The Even More Irresistible $\mathcal{SROIQ}$

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

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 practise. 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. 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, symmetric, transitive, 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 $\mathcal{SHOIQ}$. We believe that $\mathcal{SROIQ}$ could serve as a logical basis for possible future extensions of OWL-DL.

Keywords: description logics; KR languages; ontology methodology.

1 Introduction

We describe an extension, called $\mathcal{SROIQ}$, of the description logic (DL) $\mathcal{SHOIN}$ (14) underlying OWL-DL (9).[1] $\mathcal{SHOIN}$ can be said to provide most expressive means that one could reasonably expect from the logical basis of an ontology language, and to constitute a good compromise between expressive power and computational complexity/practicability of reasoning. However, it lacks e.g. qualified number restrictions which are present in the DL considered here since they are required in various applications (19) and do not pose problems (13). That is, we extend $\mathcal{SHOIQ}$—which is $\mathcal{SHOIN}$ with qualified number restrictions—and extend the work begun in (7).

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 will 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. 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` or `partOf` and `hasPart` should be declared as being disjoint.
2. *reflexive and irreflexive roles.* These features are of minor interest when considering TBoxes only, yet they add some useful constraints on ABoxes, especially in the presence of number restrictions. E.g., the role `knows` should be declared as being reflexive, and the role `hasSibling` should be declared as being irreflexive.
3. *negated role assertions.* Most Abox formalisms only allow for positive role assertions (with few exceptions (1; 5)), whereas $\mathcal{SROIQ}$ also allows for statements such as $(\mathtt{John}, \mathtt{Mary}) : \neg\mathtt{likes}$. 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.

[1] OWL also includes *datatypes*, a simple form of *concrete domain* (4). These can, however, be treated exactly as in $\mathcal{SHOQ}(\mathbf{D})/\mathcal{SHOQ}(\mathbf{D}_n)$ (10; 16), so we will not complicate our presentation by considering them here.

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 $\mathcal{RIQ}$ (12). 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: $\exists \texttt{owns}.\texttt{Car} \sqsubseteq \exists \texttt{owns}.\texttt{Engine}$.
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 (6). Nominals can also be viewed as a powerful generalisation of *ABox individuals* (17; 10). 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 $\exists \texttt{likes}.\mathsf{Self}$.

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 $\mathcal{SHIQ}$. The tableau algorithm for $\mathcal{SROIQ}$ presented here is essentially a combination of the algorithms for $\mathcal{RIQ}$ and $\mathcal{SHOIQ}$. 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.

More precisely, we employ the same technique using finite automata as in (12) 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 $R_1 \ldots R_n$ such that, in each model of the Rbox, $\langle x, y\rangle \in (R_1 \ldots R_n)^\mathcal{I}$ 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 $\forall R.C \in \mathcal{L}(x)$ and an $R_1 \ldots R_n$-neighbour y of x, whether to add C to $\mathcal{L}(y)$. 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.

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.

2 The Logic $\mathcal{SROIQ}$

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

2.1 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 subset 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*

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

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 a string of roles, it is convenient to extend both $\cdot^\mathcal{I}$ and $\mathsf{Inv}(\cdot)$ to such strings: if $w = R_1 \dots R_n$ for R_i roles, then we set $w^\mathcal{I} = R_1^\mathcal{I} \circ \dots \circ R_n^\mathcal{I}$ and $\mathsf{Inv}(w) = \mathsf{Inv}(R_n) \dots \mathsf{Inv}(R_1)$, 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 example, that a role R is reflexive or symmetric.

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 $w \mathrel{\dot\sqsubseteq} R$, where w is a finite string of roles not containing the universal role U, and R 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 $S_1 \ldots S_n \mathrel{\dot\sqsubseteq} R$, 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 $w \mathrel{\dot\sqsubseteq} R$ is* ***$\prec$-regular*** *if*

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

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

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 irreflexivity. Such cyclic dependencies are known to lead to undecidability (12).

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

Note that, due to the fourth restriction in the definition of $\prec$-regularity, we also restrict $\sqsubseteq^*$ 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 is fixed in any given model, we disallow the universal role to appear in role assertions.

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

$\mathcal{I} \models \mathsf{Sym}(R)$	*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*	$Diag^{\mathcal{I}} \subseteq R^{\mathcal{I}}$;
$\mathcal{I} \models \mathsf{Irr}(R)$	*if*	$R^{\mathcal{I}} \cap Diag^{\mathcal{I}} = \varnothing$;
$\mathcal{I} \models \mathsf{Dis}(R, S)$	*if*	$R^{\mathcal{I}} \cap S^{\mathcal{I}} = \varnothing$.

