A Better Uncle For OWL

Nominal Schemas for Integrating Rules and Ontologies

Markus Krötzsch
University of Oxford
markus.kroetzsch@comlab.ox.ac.uk

Frederick Maier, Adila A. Krisnadhi,
Pascal Hitzler
Kno.e.sis Center, Wright State University
{fred,adila,pascal}@knoesis.org

ABSTRACT

We propose a description-logic style extension of OWL 2 with *nominal schemas* which can be used like "variable nominal classes" within axioms. This feature allows ontology languages to express arbitrary *DL-safe rules* (as expressible in SWRL or RIF) in their native syntax. We show that adding nominal schemas to OWL 2 does not increase the worst-case reasoning complexity, and we identify a novel tractable language $\mathcal{SROELV}_3(\sqcap, \times)$ that is versatile enough to capture the lightweight languages OWL EL and OWL RL.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages; F.4.1 [**Mathematical Logic**]: Computational logic

General Terms

Languages, Complexity, Algorithms

Keywords

Web Ontology Language, Description Logic, $\mathcal{SROIQ}$, Semantic Web Rule Language, Datalog, tractability

1. INTRODUCTION

Despite significant recent progress, the search for a satisfactory integration of ontologies and rules for the Semantic Web is still ongoing [17, 23]. After the publication of the 2004 W3C Recommendation for the Web Ontology Language OWL [29], discussion of the problem centered on the *uncle rule*

$$\text{brotherOf}(x, y) \wedge \text{parentOf}(y, z) \to \text{uncleOf}(x, z),$$

which is easy to state using a simple rule language such as Datalog, but which cannot be modeled at all in the 2004 version of OWL. From the perspective of OWL design criteria, a core difficulty in allowing unrestricted Datalog rules along with OWL axioms is that it leads to undecidability of reasoning in the resulting combined language.

Subsequently, a significant body of work has developed, investigating the integration of description logics (DLs) [1], which form the basis for OWL, and rule languages (typically Datalog). Conceptually, one can distinguish two approaches. On the one hand, description logics have been extended with additional "description-logic-style" expressive features which make it possible to express certain types of rules. OWL 2 [36], the 2009 revision of the OWL W3C Recommendation, in fact can express the *uncle rule* mentioned above. By combining new features of OWL 2, many rules with a *tree-shaped body* can be expressed indirectly [25]. Decidability is nevertheless retained. Many rules, however, such as

$$\text{hasParent}(x,y) \wedge \text{hasParent}(x,z) \wedge \text{married}(y,z) \to C(x) \quad (1)$$

which defines a class C of children whose parents are married, are still not expressible.

On the other hand, there are approaches of a *hybrid* nature, in the sense that both OWL axioms and rules are syntactically allowed in ontologies, and a combined formal semantics defines how the hybrid language is to be understood. As already mentioned, such a combination generally leads to undecidability. This is the case for the *Semantic Web Rule Language* SWRL [19, 20], which is the most straightforward rule extension of OWL, and for the combination of OWL DL ontologies and the *Rule Interchange Format* RIF (even when restricted to RIF Core) [4, 11]. A prominently discussed idea for retaining decidability is to restrict the applicability of rules to *named individuals*, i.e., to logical constants that are explicitly mentioned in the ontology. Rules that are understood in this sense are called *DL-safe*, and the combination of OWL DL and DL-safe rules is indeed decidable [19, 33].

A generalization of DL-safe rules has been introduced in [26] in the form of *DL-safe variables*, as part of the definition of the tractable rule language ELP. Rather than restricting *all* variables in a (DL-safe) rule to binding only to known individuals, DL-safe variables allow the ontology engineer to explicitly specify the variables to be treated this way. This approach was subsequently generalized to obtain *DL+safe Rules* as a class of expressive rule languages for which reasoning is still decidable [23].

In this paper, we expand on the above idea and improve it in several ways. The key technical innovation is the introduction of *nominal schemas* as new elements of DL syntax. While the semantic intuition behind nominal schemas is the same as that behind DL-safe variables, the difference lies in the fact that DL-safe variables are tied to rule languages, while nominal schemas integrate seamlessly with DL syntax. As a consequence, the language which we propose encompasses DL-safe variable SWRL *while staying within the DL/OWL language paradigm*. It thus achieves within the DL framework what has hitherto only been achieved by hybrid approaches.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

To give an initial example, consider again the rule (1) extended by the axioms

$$\text{hasParent}(\text{mary}, \text{john}) \qquad (2)$$

$$(\exists\text{hasParent}.\exists\text{married}.\{\text{john}\})(\text{mary}) \qquad (3)$$

Axiom (2) asserts that John is a parent of Mary, while axiom (3) states that Mary belongs to the class of individuals with *some* (unnamed) parent who is married to John. Using a first-order logic semantics as in SWRL, rule (1) would thus entail that Mary belongs to the class C. Interpreting rule (1) as DL-safe, however, does not allow this conclusion, since John's spouse is not named by any constant in the ontology. To retain the conclusion, one can weaken this restriction to require only z to be DL-safe, while x and y can still take arbitrary values. This is possible in the rule-based approach of DL+safe Rules, but cannot be captured in an axiom of existing description logics.

In contrast, using nominal schemas, rule (1) can be expressed as

$$\exists\text{hasParent}.\{z\} \sqcap \exists\text{hasParent}.\exists\text{married}.\{z\} \sqsubseteq C. \qquad (4)$$

The desired conclusion again follows. The expression $\{z\}$ is a nominal schema, which is to be read as a *variable nominal* that can only represent nominals (i.e., z binds to known individuals), where the binding is the same for all occurrences of the nominal schema in an axiom.

The main contributions of this paper are as follows:

1. We introduce nominal schemas as a new general constructor for description logics, denoted by the letter $\mathcal{V}$ in the DL nomenclature, and define the expressive DL $\mathcal{SROIQV}$ as an extension of the description logic $\mathcal{SROIQ}$ underlying OWL 2.
2. We establish the complexity of the common reasoning tasks to be N2ExpTime-complete, and thus not harder than OWL 2 regarding worst-case complexity, even in the DL $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ that adds role constructors to $\mathcal{SROIQV}$.
3. We define $\mathcal{SROELV}_n(\sqcap, \times)$ ($n \geq 0$) as a new family of DLs with nominal schemas for which reasoning is possible in polynomial time. In particular, the DL $\mathcal{SROELV}_3(\sqcap, \times)$ is expressive enough to incorporate OWL EL and OWL RL, and to allow restricted semantic interaction between the two.

The expressivity of nominal schemas is also witnessed by the fact that it allows DLs to incorporate arbitrary DL-safe rules, given that concept intersections, existential role restrictions, and the universal (top) role are available. Since such rules preclude polytime reasoning, our tractable DLs $\mathcal{SROELV}_n(\sqcap, \times)$ employ restrictions on the number of certain occurrences of nominal schemas in each axiom.

The close relationship to nominals suggests simple ways of introducing nominal schemas into concrete syntactic forms of OWL 2, e.g. by using the existing syntax for nominal classes with special individual names that represent variables (using some suitable naming convention). This opens a path for introducing this feature into practical applications. While the above worst-case complexity result for $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ may seem encouraging, we believe that the tractable ontology language $\mathcal{SROELV}_3(\sqcap, \times)$ is the most promising candidate for implementations.

The paper is structured as follows. In Section 2 we introduce the syntax and semantics of nominal schemas based on the expressive DL $\mathcal{SROIQV}(\mathcal{B}_s, \times)$. The reasoning complexity of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ is studied in Section 3. Section 4 and 5 introduce the DLs $\mathcal{SROELV}_n(\sqcap, \times)$ and establish their tractability. In Section 6 we show how DL-safe rules can be expressed with nominal schemas, based on which we can explain the relationship of $\mathcal{SROELV}_3(\sqcap, \times)$ and the tractable profiles of OWL 2 in Section 7. We finish by discussing further related work (Section 8) before presenting our conclusions (Section 9).

2. NOMINAL SCHEMAS FOR OWL

We start by introducing *nominal schemas* as an extension of existing description logics. Our definition of the resulting DL is self-contained but cannot replace introductory texts like [1, 16, 23]. The Web Ontology Language OWL 2 DL is based on the description logic $\mathcal{SROIQ}$[1] but we base our extension on the slightly more expressive $\mathcal{SROIQ}(\mathcal{B}_s, \times)$. This DL extends $\mathcal{SROIQ}$ with boolean constructors ($\neg$, $\sqcap$, $\sqcup$) on simple roles, and with concept products $C \times D$ that allow the construction of roles as Cartesian products of concepts. It has been shown that this extension does not increase worst-case complexities of reasoning [23, 39].

The DL languages we study are based on a *signature* $\Sigma = \langle N_I, N_C, N_R, N_V\rangle$, where N_I, N_C, N_R, and N_V are finite and pairwise disjoint sets of *individual names*, *concept names*, *role names*, and *variables*. The set N_R is partitioned into disjoint sets N_R^s of *simple role names* and N_R^n of *non-simple role names*.

