows from clash-freeness.
- for (P7), consider a tuple $\langle s,t\rangle \in \mathcal{E}(R)$ with $\forall\mathcal{B}(p).C \in \mathcal{L}'(s)$ and $p \xrightarrow{R} q \in \mathcal{B}(p)$. We have to show that $\forall\mathcal{B}(q).C \in \mathcal{L}'(t)$ and distinguish four different cases:
  - if $\langle s,t\rangle \in \mathsf{Paths}(\mathbf{G}) \times \mathsf{Paths}(\mathbf{G})$, then $\forall\mathcal{B}(p).C \in \mathcal{L}(\mathsf{Tail}(s))$ and
    * either $\mathsf{Tail}'(t)$ is an R-successor of $\mathsf{Tail}(s)$. Hence completeness implies $\forall\mathcal{B}(q).C \in \mathcal{L}(\mathsf{Tail}'(t))$, and by definition of $\mathsf{Paths}(\mathbf{G})$, either $\mathsf{Tail}'(t) = \mathsf{Tail}(t)$, or $\mathsf{Tail}(t)$ blocks $\mathsf{Tail}'(t)$ and the blocking condition implies $\mathcal{L}(\mathsf{Tail}'(t)) = \mathcal{L}(\mathsf{Tail}(t))$.
    * or $\mathsf{Tail}'(s)$ is an $\mathsf{Inv}(R)$-successor of $\mathsf{Tail}(t)$. Again, either $\mathsf{Tail}'(s) = \mathsf{Tail}(s)$, or $\mathsf{Tail}(s)$ blocks $\mathsf{Tail}'(s)$ in which case the blocking condition implies that $\forall\mathcal{B}(p).C \in \mathcal{L}(\mathsf{Tail}'(s))$, and thus completeness implies that $\forall\mathcal{B}(q).C \in \mathcal{L}(\mathsf{Tail}(t))$.
  - if $\langle s,t\rangle \in \mathsf{Nom}(\mathbf{G}) \times \mathsf{Nom}(\mathbf{G})$, then $\forall\mathcal{B}(p).C \in \mathcal{L}(s)$ and t is an R-neighbour of s. Hence completeness implies $\forall\mathcal{B}(q).C \in \mathcal{L}(t)$.
  - if $\langle s,t\rangle \in \mathsf{Nom}(\mathbf{G}) \times \mathsf{Paths}(\mathbf{G})$, then $\forall\mathcal{B}(p).C \in \mathcal{L}(s)$ and $\mathsf{Tail}(t)$ is an R-neighbour of s. Hence completeness implies $\forall\mathcal{B}(q).C \in \mathcal{L}(\mathsf{Tail}(t))$.
  - if $\langle s,t\rangle \in \mathsf{Paths}(\mathbf{G}) \times \mathsf{Nom}(\mathbf{G})$, then $\forall\mathcal{B}(p).C \in \mathcal{L}(\mathsf{Tail}(s))$ and t is an R-neighbour of $\mathsf{Tail}(s)$. Hence completeness implies $\forall\mathcal{B}(q).C \in \mathcal{L}(t)$.