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, $RR \,\dot{\sqsubseteq}\, R$. Thus, as far as expressivity is concerned, we can assume w.l.o.g. 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 through 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 $\mathcal{SHIQ}$ (and $\mathcal{SHOIQ}$), the application of qualified number restrictions has to be restricted to certain roles, called *simple roles*, in order to preserve decidability (14). 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.

[2] This is not a serious restriction for, if $\mathcal{R}$ contains $\sqsubseteq^*$ 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).

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 called **simple in** $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$ is inductively defined as follows:

- a role name R is simple if there is no $w \dot{\sqsubseteq} R$ 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 $\mathsf{Irr}(R)$ or $\mathsf{Dis}(R, S)$ are 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 $\mathcal{R}$.*

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

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

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: $a : C$, $(a, b) : R$, $(a, b) : \neg S$, or $a \neq b$, for $a, b \in \mathbf{I}$ (the set of individual names), a (possibly inverse) role R, a (possibly inverse) simple role S, and a $\mathcal{SROIQ}$-concept C. A $\mathcal{SROIQ}$-**Abox** $\mathcal{A}$ is a finite set of individual assertions.*

Note that number restrictions $(\geqslant nS.C)$ and $(\leqslant nS.C)$, as well as the concept $\exists S.\mathsf{Self}$ and the disjointness and irreflexivity assertions for roles, $\mathsf{Dis}(R, S)$ and $\mathsf{Irr}(R)$, are all restricted to *simple* roles. In the case of number restrictions we mentioned the reason for this restriction above: without it, the satisfiability problem of $\mathcal{SHIQ}$-concepts is undecidable (14), 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 $\mathcal{SROIQ}$ and the remaining restrictions to simple roles in concept expressions as well as role assertions, it is part of future work to determine which of these restrictions to simple roles is strictly necessary in order to preserve decidability or practicability. This restriction, however, allows 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:*

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

*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 $\mathcal{T}$.*

*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*

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

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

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; 18; 3), the *internalisation* of GCIs is introduced, a technique that realises exactly this reduction. For $\mathcal{SROIQ}$, this technique can be modified accordingly. In particular, the expressivity of Rboxes in $\mathcal{SROIQ}$ and the presence of nominals allow to reduce $\mathcal{SROIQ}$ concept 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, 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 $\mathsf{Dis}(R, S)$ and $\mathsf{Ref}(R)$, and where the universal role U does not appear in C'.

For instance, to eliminate 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. 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. For more details, refer to (8).

Theorem 7 (Reduction)

1. *The satisfiability and subsumption problems 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 7, all standard inference problems for $\mathcal{SROIQ}$ 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 $\mathsf{Ref}(R)$ or $\mathsf{Dis}(R, S)$—we call such an Rbox **reduced**.

3 $\mathcal{SROIQ}$ is Decidable

In this section, we show that $\mathcal{SROIQ}$ is decidable. We have extended the tableau algorithms for $\mathcal{RIQ}$ and $\mathcal{SHOIQ}$ to $\mathcal{SROIQ}$, and will spend the remainder of this paper on its description.

In a first step, the tableau algorithm takes a reduced Rbox $\mathcal{R}$ and a concept C_0 and builds, for each (possibly inverse) role R occurring in $\mathcal{R}$ or C_0, a non-deterministic *finite automaton* $\mathcal{B}_R$.

Intuitively, such 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]

The following proposition states that $\mathcal{B}_R$ indeed captures all implications between (paths of) roles and R that are consequences of the role hierarchy $\mathcal{R}_h$, where $L(\mathcal{B}_R)$ denotes the language (a set of strings of roles) accepted by $\mathcal{B}_R$. The proof can be found in (8) or (12).

Proposition 8 (Automata) *$\mathcal{I}$ is a model of $\mathcal{R}_h$ if and only if, for each (possibly inverse) role R occurring in $\mathcal{R}_h$, each word $w \in L(\mathcal{B}_R)$, and each $\langle x, y\rangle \in w^{\mathcal{I}}$, we have $\langle x, y\rangle \in R^{\mathcal{I}}$.*

As usual, we first prove that concept satisfiability w.r.t. a reduced Rbox is equivalent to the existence of a *tableau* for C_0 and $\mathcal{R}$, that is, an abstraction of a model of C_0 and $\mathcal{R}$—for definitions and proofs, refer to (8).