For the rest of this paper, we assume that a signature Σ has been fixed and so omit further references to it. The basic building blocks of DLs are concepts and (simple or non-simple) roles:

Definition 1. The sets $\mathbf{C}$ of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ *concepts* and $\mathbf{R}$ ($\mathbf{R}^{\mathrm{s}}/\mathbf{R}^{\mathrm{n}}$) of (*simple/non-simple*) $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ *roles* are defined by the following grammar:

$$\begin{aligned}
\mathbf{R}^{\mathrm{s}} &::= N_R^s \mid (N_R^s)^- \mid U \mid \neg\mathbf{R}^{\mathrm{s}} \mid \mathbf{R}^{\mathrm{s}} \sqcap \mathbf{R}^{\mathrm{s}} \mid \mathbf{R}^{\mathrm{s}} \sqcup \mathbf{R}^{\mathrm{s}} \mid N_C \times N_C\\
\mathbf{R}^{\mathrm{n}} &::= N_R^n \mid (N_R^n)^- \mid U \mid N_C \times N_C\\
\mathbf{R} &::= \mathbf{R}^{\mathrm{s}} \mid \mathbf{R}^{\mathrm{n}}\\
\mathbf{C} &::= \top \mid \bot \mid N_C \mid \{N_I\} \mid \{N_V\} \mid \neg\mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid\\
&\quad\ \exists\mathbf{R}.\mathbf{C} \mid \forall\mathbf{R}.\mathbf{C} \mid \exists\mathbf{R}^{\mathrm{s}}.\mathsf{Self} \mid \leqslant k\,\mathbf{R}^{\mathrm{s}}.\mathbf{C} \mid \geqslant k\,\mathbf{R}^{\mathrm{s}}.\mathbf{C}
\end{aligned}$$

where k is any non-negative integer. The constant U is the *universal role*, and $\top$ and $\bot$ are the top and bottom concepts. Concepts $\{a\}$ with $a \in N_I$ are called *nominals*, and concepts $\{x\}$ with $x \in N_V$ are called *nominal schemas*. The set $\mathbf{R}_{C\times D} = \{C \times D \mid C, D \in \mathbf{C}\} \subseteq \mathbf{R}$ is the set of all *concept products*.

Roles of the form R^- with $R \in N_R^s \cup N_R^n$ are called *inverse roles*. We define a function $\mathrm{Inv} : \mathbf{R} \to \mathbf{R}$ as follows: For $R \in N_R$, set $\mathrm{Inv}(R) \coloneqq R^-$ and $\mathrm{Inv}(R^-) \coloneqq R$. For $R, S \in \mathbf{R}$, set $\mathrm{Inv}(R \sqcap S) \coloneqq \mathrm{Inv}(R) \sqcap \mathrm{Inv}(S)$, $\mathrm{Inv}(R \sqcup S) \coloneqq \mathrm{Inv}(R) \sqcup \mathrm{Inv}(S)$, and $\mathrm{Inv}(\neg R) \coloneqq \neg\mathrm{Inv}(R)$. Set $\mathrm{Inv}(U) \coloneqq U$ and $\mathrm{Inv}(C \times D) :\,= D \times C$ for C, $D \in \mathbf{C}$.

$\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge bases are constructed from axioms as follows.

[1] The exact relationship is explained in [16]. Here we just note that OWL *classes* and *properties* are called *concepts* and *roles* in DL.

Definition 2. Given roles R, $S_i \in \mathbf{R}$, a generalized *role inclusion axiom (RIA)* is a statement of the form $S_1 \circ \cdots \circ S_k \sqsubseteq R$, with either $R \notin \mathbf{R}^{\mathrm{n}}$, or $k = 1$ and $S_1 \in \mathbf{R}^{\mathrm{s}}$. A set of RIAs is *regular* if there is a strict partial order $\prec$ on $\mathbf{R}$ such that

- if $R \notin \{S, \mathrm{Inv}(S)\}$, then $S \prec R$ if and only if $\mathrm{Inv}(S) \prec R$; and
- every RIA has the form $R \circ R \sqsubseteq R$, $\mathrm{Inv}(R) \sqsubseteq R$, $R \circ S_1 \circ \cdots \circ S_k \sqsubseteq R$, $S_1 \circ \cdots \circ S_k \circ R \sqsubseteq R$, or $S_1 \circ \cdots \circ S_k \sqsubseteq R$, with R, $S_i \in \mathbf{R}$ and $S_i \prec R$ for each $i \in \{1, \ldots, k\}$.

An *RBox axiom* is a RIA. A *TBox axiom* (or *general concept inclusion axiom, GCI*) is an expression $C \sqsubseteq D$ where $C, D \in \mathbf{C}$. An *ABox axiom* is any expression of the form $C(a)$ or $R(a, b)$ where $C \in N_C$, $R \in N_R$, and $a, b \in N_I$. A $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ *axiom* is any ABox, TBox, or RBox axiom, and a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ *knowledge base* is a regular set of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axioms.

Some presentations of $\mathcal{SROIQ}$ also include RBox axioms for *role characteristics* which we omit here as they can already be expressed in $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ anyway: the *empty (bottom) role* E ($\exists E.\top \sqsubseteq \bot$), role disjointness $\mathsf{disj}(S_1, S_2)$ ($S_1 \sqcap S_2 \sqsubseteq E$), asymmetry ($\mathsf{disj}(S, \mathrm{Inv}(S))$), reflexivity ($\top \sqsubseteq \exists S_{\mathsf{aux}}.\mathsf{Self}, S_{\mathsf{aux}} \sqsubseteq R$), irreflexivity ($\exists S.\mathsf{Self} \sqsubseteq \bot$), symmetry ($\mathrm{Inv}(R) \sqsubseteq R$), transitivity ($R \circ R \sqsubseteq R$).

An example of a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ TBox axiom has been given as axiom (4) above, where $\{z\}$ is a nominal schema. Intuitively, each nominal schema appearing in an axiom is universally quantified, but ranges only over elements that are referred to by an individual name.

Definition 3. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a domain of discourse $\Delta^{\mathcal{I}} \neq \emptyset$ and a function $\cdot^{\mathcal{I}}$ which maps N_C, N_R, and N_I to elements, sets, and relations of $\Delta^{\mathcal{I}}$ as shown in Table 1. A *variable assignment* $\mathcal{Z}$ for an interpretation $\mathcal{I}$ is a function $\mathcal{Z} : N_V \to \Delta^{\mathcal{I}}$ such that for each $v \in N_V$, $\mathcal{Z}(v) = a^{\mathcal{I}}$ for some $a \in N_I$. For any interpretation $\mathcal{I}$, assignment $\mathcal{Z}$, and $C_{(i)} \in \mathbf{C}$, $R_{(i)} \in N_R$, $t_{(i)} \in \mathbf{T}$, the function $\cdot^{\mathcal{I},\mathcal{Z}}$ is defined as shown in Table 1.

$\mathcal{I}$ and $\mathcal{Z}$ *satisfy* a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axiom α, written $\mathcal{I}, \mathcal{Z} \models \alpha$, if the corresponding condition shown in Table 1 holds. $\mathcal{I}$ *satisfies* α, written $\mathcal{I} \models \alpha$, if $\mathcal{I}, \mathcal{Z} \models \alpha$ for all variable assignments $\mathcal{Z}$ for $\mathcal{I}$. $\mathcal{I}$ *satisfies* a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge base KB, written $\mathcal{I} \models KB$, if $\mathcal{I} \models \alpha$ for all $\alpha \in KB$, and KB is *satisfiable* if such an $\mathcal{I}$ exists. The axiom α is *entailed* by KB, written $KB \models \alpha$, if all models of KB are also models of α.

The logic $\mathcal{SROIQ}(\mathcal{B}_s)$ is obtained from $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ by disallowing nominal schemas; concept products are already covered by $\mathcal{SROIQ}(\mathcal{B}_s)$ since they can be simulated using role negations [39]. The logic $\mathcal{SROIQ}$ is in turn obtained from $\mathcal{SROIQ}(\mathcal{B}_s)$ by disallowing boolean role constructors. In Section 6, we show that $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ is also expressive enough to encompass DL-safe rules (and thus DL-safe SWRL and DL-safe RIF-Core).

We note that it is straightforward to introduce nominal schemas into the normative RDF syntax for OWL 2 [37]. One way to do this would be to provide URIs for variables in the OWL namespace, used instead of individuals in `owl:oneOf` statements (which are used for the RDF syntax for nominals in OWL 2).

3. REASONING WITH $\mathcal{SROIQV}(\mathcal{B}_s, \times)$

We now show that the standard inferencing problems for $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge bases are decidable and have the same worst-case complexity as for $\mathcal{SROIQ}$. Specifically, they are N2ExpTime-complete.

Many common inference problems for DLs require us to check whether a certain axiom or set of axioms is entailed by the given knowledge base. For example, a concept C is subsumed by a concept D if $C \sqsubseteq D$ is entailed. It is well known that such entailment questions can be reduced to checking knowledge base satisfiability, i.e. asking whether some axiom is entailed is the same as asking if some (modified) knowledge base is satisfiable; see [1, 23] for details. Hence we focus on satisfiability checking only.

Reasoning with $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge bases can be simplified by first *grounding* them, i.e. eliminating nominal schemas by replacing them with the (finitely many) nominals that they can represent. Reasoning in the resulting $\mathcal{SROIQ}(\mathcal{B}_s)$ knowledge base is then possible as in [39].

Definition 4. If α is a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axiom, then its *grounding* $\mathsf{ground}(\alpha)$ is the set of all axioms that can be obtained by uniformly replacing nominal schemas in α with nominals of the given signature. Given a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge base KB, $\mathsf{ground}(KB) \coloneqq \bigcup_{\alpha \in KB} \mathsf{ground}(\alpha)$.