  In all four cases, by definition of $\mathcal{L}'$, we have $\forall\mathcal{B}(q).C \in \mathcal{L}'(t)$.
- (P8) and (P9) follow from completeness of $\mathbf{G}$.
- for (P10), consider some $s \in \mathbf{S}$ with $\exists R.C \in \mathcal{L}'(s)$.
  - If $s \in \mathsf{Paths}(\mathbf{G})$, then $\exists R.C \in \mathcal{L}(\mathsf{Tail}(s))$, $\mathsf{Tail}(s)$ is not blocked, and completeness of $\mathcal{T}$ implies the existence of an R-neighbour y of $\mathsf{Tail}(s)$ with $C \in \mathcal{L}(y)$.
    * If y is a nominal node, then $y \in \mathbf{S}$, $C \in \mathcal{L}'(y)$, and $\langle s,y\rangle \in \mathcal{E}(R)$.
    * If y is blockable and a successor of $\mathsf{Tail}(s)$, then $\langle s|(\tilde{y},y)\rangle \in \mathbf{S}$, for $\tilde{y} = y$ or $\tilde{y} = b(y)$, $C \in \mathcal{L}'(\langle s|(\tilde{y},y)\rangle)$, and $\langle s, \langle s|(\tilde{y},y)\rangle\rangle \in \mathcal{E}(R)$.
    * If y is blockable and a predecessor of $\mathsf{Tail}(s)$, then $s = \langle p|(y,y)|(\mathsf{Tail}(s), \mathsf{Tail}'(s))\rangle$, $C \in \mathcal{L}'(\langle p|(y,y)\rangle)$, and
      $\langle s, \langle p|(y,y)\rangle\rangle \in \mathcal{E}(R)$.
  - If $s \in \mathsf{Nom}(\mathbf{G})$, then completeness implies the existence of some R-successor x of s with $C \in \mathcal{L}(x)$.
    * If x is a nominal node, then $\langle s,x\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}'(x)$.
    * If x is a blockable node, then x is a safe R-neighbour of s and thus not blocked. Hence there is a path $p \in \mathsf{Paths}(\mathbf{G})$ with $\mathsf{Tail}(p) = x$, $\langle s,p\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}'(p)$.
- (P11) and (P12) are immediate consequences of the definition of "R-successor" and "R-neighbour", as well as the definition of $\mathcal{E}$.
- for (P13), consider some $s \in \mathbf{S}$ with $(\leqslant nR.C) \in \mathcal{L}'(s)$. Clash-freeness implies the existence of at most n R-neighbours y_i of s with $C \in \mathcal{L}(y_i)$. By construction, each $t \in \mathbf{S}$ with $\langle s,t\rangle \in \mathcal{E}(R)$ corresponds to an R-neighbour y_i of s or $\mathsf{Tail}(s)$, and none of these R-neighbours gives rise to more than one such y_i. Moreover, since $\mathcal{L}'(t) = \mathcal{L}(y_i)$, (P13) is satisfied.
- for (P14), consider some $s \in \mathbf{S}$ with $(\geqslant nR.C) \in \mathcal{L}'(s)$.
  - if $s \in \mathsf{Nom}(\mathbf{G})$, then completeness implies the existence of n safe R-neighbours $y_1,\ldots,y_n$ of s with and $y_j \neq y_j$, for each $i \neq j$, and $C \in \mathcal{L}(y_i)$, for each $1 \leq i \leq n$. By construction, each y_i corresponds to a $t_i \in \mathbf{S}$ with $t_i \neq t_j$, for each $i \neq j$:
    * if y_i is blockable, then it cannot be blocked since it is a safe R-neighbour of s. Hence there is a path $\langle p|(y_i,y_i)\rangle \in \mathbf{S}$ and $\langle s, \langle p|(y_i,y_i)\rangle\rangle \in \mathcal{E}(R)$.
    * if y_i is a nominal node, then $\langle s,y_i\rangle \in \mathcal{E}(R)$.
  - if $s \in \mathsf{Paths}(\mathbf{G})$, then completeness implies the existence of n R-neighbours $y_1,\ldots,y_n$ of $\mathsf{Tail}(s)$ with $y_j \neq y_j$, for each $i \neq j$, and $C \in \mathcal{L}(y_i)$, for each $1 \leq i \leq n$. By construction, each y_i corresponds to a $t_i \in \mathbf{S}$ with $t_i \neq t_j$, for each $i \neq j$:
    * if y_i is safe, then it can be blocked if it is a successor of $\mathsf{Tail}(s)$. In this case, the "pair" construction in our definition of paths ensure that, even if $b(y_i) = b(y_j)$, for some $i \neq j$, we still have $\langle p|(b(y_i),y_i)\rangle \neq \langle p|(b(y_j),b_j)\rangle$.
    * if y_i is unsafe, then $\langle s,y_i\rangle \in \mathcal{E}(R)$.

    Hence all t_i are different and, by construction, $C \in \mathcal{L}'(t_i)$, for each $1 \leq i \leq n$.
- (P15) is satisfied due to completeness of $\mathbf{G}$ and the fact that each $t \in \mathbf{S}$ with $\langle s,t\rangle \in \mathcal{E}(R)$ corresponds to an R-neighbour of s (in case $s \in \mathsf{Nom}(\mathbf{G})$) or of $\mathsf{Tail}(s)$ (in case $s \in \mathsf{Paths}(\mathbf{G})$).
- (P16) and (P18) follow from clash-freeness and definition of $\mathcal{E}$, (P19) follows trivially from the initialisation of $\mathbf{G}$, and (P20) is due to completeness of $\mathbf{G}$ and the fact that nominal nodes are not "unravelled".

For the "only if" direction, given a tableau $T = (\mathbf{S}, \mathcal{L}', \mathcal{E})$ for C_0 w.r.t. $\mathcal{R}$, we can apply the non-deterministic rules, i.e., the $\sqcup$-, *choose*-, $\leqslant$-, and *NN*-rule, in such a way that we obtain a complete and clash-free graph: inductively with the generation of new nodes, we define a mapping π from nodes in the completion graph to individuals in $\mathbf{S}$ of the tableau in such a way that,

1. for each node x, $\mathcal{L}(x) \subseteq \mathcal{L}'(\pi(x))$,
2. for each pair of nodes x, y and each role R, if y is an R-successor of x, then $\langle \pi(x), \pi(y) \rangle \in \mathcal{E}(R)$, and
3. $x \neq y$ implies $\pi(x) \neq \pi(y)$.