Theorem 9 (Tableau) *A $\mathcal{SROIQ}$-concept C_0 is satisfiable w.r.t. a reduced $\mathcal{R}$ iff there exists a tableau for C_0 w.r.t. $\mathcal{R}$.*

The algorithm generates a *completion graph*, a structure that, if complete and clash-free, can be unravelled to an (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 9, if and only if the concept C_0 is satisfiable w.r.t. $\mathcal{R}$. Similar as for $\mathcal{RIQ}$, we define a set $\mathsf{fclos}(C_0, \mathcal{R})$ of "relevant sub-concepts" of those concepts occurring in C_0. Furthermore, all concepts are assumed to be given in *negation normal form* (NNF for short), i.e., negation occurs only in front of concept names or $\exists R.\mathsf{Self}$; see (8).

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

Definition 10 (Completion Graph) *Let $\mathcal{R}$ be a reduced Rbox, C_0 a concept in NNF not using the universal role, and $\mathbf{N}$ 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 $\mathbf{G}$.

*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. *for some $\mathsf{Dis}(R, S) \in \mathcal{R}_a$, y is an R- and an S-neighbour of x, or*
5. *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*
6. *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 starts with the completion graph $\mathbf{G} = (\{r_0, r_1 \ldots, r_\ell\}, \varnothing, \mathcal{L}, \varnothing)$ 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.

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 (15). Furthermore, to ensure termination, we need to apply similar strategies of rule application as were introduced for $\mathcal{SHOIQ}$, e.g., the o-rule is applied with highest priority.

For detailed definitions of the blocking technique and the terms and operations used in the (application of the) expansion rules, please consult (13; 8).

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.

All but the Self–Ref-rule have been used before for fragments of $\mathcal{SROIQ}$, see (14; 11; 12), and the three $\forall_i$-rules are the obvious counterparts to the tableau conditions (P4a), (P4b), and (P6) of (12).

Theorem 11 (Termination, Soundness, and Completeness) *Let C_0 be a $\mathcal{SROIQ}$-concept in NNF and $\mathcal{R}$ a reduced Rbox.*

- *The tableau algorithm terminates when started with C_0 and $\mathcal{R}$.*
- *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. $\mathcal{R}$.*

From Theorems 7, 9 and 11, we thus arrive at the following theorem:

Theorem 12 (Decidability) *The tableau algorithm decides satisfiability and subsumption of $\mathcal{SROIQ}$-concepts with respect to Aboxes, Rboxes, and Tboxes.*

Conclusion

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

$\sqcap$-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 $\{C_1, C_2\} \cap \mathcal{L}(x) = \varnothing$ 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 $\{C, \dot{\neg} C\} \cap \mathcal{L}(y) = \varnothing$ 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, and 2. there are not n safe S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{L}(y_i)$ and $y_i \not\doteq y_j$ for $1 \le i < j \le n$ 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 $y_i \not\doteq y_j$ for $1 \le i < j \le n$.
$\leqslant$-rule:	if 1. $(\leqslant nS.C) \in \mathcal{L}(z)$, z is not indirectly blocked, and 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 $x \not\doteq y$ 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 $x \not\doteq y$ 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 $z_i \not\doteq z_j$ for all $1 \le i < j \le m$. 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 $y_i \not\doteq y_j$ for $1 \le i < j \le m$,

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

References

[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., and Hanschke, P. A schema for integrating concrete domains into concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)* (1991), pp. 452–457.

[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] Blackburn, P., and Seligman, J. Hybrid languages. *J. of Logic, Language and Information 4* (1995), 251–272.

[7] Horrocks, I., Kutz, O., and Sattler, U. The Irresistible $\mathcal{SRIQ}$. In *Proc. of Workshop: OWL: Experiences and Directions, Galway, Ireland, November 11th-12th* (2005).

[8] Horrocks, I., Kutz, O., and Sattler, U. The Even More Irresistible $\mathcal{SROIQ}$. Tech. rep., University of Manchester, 2005. `http://www.cs.man.ac.uk/~sattler/publications/sroiq-tr.pdf`.

[9] 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.

[10] Horrocks, I., and Sattler, U. Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)* (2001), pp. 199–204.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] Pan, J., and Horrocks, I. Web ontology reasoning with datatype groups. In *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)* (2003), D. Fensel, K. Sycara, and J. Mylopoulos, Eds., no. 2870 in Lecture Notes in Computer Science, Springer, pp. 47–63.

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

[18] 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.

[19] 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.