Theorem 1. *A $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge base KB is satisfiable if and only if* $\mathsf{ground}(KB)$ *is satisfiable. In particular, checking satisfiability of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge bases is decidable.*

Proof. Consider any interpretation $\mathcal{I}$ of the signature, which is the same for KB and $\mathsf{ground}(KB)$. For any TBox axiom $C \sqsubseteq D \in KB$ and a variable assignment $\mathcal{Z}$ for $\mathcal{I}$, it is clear that $\mathcal{I}, \mathcal{Z} \models C \sqsubseteq D$ iff $\mathcal{I} \models C' \sqsubseteq D'$ where $C' \sqsubseteq D' \in \mathsf{ground}(KB)$ was obtained from $C \sqsubseteq D$ by replacing each nominal schema $\{x\}$ with a nominal $\{c\}$ such that $c^{\mathcal{I}} = \mathcal{Z}(x)$. A suitable constant c must exist for $\mathcal{Z}$ by Definition 3. Conversely, every uniform replacement of nominal schemas in $C \sqsubseteq D$ corresponds to a choice of $\mathcal{Z}$. We conclude that $\mathcal{I} \models C \sqsubseteq D$ iff $\mathcal{I} \models \mathsf{ground}(C \sqsubseteq D)$. Analogous arguments apply to ABox and RBox axioms, showing the first part of the claim.

The second part follows as $\mathsf{ground}(KB)$ is a $\mathcal{SROIQ}(\mathcal{B}_s)$ knowledge base, so satisfiability can be checked as in [39]. □

Intuitively, every $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axiom represents an exponential number of $\mathcal{SROIQ}(\mathcal{B}_s)$ axioms that are obtained by grounding. This yields an upper bound for the complexity of reasoning with $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ that is exponentially larger than that of $\mathcal{SROIQ}(\mathcal{B}_s)$, i.e. N3ExpTime. We now prove that this result can be refined to obtain an N2ExpTime upper complexity bound, showing that this reasoning problem must be N2ExpTime-complete. To accomplish this, we extend the original proof for the worst-case complexity of $\mathcal{SROIQ}$ [22].

We first recall the complexity proof of [22], including its extension to $\mathcal{SROIQ}(\mathcal{B}_s)$ [39]. The proofs are based on an exponential reduction of DL knowledge bases to theories of $\mathcal{C}^2$, the two-variable fragment of first-order logic with counting quantifiers, for which satisfiability can be checked in NExpTime [38]. The reduction proceeds in three steps: (1) axioms are transformed into a simplified normal form, (2) complex RIAs are eliminated, and (3) the resulting axioms are expressed as formulae of $\mathcal{C}^2$.

Table 1: Semantics of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role name	V	$V^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual name	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
variable	x	$\mathcal{Z}(x) \in \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal (schema)	$\{t\}$	$\{t^{\mathcal{I},\mathcal{Z}}\}$
existential restriction	$\exists R.C$	$\{\delta \mid \text{there is } \epsilon \text{ with } \langle\delta,\epsilon\rangle \in R^{\mathcal{I},\mathcal{Z}} \text{ and } \epsilon \in C^{\mathcal{I},\mathcal{Z}}\}$
universal restriction	$\forall R.C$	$\{\delta \mid \text{for all } \epsilon \text{ with } \langle\delta,\epsilon\rangle \in R^{\mathcal{I},\mathcal{Z}} \text{ we have } \epsilon \in C^{\mathcal{I},\mathcal{Z}}\}$
self restriction	$\exists R.\mathsf{Self}$	$\{\delta \mid \langle\delta,\delta\rangle \in R^{\mathcal{I},\mathcal{Z}}\}$
concept complement	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{Z}}$
concept conjunction	$C \sqcap D$	$C^{\mathcal{I},\mathcal{Z}} \cap D^{\mathcal{I},\mathcal{Z}}$
concept disjunction	$C \sqcup D$	$C^{\mathcal{I},\mathcal{Z}} \cup D^{\mathcal{I},\mathcal{Z}}$
qualified number restrictions	$\leqslant n\, R.C$	$\{\delta \mid \#\{\langle\delta,\epsilon\rangle \in R^{\mathcal{I},\mathcal{Z}} \mid \epsilon \in C^{\mathcal{I},\mathcal{Z}}\} \leq n\}$
	$\geqslant n\, R.C$	$\{\delta \mid \#\{\langle\delta,\epsilon\rangle \in R^{\mathcal{I},\mathcal{Z}} \mid \epsilon \in C^{\mathcal{I},\mathcal{Z}}\} \geq n\}$
universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	V^-	$\{\langle\delta,\epsilon\rangle \mid \langle\epsilon,\delta\rangle \in V^{\mathcal{I},\mathcal{Z}}\}$
concept product	$A \times B$	$\{\langle\delta,\epsilon\rangle \mid \delta \in A^{\mathcal{I},\mathcal{Z}} \text{ and } \epsilon \in B^{\mathcal{I},\mathcal{Z}}\}$
role negation	$\neg R$	$(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I},\mathcal{Z}}$
role conjunction	$R \sqcap S$	$R^{\mathcal{I},\mathcal{Z}} \cap S^{\mathcal{I},\mathcal{Z}}$
role disjunction	$R \sqcup S$	$R^{\mathcal{I},\mathcal{Z}} \cup S^{\mathcal{I},\mathcal{Z}}$
concept assertion (ABox)	$A(t)$	$t^{\mathcal{I},\mathcal{Z}} \in A^{\mathcal{I},\mathcal{Z}}$
role assertion (ABox)	$V(t,u)$	$\langle t^{\mathcal{I},\mathcal{Z}}, u^{\mathcal{I},\mathcal{Z}}\rangle \in V^{\mathcal{I},\mathcal{Z}}$
TBox axiom	$C \sqsubseteq D$	$C^{\mathcal{I},\mathcal{Z}} \subseteq D^{\mathcal{I},\mathcal{Z}}$
RBox axiom (RIA)	$R \sqsubseteq S$	$R^{\mathcal{I},\mathcal{Z}} \subseteq S^{\mathcal{I},\mathcal{Z}}$
	$R_1 \circ \cdots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I},\mathcal{Z}} \circ \cdots \circ R_n^{\mathcal{I},\mathcal{Z}} \subseteq S^{\mathcal{I},\mathcal{Z}}$ where '$\circ$' denotes the usual composition of binary relations

$\mathcal{I}$ an interpretation; $\mathcal{Z}$ a variable assignment for $\mathcal{I}$; $A, B \in N_C$; $C, D \in \mathbf{C}$; $V \in N_R$; $R_{(i)}, S \in \mathbf{R}$; $a \in N_I$; $x \in N_V$; $t, u \in \mathbf{T}$.

Step (1) yields an equisatisfiable knowledge base that contains only axioms of the following forms:

$$\begin{array}{ccc} A \sqsubseteq \forall R.B & \bigsqcap A_i \sqsubseteq \bigsqcup B_j & S_1 \sqsubseteq S_2 \\ A \sqsubseteq \geqslant n\, S.B & A \equiv \{a\} & R_1 \sqsubseteq R^- \\ A \sqsubseteq \leqslant n\, S.B & A \equiv \exists S.\mathsf{Self} & R_1 \circ \cdots \circ R_n \sqsubseteq R \end{array}$$

where $R_{(i)} \in N_R$ and S_1, $S_2 \in \mathbf{R}^{\mathrm{s}}$, and $C \equiv D$ is short for $\{C \sqsubseteq D, D \sqsubseteq C\}$. This normalization can be done in linear time; see [39] for details. The only axioms that are not readily expressed in $\mathcal{C}^2$ are complex RIAs. They are eliminated next, with exponential effort.

Step (2) applies a technique from [12] using nondeterministic finite automata (NFA) to represent RIAs that entail non-simple roles. Suitable NFA for $\mathcal{SROIQ}$ were defined in [18, 21]. We do not repeat the details of this construction here, and merely quote the essential results. Proofs for the following facts can be found in [18] and the accompanying technical report.

Fact 1. *Consider a $\mathcal{SROIQ}$ knowledge base KB. For each (possibly inverse) role $R \in N_R^n$, there is an NFA $\mathcal{A}_R$ over the alphabet N_R such that the following holds for every model $\mathcal{I}$ of KB, and for every word $S_1 \ldots S_n$ accepted by $\mathcal{A}_R$: If $\langle\delta_i, \delta_{i+1}\rangle \in S_i^{\mathcal{I}}$ for all $i = 1, \ldots, n$, then $\langle\delta_1, \delta_{n+1}\rangle \in R^{\mathcal{I}}$.*

Moreover, let $\prec$ denote a strict linear order that witnesses regularity of KB as required in Definition 2. For each $R \in N_R^n$, the number of states of $\mathcal{A}_R$ is bounded exponentially in the depth *of KB that is defined as:*

$$\max\{n \mid \text{there are } S_1 \prec \ldots \prec S_n \text{ such that } T_{i1} \circ \ldots \circ S_i \circ \ldots \circ T_{im_i} \sqsubseteq S_{i+1} \in KB\}$$

It suffices to construct the respective NFA for non-simple roles, and our additional role expressions thus do not interfere with this construction. Now step (2) proceeds by replacing every axiom of the form $A \sqsubseteq \forall R.B$ by the following set of axioms, where $\mathcal{A}_R$ is the NFA as introduced above, and X_q are fresh concept names for each state q of $\mathcal{A}_R$:

$$\begin{array}{rl} A \sqsubseteq X_q & q \text{ is the initial state of } \mathcal{A}_R \\ X_q \sqsubseteq \forall S.X_{q'} & \mathcal{A}_R \text{ has a transition } q \xrightarrow{S} q' \\ X_q \sqsubseteq B & q \text{ is a final state of } \mathcal{A}_R \end{array}$$

Moreover, all complex RIAs of the form $S_1 \circ \ldots \circ S_n \sqsubseteq R$ with $n \geq 2$ are deleted. The number of new axioms (and fresh concept names) that are introduced for each axiom of the form $A \sqsubseteq \forall R.B$ is bounded by the sum of the number of states and transitions in $\mathcal{A}_R$, and the number of transitions in turn is linear in the number of role names and states. According to Fact 1, the number of axioms introduced for each axiom $A \sqsubseteq \forall R.B$ is exponentially bounded in the depth of the knowledge base. The overall size of the knowledge base after step (2) therefore is bounded by a function that is linear in the size of the knowledge base and exponential in the depth of the knowledge base.