This is analogous to the proof in Horrocks, Sattler, & Tobies (1999b) with the additional observation that, due to (P20), application of the o-rule does not lead to a clash of the form (7) as given in Definition 15. Similarly, an application of the Self–Ref-rule does not lead to a clash of the form (3) due to Condition (P4), a clash of the form (4) can not occur due to (P16), and a clash of the form (5) can not occur due to (P18).

□

From Theorems 9, 14 and 16, we thus arrive at the following theorem:

Theorem 17 (Decidability) *The tableau algorithm decides satisfiability and subsumption of $\mathcal{SROIQ}$-concepts with respect to Aboxes, Rboxes, and Tboxes.*

Outlook and Future Work

We introduced a description logic, $\mathcal{SROIQ}$, that overcomes certain shortcomings in expressiveness of other DLs. We have used $\mathcal{SHOIQ}$ and $\mathcal{RIQ}$ as a starting point, extended them with "useful-yet-harmless" expressive means, and extended the tableau algorithm accordingly. $\mathcal{SROIQ}$ is intended to be a basis for future extensions of OWL, and has already been adopted as the logical basis of OWL 1.1.

It is left for future work to determine whether the restrictions to simple roles can be relaxed, to pinpoint the exact computational complexity of $\mathcal{SROIQ}$, and to include further role assertions such as the more general version of antisymmetry to allow a better modeling of mereological notions (Goodwin, 2005).

A further line of investigation concerns concrete datatypes with inverse functional datatype properties: these are of interest since they allow to express simple key constraints. For instance, we might want to use a datatype property `SSN` for the social security number as a key for US citizen.

References

Areces, C.; Blackburn, P.; Hernandez, B.; and Marx, M. 2003. Handling Boolean Aboxes. In *Proc. of DL 2003*.

Baader, F., and Hanschke, P. 1991. A Schema for Integrating Concrete Domains into Concept Languages. In *Proc. of IJCAI-12*, 452–457.

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

Baader, F.; Lutz, C.; Milicic, M.; Sattler, U.; and Wolter, F. 2005. Integrating Description Logics and Action Formalisms: First Results. In *Proc. of AAAI-20*. AAAI Press.

Baader, F. 1991. Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles. In *Proc. of IJCAI-12*.

Blackburn, P., and Seligman, J. 1995. Hybrid languages. *J. of Logic, Language and Information* 4:251–272.

Casati, R., and Varzi, A. 1999. *Parts and Places. The Structure of Spatial Representation*. Cambridge, MA: MIT Press.

Goodwin, J. 2005. Experiences of Using OWL at the Ordnance Survey. In *Proc. of OWL: Experiences and Directions*.

Horrocks, I., and Sattler, U. 2001. Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In *Proc. of IJCAI-17*, 199–204.

Horrocks, I., and Sattler, U. 2002. Optimised reasoning for $\mathcal{SHIQ}$. In *Proc. of ECAI-15*.

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

Horrocks, I., and Sattler, U. 2005. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. In *Proc. of IJCAI-19*. Morgan Kaufmann, Los Altos.

Horrocks, I.; Kutz, O.; and Sattler, U. 2005. The Irresistible $\mathcal{SRIQ}$. In *Proc. of OWL: Experiences and Directions*.

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

Horrocks, I.; Sattler, U.; and Tobies, S. 1999a. Practical Reasoning for Expressive Description Logics. In Ganzinger, H.; McAllester, D.; and Voronkov, A., eds., *Proc. of LPAR-6*, volume 1705 of *LNAI*, 161–180. Springer.

Horrocks, I.; Sattler, U.; and Tobies, S. 1999b. Practical reasoning for expressive description logics. In Ganzinger, H.; McAllester, D.; and Voronkov, A., eds., *Proc. of LPAR-6*, number 1705 in LNAI, 161–180. Springer.

Horrocks, I.; Sattler, U.; and Tobies, S. 2000. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In MacAllester, D., ed., *Proc. of CADE-17*, volume 1831 of *LNCS*. Germany: Springer.

Pan, J., and Horrocks, I. 2003. Web ontology reasoning with datatype groups. In Fensel, D.; Sycara, K.; and Mylopoulos, J., eds., *Proc. of ISWC 2003*, number 2870 in LNCS, 47–63. Springer.

Schaerf, A. 1994. Reasoning with individuals in concept languages. *Data and Knowledge Engineering* 13(2):141–176.

Schild, K. 1991. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proc. of IJCAI-12*, 466–471.

Simons, P. 1987. *Parts: A Study in Ontology*. Oxford: Clarendon Press.

Wolstencroft, K.; Brass, A.; Horrocks, I.; Lord, P.; Sattler, U.; Turi, D.; and Stevens, R. 2005. A Little Semantic Web Goes a Long Way in Biology. In *Proc. of ISWC-4*, number 3729 in LNCS, 786–800. Springer.