Step (3), finally, is a simple rewriting to $\mathcal{C}^2$ that does not increase the size of the knowledge base. To obtain the main result of this section, it suffices to observe that grounding does not increase the depth of the knowledge base:

THEOREM 2. *The problem of deciding satisfiability in $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ KBs is* N2EXPTIME*-complete.*

PROOF. By adopting a result from [23, Proposition 5.2.1], one can show that for each $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge base KB, one can find an equisatisfiable $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge base without $\times$, the size of which is linear in the size of KB. Thus assume that KB has no concept products. The depth of KB is only affected by RBox axioms. In the absence of concept products, RBox axioms are not affected by grounding, hence the depth of $\mathsf{ground}(KB)$ is equal to the depth of KB.

Since $\mathsf{ground}(KB)$ is in $\mathcal{SROIQ}(\mathcal{B}_s)$, one can apply the transformation steps (1)–(3). This yields a $\mathcal{C}^2$ theory T that is equisatisfiable to $\mathsf{ground}(KB)$ [39] and thus to KB (Theorem 1). The size of T is linear in the size of $\mathsf{ground}(KB)$ and exponential in the depth of KB. Both measures are exponential in the size of KB, and so is T. Deciding satisfiability of T can be done in NEXPTIME [38], thus deciding satisfiability of KB in N2EXPTIME.

$\mathcal{SROIQV}(\mathcal{B}_s, \times)$ includes $\mathcal{SROIQ}$, for which deciding satisfiability is N2EXPTIME-hard [22]. From this, hardness follows. □

4. A TRACTABLE FRAGMENT

The result that reasoning in $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ has the same worst-case complexity as $\mathcal{SROIQ}$ (and OWL 2) is encouraging, yet we are far from a practical reasoning procedure for this DL. In particular, Theorem 2 is based on a procedure that still takes exponentially longer than the original approach for $\mathcal{SROIQ}$, without this affecting the worst-case complexity. In this section, we therefore focus on identifying cases where inferencing is possible in polynomial time. This still leads to a rather expressive tractable DL. Subsequent sections will highlight the relationship to the tractable profiles of OWL 2.

Concretely, we define DLs $\mathcal{SROELV}_n(\sqcap, \times)$ for each integer $n \geq 0$, n restricting the number of "problematic" occurrences of nominal schemas detailed below. The DLs are based on the tractable DL $\mathcal{SROEL}(\sqcap, \times)$, introduced as an extension of OWL EL [24]. In essence, $\mathcal{SROEL}(\sqcap, \times)$ is $\mathcal{SROIQ}(\mathcal{B}_s)$ restricted to operators $\sqcap$, $\exists$ (possibly with Self), $\circ$, and some uses of $\times$. To preserve tractability when adding nominal schemas, we must avoid the increase in the number of axioms during grounding, which is exponential in the number of nominal schemas per axiom.

Unfortunately, one cannot reduce the number of nominal schemas by normal form transformations in general, since they represent complex dependencies that cannot be simplified. But there are special cases where nominal schemas on the left-hand side of TBox axioms can be eliminated, or separated using independent axioms. One such case was identified in [26] for the rule language ELP: if the dependencies expressed in a rule body are *tree-shaped* then the rule can always be reduced to a small set of normalized rules with a limited number of variables in each. For example, a rule body that consists of a conjunction $A(x) \wedge R(x, z) \wedge S(x, y) \wedge B(y) \wedge T(y, z)$ is not tree-shaped since there are parallel paths $x \xrightarrow{R} z$ and $x \xrightarrow{S} y \xrightarrow{T} z$ in the corresponding dependency structure. In our case, binary predicates are role names, unary predicates are concept names, and constant symbols correspond to nominals. Variables can either be "hidden" in the structure of the DL concept expression, or occur explicitly as nominal schemas (the latter are called *DL-safe variables* in ELP). For example, the above rule body can be expressed as a concept $A \sqcap \exists R.\{z\} \sqcap \exists S.(B \sqcap \exists T.\{z\})$.

Here, we do not introduce tree-shaped dependency structures as a general mechanism for ensuring that normal form transformations are possible, and merely identify sufficient conditions for which this is the case. This allows us to provide somewhat simpler proofs. An obvious condition that implies tree-shaped dependencies is that a nominal schema occurs only once, and only on the left-hand side of a TBox axiom. As in [26], the tree-shape only refers to variables (DL-safe or not), not to constants, in rule bodies. This means that nominals (our syntax for constants) disconnect a concept's dependency structure. E.g., if B in the above rule body is replaced by a nominal $\{a\}$, then the concept would be tree-shaped. In such a case, we say that the nominal $\{z\}$ occurs in a *safe environment*, as defined next.

Definition 5. An occurrence of a nominal schema $\{x\}$ in a concept C is *safe* if C has a sub-concept of the form $\{a\} \sqcap \exists R.D$ for some $a \in N_I$, such that D contains the occurrence of $\{x\}$ but no other occurrence of any nominal schema. In this case, $\{a\} \sqcap \exists R.D$ is a *safe environment* for this occurrence of $\{x\}$. $S(a, x)$ will sometimes be used to denote an expression of the form $\{a\} \sqcap \exists R.D$ within which $\{x\}$ occurs safely.

A nominal schema $\{x\}$ is *safe* for a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ TBox axiom $C \sqsubseteq D$ if $\{x\}$ does not occur in D, and at most one occurrence of $\{x\}$ in C is not safe.

Definition 6. Let $n \geq 0$. A $\mathcal{SROELV}_n(\sqcap, \times)$ *concept* is a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ concept that may contain $\top$, $\bot$, $\sqcap$, $\exists$, Self, nominals and nominal schemas, but which does not contain $\sqcup$, $\neg$, $\forall$, $\leqslant k$, and $\geqslant k$. $\mathcal{SROELV}_n(\sqcap, \times)$ *roles* (simple or non-simple) are $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ *roles* (simple or non-simple) that may contain $\sqcap$ (for simple roles) and U but no inverse roles, $\sqcup$, or $\neg$.

A $\mathcal{SROELV}_n(\sqcap, \times)$ *TBox axiom* is a $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ TBox axiom α that uses $\mathcal{SROELV}_n(\sqcap, \times)$ concepts only, and where at most n nominal schemas are not safe for α. An *RBox axiom* of $\mathcal{SROELV}_n(\sqcap, \times)$ is an RBox axiom of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ using only $\mathcal{SROELV}_n(\sqcap, \times)$ roles. An *ABox axiom* of $\mathcal{SROELV}_n(\sqcap, \times)$ is the same as an ABox axiom of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$.

For a knowledge base KB and role $R \in N_R$, let $\mathsf{ran}(R)$ be the set of all concepts $B \in N_C$ for which there is a set of the form $\{R \sqsubseteq R_1, R_1 \sqsubseteq R_2, \ldots, R_{n-1} \sqsubseteq R_n, R_n \sqsubseteq A \times B\} \subseteq KB$ with $n \geq 0$ and $R_0 = R$. KB is a $\mathcal{SROELV}_n(\sqcap, \times)$ *knowledge base* if $R_1 \circ \cdots \circ R_n \sqsubseteq S$ implies $\mathsf{ran}(S) \subseteq \mathsf{ran}(R_n)$, and $R_1 \sqcap R_2 \sqsubseteq S$ implies $\mathsf{ran}(S) \subseteq \mathsf{ran}(R_1) \cup \mathsf{ran}(R_2)$.

The additional condition using ran is called *admissibility of range restrictions*, see [24] for details. As explained below, restricting to at most n non-safe nominal schemas per axiom ensures that at most $|N_I|^n$ axioms are introduced during grounding. We will fix n at a constant small value, so this increase is polynomial. It is easy to see that axiom (4) is an example of a $\mathcal{SROELV}_1(\sqcap, \times)$ axiom.

5. REASONING WITH $\mathcal{SROELV}_n(\sqcap, \times)$

If n is constant, the problem of checking satisfiability in $\mathcal{SROELV}_n(\sqcap, \times)$ is possible in polynomial time w.r.t. the size of the knowledge base. To show this, we provide a polynomial transformation to the DL $\mathcal{SROEL}(\sqcap, \times)$, which was shown to be tractable in [24].

Let KB be a $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base. We define a $\mathcal{SROEL}(\sqcap, \times)$ knowledge base $\mathsf{ground}^+(KB)$ as follows. The RBox and ABox of $\mathsf{ground}^+(KB)$ are the same as the RBox and ABox of KB. For each TBox axiom $\alpha = C \sqsubseteq D \in KB$, the following axioms are added to $\mathsf{ground}^+(KB)$:

1. For each nominal schema $\{x\}$ safe for α, with safe occurrences in environments $S_i(a_i, x)$ for $i = 1, \ldots, l$, introduce a fresh concept name $O_{x,\alpha}$. For every individual $b \in N_I$ in KB, $\mathsf{ground}^+(KB)$ contains an axiom

   $$\prod_{i=1}^{l} \exists U.S_i(a_i, b) \sqsubseteq \exists U.(\{b\} \sqcap O_{x,\alpha}),$$

   where $S_i(a_i, b)$ denotes $S_i(a_i, x)$ with $\{x\}$ replaced by $\{b\}$, and the empty conjunction ($l = 0$) denotes $\top$.

2. A concept C' is obtained from C as follows. Initialize $C' \coloneqq C$. For each nominal schema $\{x\}$ that is safe for α: (a) replace all safe occurrences $S(a, x)$ in C' by $\{a\}$; (b) replace the non-safe occurrence (if any) of $\{x\}$ in C' by $O_{x,\alpha}$; (c) set $C' \coloneqq C' \sqcap \exists U.O_{x,\alpha}$. After these steps, C' contains only nominal schemas that are not safe for α, and neither for $C' \sqsubseteq D$.

   Now add axioms $\mathsf{ground}(C' \sqsubseteq D)$ to $\mathsf{ground}^+(KB)$.

THEOREM 3. *Given a $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base KB, the size of $\mathsf{ground}^+(KB)$ is exponential in n and polynomial in the size of KB.*

PROOF. The size of the RBox and ABox of $\mathsf{ground}^+(KB)$ is linear in the size of KB and does not depend on n. If m is the number of individual names in KB, then step 1 above introduces at most mk axioms for each axiom α with k nominal schemas. This is polynomial in the size of KB. The second step introduces $|\mathsf{ground}(C' \sqsubseteq D)|$ many axioms, and hence at most m^n axioms for each α. □

THEOREM 4. *A $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base KB is satisfiable if and only if $\mathsf{ground}^+(KB)$ is satisfiable.*

PROOF. We first introduce some notation to simplify the proof. Let C be a $\mathcal{SROELV}_n(\sqcap, \times)$ concept. A *position* is a word $p \in \{1, 2\}^*$. The *sub-concept $C|p$ of C at position p* is defined recursively, where ε denotes the empty word: $D|_\varepsilon \coloneqq D$; $(D_1 \sqcap D_2)|_{pi} \coloneqq D_i|_p$ ($i \in \{1, 2\}$); $\exists R.D|_{p1} \coloneqq D|_p$. The *positions of C* are the positions p for which $C|_p$ is defined. Consider an interpretation $\mathcal{I}$ and variable assignment $\mathcal{Z}$ such that $\delta \in C^{\mathcal{I},\mathcal{Z}}$ for some $\delta \in \Delta^{\mathcal{I}}$. We non-deterministically define *witnesses* $\delta_p \in \Delta^{\mathcal{I}}$ for all positions p of C, such that $\delta_p \in (C|_p)^{\mathcal{I},\mathcal{Z}}$. Set $\delta_\varepsilon \coloneqq \delta$. For the recursion, assume that δ_p has been defined. If $C|_p = D_1 \sqcap D_2$, then $\delta_{p1} \coloneqq \delta_p$ and $\delta_{p2} \coloneqq \delta_p$. If $C|_p = \exists R.D$, then $\delta_p \in (C|_p)^{\mathcal{I},\mathcal{Z}}$ implies that there is some ϵ with $\langle \delta_p, \epsilon \rangle \in R^{\mathcal{I}}$ and $\epsilon \in D^{\mathcal{I},\mathcal{Z}}$. Set $\delta_{p1} \coloneqq \epsilon$. Below, the selection of δ_p is always assumed to be arbitrary but fixed. To clarify the context, we say that δ_p is a witness *for $\delta \in C^{\mathcal{I},\mathcal{Z}}$*. Intuitively, δ_p thus witnesses a substructure of $\mathcal{I}$ that satisfies the semantic conditions for $\delta \in C^{\mathcal{I},\mathcal{Z}}$.

An interpretation $\mathcal{I}$ for $\mathsf{ground}^+(KB)$ is *O-minimal* if the following holds for all concept names $O_{x,\alpha}$ introduced in step 1 and all $b \in N_I$ in KB: $\mathcal{I} \models O_{x,\alpha}(b)$ iff $\prod_{i=1}^{l} \exists U.S_i(a_i, b)^{\mathcal{I}} \neq \emptyset$ (using the notation from step 1). Clearly, every model of KB can be extended to an O-minimal interpretation of $\mathsf{ground}^+(KB)$. Conversely, if $\mathsf{ground}^+(KB)$ is satisfiable, it surely has an O-minimal model. Namely, any model can be made O-minimal by reducing the extensions of $O_{x,\alpha}$ as required. This does not affect the truth of other axioms, since $O_{x,\alpha}$ only occurs in positive (non-negated) positions in the premise of TBox axioms – reducing its extension makes the premise smaller while not affecting the conclusion.

So the claim can be obtained by showing: the O-minimal extension of every model of KB is a model of $\mathsf{ground}^+(KB)$, and every O-minimal model of $\mathsf{ground}^+(KB)$ is a model of KB if we ignore the interpretation of concepts $O_{x,\alpha}$. This can be shown individually for each axiom of KB. It is immediate for ABox and RBox axioms.

Now consider a TBox axiom $\alpha = C \sqsubseteq D \in KB$, and the axiom $C' \sqsubseteq D$ constructed in step 2. By the proof of Theorem 1, the models of $C' \sqsubseteq D$ and $\mathsf{ground}(C' \sqsubseteq D)$ are the same. To complete the proof, we show that an O-minimal model satisfies $C \sqsubseteq D$ iff it satisfies $C' \sqsubseteq D$. This is implied by the following property ($*$): given an O-minimal interpretation $\mathcal{I}$ of $\mathsf{ground}^+(KB)$, we find that $\delta \in C^{\mathcal{I},\mathcal{Z}}$ for some variable assignment $\mathcal{Z}$ iff $\delta \in C'^{\mathcal{I},\mathcal{Z}'}$ for some variable assignment $\mathcal{Z}'$, where assignments in either direction of the claim can be chosen such that $D^{\mathcal{I},\mathcal{Z}} = D^{\mathcal{I},\mathcal{Z}'}$. Indeed, if $\mathcal{I} \models C \sqsubseteq D$ and $\delta \in C'^{\mathcal{I},\mathcal{Z}'}$, then by ($*$) we obtain $\mathcal{Z}$ with $\delta \in C^{\mathcal{I},\mathcal{Z}}$, thus $\delta \in D^{\mathcal{I},\mathcal{Z}}$, and $\delta \in D^{\mathcal{I},\mathcal{Z}'}$. The reverse direction is similar.

We show both directions of ($*$) by induction over the steps used to construct C' from C. The claim holds initially when $C = C'$. Now consider the modifications (a)–(c) for one nominal schema $\{x\}$ that is safe for α, and let C_0, C_a, C_b, and C_c denote C' in its initial state and after each of the steps (a)–(c).

For the "only if" direction, assume $\delta \in C_0^{\mathcal{I},\mathcal{Z}}$. We show $\delta \in C_c^{\mathcal{I},\mathcal{Z}}$, i.e., we set $\mathcal{Z}' \coloneqq \mathcal{Z}$. Let $S_i(a_i, x)$ be as in step 1. Since all $S_i(a_i, x)$ are sub-concepts of C_0 and $C_0^{\mathcal{I},\mathcal{Z}} \neq \emptyset$, we can apply our initial construction of witnesses δ_p to conclude $S_i(a_i, b)^{\mathcal{I}} \neq \emptyset$ for all $i \in \{1, \ldots, l\}$ and some b with $b^{\mathcal{I}} = \mathcal{Z}(x)$. As $\mathcal{I}$ satisfies the axioms of step 1 (by O-minimality), this shows $\exists U.O_{x,\alpha}^{\mathcal{I}} = \top^{\mathcal{I}}$. We thus have $C_c^{\mathcal{I},\mathcal{Z}} = C_b^{\mathcal{I},\mathcal{Z}}$. Now C_b is obtained by replacing sub-concepts E of C_0 by concepts E' in (a) and (b). Since all such E occur positively in C_0, the claim follows from $E^{\mathcal{I},\mathcal{Z}} \subseteq E'^{\mathcal{I},\mathcal{Z}}$. For (a) this follows since $\delta_{S_i(a_i,x)} = a_i^{\mathcal{I}} \in \{a_i\}^{\mathcal{I}}$. For (b), we have $E = \{x\}$, and thus $\mathcal{Z}(x) = b^{\mathcal{I}}$. Hence $\delta_{S_i(a_i,x)} \in S_i(a_i, b)$ for all $i \in \{1, \ldots, l\}$ so $E^{\mathcal{I},\mathcal{Z}} \subseteq E'^{\mathcal{I},\mathcal{Z}}$ follows the fact that $\mathcal{I}$ satisfies the axioms of step 1.

For the "if" direction, assume $\delta \in C_c^{\mathcal{I},\mathcal{Z}'}$. For the concept introduced in (c), we have $\exists U.O_{x,\alpha}^{\mathcal{I}} \neq \emptyset$, so $O_{x,\alpha}^{\mathcal{I}} \neq \emptyset$, and thus $\exists U.O_{x,\alpha}^{\mathcal{I}} = \Delta^{\mathcal{I}}$. This shows $C_c^{\mathcal{I},\mathcal{Z}'} = C_b^{\mathcal{I},\mathcal{Z}'}$. If an occurrence of $\{x\}$ at position p in C_a was replaced by $O_{x,\alpha}$ in (b), then set $\epsilon \coloneqq \delta_p$, where δ_p is a witness for $\delta \in C_b^{\mathcal{I},\mathcal{Z}'}$. Otherwise, if $C_a = C_b$, then select $\epsilon \in O_{x,\alpha}^{\mathcal{I}}$ arbitrarily. Define $\mathcal{Z}$ such that $\mathcal{Z}(x) = \epsilon$ and $\mathcal{Z}(y) = \mathcal{Z}'(y)$ for all $y \neq x$. We claim $\delta \in C_0^{\mathcal{I},\mathcal{Z}}$. We showed $C_c^{\mathcal{I},\mathcal{Z}'} = C_b^{\mathcal{I},\mathcal{Z}'}$ above, and as x does not occur in C_b, we have $C_b^{\mathcal{I},\mathcal{Z}'} = C_b^{\mathcal{I},\mathcal{Z}}$. Thus $\delta \in C_b^{\mathcal{I},\mathcal{Z}}$. Now consider the witnesses δ_q for $\delta \in C_b^{\mathcal{I},\mathcal{Z}}$. It is easy to see that the same elements can be chosen as witnesses for $\delta \in C_0^{\mathcal{I},\mathcal{Z}}$, showing that the latter holds. This follows since $\epsilon \in \{x\}^{\mathcal{I},\mathcal{Z}}$ (for the sub-concept replaced in (b)), and since $\delta_{S(a_i,x)} = a_i^{\mathcal{I}} \in \{a_i\}^{\mathcal{I}}$ (for the sub-concepts replaced in (a)). □

A knowledge base is unsatisfiable if and only if it entails $\{a\} \sqsubseteq \bot$ for arbitrary $a \in N_I$. This reduces satisfiability testing to instance retrieval (checking if a is an instance of $\bot$). Using the polynomial time instance retrieval method for $\mathcal{SROEL}(\sqcap, \times)$ from [24] together with Theorems 3 and 4, we thus obtain the following result. Hardness for P follows from the hardness of $\mathcal{SROEL}(\sqcap, \times)$.

Theorem 5. *If KB is a $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base of size s, satisfiability of KB can reduced to instance retrieval w.r.t. a set of Datalog rules of size proportional to s^n and at most four variables per rule. If n is constant, the problem is* P*-complete.*

6. DL-SAFE RULES

As shown here, an interesting feature of nominal schemas is that they can be used to express arbitrary DL-safe rules [33]. These are Datalog rules with unary and binary predicates that are restricted – just like nominal schemas – to apply to domain elements that are represented by individual names.[2] Identifying unary predicates with concept names, binary predicates with role names, constants with individual names, and (DL-safe) variables with the variables in nominal schemas, the syntax of DL-safe rules can be based on a DL signature. As before, we assume the signature $\Sigma = \langle N_I, N_C, N_R, N_V \rangle$ to be fixed and omit explicit references to it. The set of *terms* $\mathbf{T}$ of Σ is $N_I \cup N_V$. We can now define the syntax and semantics of DL-safe rules.

Definition 7. A *concept atom* is an expression of the form $A(t)$ with $t \in \mathbf{T}$ and $A \in N_C \cup \{\top, \bot\}$. A *role atom* is an expression of the form $R(s,t)$ with $s, t \in \mathbf{T}$ and $R \in N_R$. An *atom* is a concept or role atom.

If B is a finite and non-empty set of atoms and H is an atom, then $B \to H$ is a *DL-safe rule. B* is called the *body*, and H is called the *head*. A DL-safe rule that contains at most n distinct variables is called an *n-variable rule.*

Definition 8. Interpretations $\mathcal{I}$ and *variable assignments* $\mathcal{Z}$ for DL-safe rules are defined as in Definition 3. An atom F is *satisfied* by $\mathcal{I}$ and $\mathcal{Z}$, written $\mathcal{I}, \mathcal{Z} \models F$, if either $F = A(t)$ and $t^{\mathcal{I},\mathcal{Z}} \in A^{\mathcal{I},\mathcal{Z}}$, or $F = R(s,t)$ and $\langle s^{\mathcal{I},\mathcal{Z}}, t^{\mathcal{I},\mathcal{Z}} \rangle \in A^{\mathcal{I},\mathcal{Z}}$. A set of atoms B is *satisfied* by $\mathcal{I}$ and $\mathcal{Z}$, written $\mathcal{I}, \mathcal{Z} \models B$, if $\mathcal{I}, \mathcal{Z} \models F$ for all $F \in B$.

$\mathcal{I}$ *satisfies* a DL-safe rule $B \to H$, written $\mathcal{I} \models B \to H$, if for all assignments $\mathcal{Z}$ for $\mathcal{I}$, either $\mathcal{I}, \mathcal{Z} \models H$ or $\mathcal{I}, \mathcal{Z} \not\models H$. A set of rules is satisfied if all of its elements are. Models, satisfiability, and entailment are defined as in Definition 3.

The above provides a first-order logic semantics for DL-safe rules that is fully compatible with the semantics of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$: it uses the same kinds of models. As such, it is meaningful to combine DL-safe rules and DL knowledge bases. The entailment relation is immediate: a DL-safe rule or DL axiom φ is entailed by a DL knowledge base *KB* extended with a set of rules *RB* if φ is satisfied by all interpretations that satisfy both *KB* and *RB*.

DL-safe rules can also be used to capture the fragment of the rule language Datalog with predicates of arity at most 2, given that we are interested in the first-order semantics of such Datalog rules.

DL-safe rules can be transformed into $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axioms as follows.

Definition 9. A syntactic transformation dl from atoms and DL-safe rules to $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ concepts and TBox axioms is defined as follows. For a unary atom $A(t)$, set $\mathsf{dl}(A(t)) := \exists U.(\{t\} \sqcap A)$; for a binary atom $R(s,t)$, set $\mathsf{dl}(R(s,t)) := \exists U.(\{s\} \sqcap \exists R.\{t\})$. For a DL-safe rule $B \to H$, set $\mathsf{dl}(B \to H) := \bigsqcap_{F \in B} \mathsf{dl}(F) \sqsubseteq \mathsf{dl}(H)$. A set of DL-safe rules *RB* is translated as $\mathsf{dl}(RB) := \bigcup_{B \to H \in RB} \mathsf{dl}(B \to H)$.

The function dl transforms rules into $\mathcal{SROELV}_n(\sqcap, \times)$ TBox axioms, where n is the number of variables in the rule. This ensures that none of the restrictions on simple and non-simple roles, regularity, or admissibility of ranges in $\mathcal{SROELV}_n(\sqcap, \times)$ are violated. In consequence, $\mathsf{dl}(RB)$ is a $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base if *RB* is a set of n-variable rules. Before showing that dl actually preserves the semantics of DL-safe rules, we present a useful lemma.

Lemma 1. *For an atom F, interpretation $\mathcal{I}$, and variable assignment $\mathcal{Z}$ we have (1) $\mathcal{I}, \mathcal{Z} \models F$ if and only if $\mathsf{dl}(F)^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$; (2) $\mathcal{I}, \mathcal{Z} \not\models F$ if and only if $\mathsf{dl}(F)^{\mathcal{I},\mathcal{Z}} = \emptyset$.*

Proof. Since $\mathsf{dl}(F)$ has the form $\exists U.D$, there are only two options: either $D^{\mathcal{I},\mathcal{Z}} \neq \emptyset$ and $\mathsf{dl}(F)^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$, or $D^{\mathcal{I},\mathcal{Z}} = \emptyset$ and $\mathsf{dl}(F)^{\mathcal{I},\mathcal{Z}} = \emptyset$. Thus (2) is obtained as the contrapositive of (1), and we only need to show the latter.

Given $\mathsf{dl}(F) = \exists U.D$, it thus suffices to show that $\mathcal{I}, \mathcal{Z} \models F$ iff $D^{\mathcal{I},\mathcal{Z}} \neq \emptyset$. If $F = A(t)$, this claims that $t^{\mathcal{I},\mathcal{Z}} \in A^{\mathcal{I},\mathcal{Z}}$ is equivalent to $(A \sqcap \{t\})^{\mathcal{I},\mathcal{Z}} \neq \emptyset$. For $F = R(s,t)$, the claim is that $\langle s^{\mathcal{I},\mathcal{Z}}, t^{\mathcal{I},\mathcal{Z}} \rangle \in R^{\mathcal{I},\mathcal{Z}}$ is equivalent to $(\{s\} \sqcap \exists R.\{t\})^{\mathcal{I},\mathcal{Z}} \neq \emptyset$. Both claims are clear from Definitions 3 and 8. □

Theorem 6. *The models of a set RB of DL-safe rules are the same as the models of $\mathsf{dl}(RB)$, i.e. RB and $\mathsf{dl}(RB)$ are semantically equivalent.*

Proof. Consider interpretation $\mathcal{I}$, an assignment $\mathcal{Z}$ for $\mathcal{I}$, and a rule $B \to H \in RB$ such that $\mathsf{dl}(B \to H) = C_B \sqsubseteq C_H$. Lifting Lemma 1 to sets of atoms B, we find $\mathcal{I}, \mathcal{Z} \not\models B$ iff $C_B^{\mathcal{I},\mathcal{Z}} = \emptyset$. It is clear that this entails the claim: either $\mathcal{I}, \mathcal{Z} \models H$ and $C_H^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$, or $\mathcal{I}, \mathcal{Z} \not\models B$ and $C_B^{\mathcal{I},\mathcal{Z}} = \emptyset$. □

Importantly, this result confirms that nominal schemas are powerful enough to express arbitrary DL-safe rules. The use of nominal schemas, however, in $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ is more general than the extension of $\mathcal{SROIQ}(\mathcal{B}_s)$ with DL-safe rules, since the latter correspond to a special form of $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ axioms only. Combining Theorem 5 with the observation that $\mathsf{dl}(RB)$ is linear in the size of *RB*, we can state the following:

Theorem 7. *The problem of deciding whether a knowledge base $RB \cup KB$ is satisfiable, where RB is a set of n-variable rules with n constant, and KB is a $\mathcal{SROELV}_n(\sqcap, \times)$ knowledge base, is* P*-complete.*

[2] We here formalize DL-safe rules by restricting permissible variable assignments. An equivalent alternative is to enforce this restriction by additional premises in rules. See [16] for a discussion.

Table 2: OWL RL normal forms

$A \sqsubseteq C$	$A \sqcap B \sqsubseteq C$	$R \sqsubseteq T$	$R^- \sqsubseteq T$
$A \sqsubseteq \forall R.C$	$A \sqsubseteq \leqslant 1\, R.C$	$R \circ S \sqsubseteq T$	
$A \sqsubseteq \{a\}$	$\{a\} \sqsubseteq C$	$R \sqcap S \sqsubseteq T$	

7. RELATION TO OWL PROFILES

The OWL 2 standard proposes three *tractable profiles*, i.e. language fragments for which reasoning is possible in (sub)polynomial time [31]. Each of the profiles is closely related to a description logic: OWL EL is contained in $\mathcal{SROEL}(\times,\sqcap)$ [24], OWL RL is an extension of DLP [15], and OWL QL is based on DL-Lite [10]. All OWL 2 profiles include special support for datatypes and concrete data values that we have not considered here. The respective extensions would not lead to any technical difficulty, since dataype literals can be treated like individuals in all profiles.

Here, we note that with certain restrictions (specified below), the profiles OWL RL and OWL EL are contained within $\mathcal{SROELV}_3(\sqcap,\times)$. The latter also covers most features of OWL QL, but we also note that the typical application areas of this profile are not targeted by our approach.

The relation to OWL EL is obvious: a $\mathcal{SROEL}(\times,\sqcap)$ knowledge base is also a $\mathcal{SROELV}_n(\sqcap,\times)$ knowledge base, for each n (including $n = 3$). So this approach subsumes the profile OWL EL without datatypes. Similar to OWL EL, OWL RL disallows any kind of (truly) disjunctive information, but it also excludes all forms of existential quantification. In return, this allows OWL RL to include inverse roles and unrestricted range restrictions which are excluded from OWL EL to preserve tractability. Due to the lack of disjunctive and existential expressions, however, OWL RL axioms can be faithfully represented using DL-safe rules only.

Concretely, OWL RL is based on a *Horn Description Logic*, the axioms of which can be transformed into a number of normal forms as shown in [27]. As discussed in [23], the TBox and RBox axioms of OWL RL can thus be reduced to the normal forms in Table 2, where A, B, $C \in N_C \cup \{\top,\bot\}$, R, S, $T \in N_R$, and $a \in N_I$. Only three of these axiom types are not in $\mathcal{SROELV}_n(\sqcap,\times)$. Using DL-safe rules, we can encode $A \sqsubseteq \forall R.C$ into $A(x), R(x,y) \to C(y)$, and $R^- \sqsubseteq T$ into $R(x,z) \to T(z,x)$. For qualified functionality restrictions, an auxiliary "DL-safe equality" role $R_\approx$ is encoded with the axiom $\{x\} \sqcap \exists R_\approx.\{y\} \sqsubseteq \exists U.(\{x\} \sqcap \{y\})$. The axiom $A \sqsubseteq \leqslant 1\, R.C$ is then represented by the DL-safe rule

$$A(x), R(x,y_1), C(x_1), R(x,y_2), C(x_2) \to R_\approx(y_1,y_2). \qquad (5)$$

Transforming DL-safe rules as in Section 6 and keeping ABox axioms without modification, we thus obtain a simple translation from OWL RL (without datatype-related features) to $\mathcal{SROELV}_3(\sqcap,\times)$.

OWL QL, finally, is based on DL-Lite, which is designed for its sub-polynomial AC$_0$ data complexity [10]. This also implies that no complex RIAs are included, but inverse roles and some forms of existential quantification are allowed. Inverse roles R^- can be replaced by new role names R_{inv}, with the original semantics approximated by DL-safe rules $R(x,y) \to R_{\text{inv}}(y,x)$ and $R_{\text{inv}}(x,y) \to R(y,x)$. As in the case of OWL RL, this means that certain conclusions are lost, while tractability is preserved. Furthermore, axioms of the form $\top \sqsubseteq \exists R^-.C$ can be expressed as $R \sqsubseteq \top \times C$. The related restrictions of Definition 6 do not apply in the absence of complex RIAs.

It must be noted that OWL RL axioms that are translated to DL-safe rules are no longer interpreted under their first-order semantics. Entailments of ABox axioms – the main inference task in applications of OWL RL – are preserved, but translated axioms are not semantically equivalent to the original ontology. Indeed, the DL obtained by allowing unions of OWL EL and OWL RL knowledge bases is 2ExpTime-complete, as it encompasses all features of Horn-$\mathcal{SROIQ}$ [35]. The combination of "DL-safe" OWL RL and OWL EL, in contrast, is still tractable, but does not entail all inferences that the unrestricted combination would.

Also, it should be pointed out that the typical uses of OWL QL for ontology-based querying of large datasets is not supported by $\mathcal{SROEL}(\times,\sqcap)$, at least not as it is by OWL QL. Namely, the low data complexity of OWL QL enables an efficient way of query rewriting that is not available in $\mathcal{SROEL}(\times,\sqcap)$. This limitation cannot be overcome, as AC$_0 \subsetneq$ P. Conversely, $\mathcal{SROEL}(\times,\sqcap)$ includes many features not available in OWL QL, e.g, role transitivity.

8. RELATED WORK

8.1 Description Logic Rules

The DLs introduced here are closely related to *Description Logic Rules*, i.e. first-order rule languages that allow sets of rules to be expressed in description logics [23, 25]. As discussed in Section 6, nominal schemas are closely related to variables in *DL-safe rules* [33].

In [26], *ELP* was proposed as an extension of DL Rules for $\mathcal{EL}^{++}$ with additional *DL-safe variables*. ELP is more general than the mere union of DL-safe rules and DL Rules, since a single rule can contain some variables that are DL-safe and others that are not. However, the definitions in [26] aim at tractability, and do not allow all uses of DL-safe variables. As discussed in Section 4, safe environments in $\mathcal{SROELV}_n(\sqcap,\times)$ are closely related to this approach.

DL-safe variables in ELP do not encompass the unrestricted use of up to n nominal schemas that is permitted in $\mathcal{SROELV}_n(\sqcap,\times)$. Like the general use of nominal schemas in $\mathcal{SROIQV}(\mathcal{B}_s,\times)$, this feature corresponds rather to *DL+safe Rules* as introduced in [23]. DL+safe Rules are obtained by allowing DL-safe variables to be used in DL Rules, and they can be viewed as a rule version of our approach. Our complexity proofs in Section 5 are based on analogous proofs for DL+safe Rules. In [23], the term "variable nominals" had been proposed for nominal schemas, but no according DL syntax was introduced.

8.2 Existential Rules

Another approach toward integrating ontological modeling and rules are *existential rules*, that extend Datalog with existential quantifiers in rule heads. This paradigm has attracted much interest recently, and has been studied under a variety of names such as *Datalog+/–*, *∀∃-rules*, and – primarily in the database community – *tuple-generating dependencies* (TGDs) [2, 3, 6, 7, 8, 9, 13, 14]. As in the combination of rules and DL, reasoning with existential rules is undecidable without further restrictions. A chief interest of many of the above works thus is to establish formalisms for which (conjunctive) query answering is decidable, possibly with a low data complexity. For example, it has been shown that certain dialects of Datalog+/– capture and extend languages of the DL-Lite family [7, 10].

Although the general motivation of this research is similar, there are significant technical differences to our approach. In particular, none of the above rule languages is expressive enough to capture OWL EL. Moreover, features like cardinality restrictions (or equality constraints) and disjunctive modeling are hardly considered in current works. On the other hand, many decidable fragments of existential rules are highly expressive and exhibit combined complexities of ExpTime and 2ExpTime.

8.3 Description Graphs

Extending DLs with description graphs and rules, as proposed in [30], enables a more explicit way of modeling structured objects such as those found in the medical sciences and ontologies. As noted in [32], in order to retain decidability, DLs are often designed with some syntactic restrictions which limit their ability to model non-tree-like structures as featured in the *uncle* example earlier. On the other hand, description graphs allow us to explicitly state relationships between domain elements in the knowledge base which cannot be expressed using standard DL constructs alone. In addition, this approach also allows description graphs to appear as *graph atoms* in rules, thus allowing conditional statements about the structured objects modeled by the graphs.

In its unrestricted form, extending DLs with description graphs leads to undecidability even under acyclicity conditions. Therefore, some restrictions to prevent roles in DL axioms from interacting with roles appearing in graphs and rules are imposed to retain decidability. Corresponding decidability results for $\mathcal{SHOQ}^+$ and $\mathcal{SHOIQ}^+$ have been established. However, we are not aware of any tractable fragments of these formalisms that have been proposed.

Note that $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ knowledge bases also induce graph-like structures due to the presence of nominals and nominal schemas, as evidenced by the fact that they can model DL-safe rules. Yet, neither of the two languages – $\mathcal{SROIQV}(\mathcal{B}_s, \times)$ and DLs extended with description graphs and rules – contains the other. A merging of both approaches is conceivable, but remains to be worked out.

9. CONCLUSIONS AND FUTURE WORK

We have introduced nominal schemas as an extension to DL-based ontology languages, that provides sufficient expressivity to incorporate rule-based modeling into ontologies. In particular, it supports the integration of Web rule languages such as SWRL and RIF with OWL 2 ontologies. An important next step is to realize these ideas for the concrete serialization formats of these languages, and to make the corresponding modeling features available in practice.

The latter task especially includes the implementation of inference algorithms to handle nominal schemas more efficiently. We have shown that our extension does not increase the worst-case complexity of reasoning in OWL 2, and that versatile tractable sub-languages exist. Whether and how these theoretical results can be put into efficient reasoning algorithms is an important research question. Two different approaches seem viable to address this problem. On the one hand, nominal schemas could be implemented by modifying/extending existing OWL 2 implementations that have good support for nominals, such as the OWL 2 reasoner *HermiT* [34]. This can be accomplished by treating nominal schemas like nominals in the deduction procedure, instantiating them with concrete individuals only when this enables relevant deduction steps. This can be viewed as a method of *deferred grounding*.

On the other hand, our light-weight ontology languages could be implemented using rule-based procedures as proposed for $\mathcal{SROEL}$ [24]. In this setting, nominal schemas can be treated like DL-safe variables. Thus, the rule-based deduction remains similar with the only modification that some variables can only be instantiated with certain constants (the approach in [24] introduces new constant symbols for eliminating existentials, and DL-safe variables are not allowed to represent these auxiliary symbols).

In conclusion, the close relationship to nominals is not merely of syntactic convenience, but prepares a path for the further practical adoption of this feature. Instead of a paradigm shift from ontologies to rules, existing applications could be augmented with bits of rule-based modeling to overcome restrictions of classical DLs. Nominal schemas thus may provide an exceptional opportunity for enhancing the expressive power of ontologies without giving up on established tools, format, or methodologies.

Acknowledgements This work was partially supported by the National Science Foundation under award 1017225 "III: Small: TROn—Tractable Reasoning with Ontologies" and by EPSRC in project "HermiT: Reasoning with Large Ontologies" (EP/F065841/1). The third author acknowledges support by a Fulbright Indonesia Presidential Scholarship PhD Grant 2010.

10. REFERENCES

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, second edition, 2007.

[2] J.-F. Baget, M. Leclère, and M.-L. Mugnier. Walking the decidability line for rules with existential variables. In Lin et al. [28], pages 466–476.

[3] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In Craig Boutilier, editor, *Proc. 21st Int. Conf. on Artificial Intelligence (IJCAI'09)*, pages 677–682. IJCAI, 2009.

[4] H. Boley, G. Hallmark, M. Kifer, A. Paschke, A. Polleres, and D. Reynolds, editors. *RIF Core Dialect.* W3C Recommendation, 22 June 2010. Available at http://www.w3.org/TR/rif-core/.

[5] G. Brewka and J. Lang, editors. *Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08).* AAAI Press, 2008.

[6] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In Brewka and Lang [5], pages 70–80.

[7] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In Jan Paredaens and Jianwen Su, editors, *Proc. 28th Symposium on Principles of Database Systems (PODS'09)*, pages 77–86. ACM, 2009.

[8] Andrea Calì, Georg Gottlob, and Andreas Pieris. Advanced processing for ontological queries. *Proceedings of VLDB 2010*, 3(1):554–565, 2010.

[9] Andrea Calì, Georg Gottlob, and Andreas Pieris. Query answering under non-guarded rules in Datalog+/-. In Pascal Hitzler and Thomas Lukasiewicz, editors, *Proc. 4th Int. Conf. on Web Reasoning and Rule Systems (RR 2010)*, volume 6333 of *LNCS*, pages 1–17. Springer, 2010.

[10] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[11] J. de Bruijn. *RIF RDF and OWL Compatibility.* W3C Recommendation, 22 June 2010. Available at http://www.w3.org/TR/rif-rdf-owl/.

[12] S. Demri and H. Nivelle. Deciding regular grammar logics with converse through first-order logic. *J. of Logic, Language and Information*, 14(3):289–329, 2005.

[13] Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In Diego Calvanese, Maurizio Lenzerini, and Rajeev Motwani, editors, *Proc. 9th Int. Conf. on Database Theory (ICDT'03)*, volume 2572 of *LNCS*, pages 225–241. Springer, 2003.

[14] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

[15] B.N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *Proc. 12th Int. Conf. on World Wide Web (WWW'03)*, pages 48–57. ACM, 2003.

[16] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies.* Chapman & Hall/CRC, 2009.

[17] P. Hitzler and B. Parsia. Ontologies and rules. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 111–132. Springer, 2 edition, 2009.

[18] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In P. Doherty, J. Mylopoulos, and C.A. Welty, editors, *Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 57–67. AAAI Press, 2006.

[19] I. Horrocks, P.F. Patel-Schneider, S. Bechhofer, and D. Tsarkov. OWL Rules: A proposal and prototype implementation. *J. of Web Semantics*, 3(1):23–40, 2005.

[20] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A Semantic Web Rule Language.* W3C Member Submission, 21 May 2004. See http://www.w3.org/Submission/SWRL/.

[21] I. Horrocks and U. Sattler. Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. *Artificial Intelligence*, 160(1):79–104, 2004.

[22] Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In Brewka and Lang [5], pages 274–284.

[23] M. Krötzsch. *Description Logic Rules*, volume 008 of *Studies on the Semantic Web.* IOS Press/AKA, 2010.

[24] M. Krötzsch. Efficient inferencing for OWL EL. In T. Janhunen and I. Niemelä, editors, *Proc. 12th European Conf. on Logics in Artificial Intelligence (JELIA'10)*, volume 6341 of *LNAI*, pages 234–246. Springer, 2010.

[25] M. Krötzsch, S. Rudolph, and P. Hitzler. Description logic rules. In M. Ghallab et al., editors, *Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008*, pages 80–84. IOS Press, 2008.

[26] M. Krötzsch, S. Rudolph, and P. Hitzler. ELP: Tractable rules for OWL 2. In A. Sheth et al., editors, *Proceedings of the 7th International Semantic Web Conference (ISWC-08)*, volume 5318 of *Lecture Notes in Computer Science*, pages 649–664. Springer, 2008.

[27] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Complexity boundaries for Horn description logics. In *Proc. 22nd AAAI Conf. on Artificial Intelligence (AAAI'07)*, pages 452–457. AAAI Press, 2007.

[28] F. Lin, U. Sattler, and M. Truszczynski, editors. *Proc. 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'10).* AAAI Press, 2010.

[29] D.L. McGuinness and F. van Harmelen, editors. *OWL Web Ontology Language Overview.* W3C Recommendation, 10 February 2004. Available at http://www.w3.org/TR/owl-features/.

[30] B. Motik, B. Cuenca Grau, I. Horrocks, and U. Sattler. Representing ontologies using description logics, description graphs, and rules. *Artificial Intelligence*, 173(14):1275–1309, 2009.

[31] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz, editors. *OWL 2 Web Ontology Language: Profiles.* W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-profiles/.

[32] B. Motik and R. Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5), 2010.

[33] B. Motik, U. Sattler, and R. Studer. Query answering for OWL DL with rules. *J. of Web Semantics*, 3(1):41–60, 2005.

[34] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *J. of Artificial Intelligence Research*, 36:165–228, 2009.

[35] M. Ortiz, S. Rudolph, and M. Simkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In Lin et al. [28], pages 269–279.

[36] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview.* W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-overview/.

[37] P.F. Patel-Schneider and B. Motik, editors. *OWL 2 Web Ontology Language: Mapping to RDF Graphs.* W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-mapping-to-rdf/.

[38] I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *J. of Logic, Language and Information*, 14:369–395, 2005.

[39] S. Rudolph, M. Krötzsch, and P. Hitzler. Cheap Boolean role constructors for description logics. In S. Hölldobler et al., editors, *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*, volume 5293 of *LNAI*, pages 362–374. Springer, 2008.