Nominal Schemas for Integrating Rules and Description Logics

Markus Krötzsch[1], Frederick Maier[2], Adila A. Krisnadhi[2], and Pascal Hitzler[2]

[1] University of Oxford
markus.kroetzsch@comlab.ox.ac.uk
[2] Kno.e.sis Center, Wright State University
{fred,adila,pascal}@knoesis.org

Abstract. We propose an extension of $\mathcal{SROIQ}$ with *nominal schemas* which can be used like "variable nominal concepts" within axioms. This feature allows us to express arbitrary *DL-safe rules* in description logic syntax. We show that adding nominal schemas to $\mathcal{SROIQ}$ does not increase its worst-case reasoning complexity, and we identify a family of tractable DLs $\mathcal{SROELV}_n$ that allow for restricted use of nominal schemas.

1 Introduction

A significant body of work has developed investigating the integration of description logics (DLs) 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. For instance, $\mathcal{SROIQ}$ role inclusion axioms (RIAs) can be viewed as a type of rule. By combining RIAs with local reflexivity (Self) and the universal role U, many rules with a *tree-shaped body* can be expressed indirectly [10]. The restriction to tree-shaped rules ensures decidability, but it also excludes many rules. An example is the following rule that defines a concept C of children whose parents are married:

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

On the other hand, there are approaches of a *hybrid* nature, in the sense that both DL axioms and rules are syntactically allowed, and a combined formal semantics defines how the hybrid language is to be understood. Unfortunately, such a combination often leads to undecidability. This is the case for the *Semantic Web Rule Language* SWRL [5,6], 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) [1,2]. 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 [5,14].

A generalization of DL-safe rules, based on *DL-safe variables*, has been introduced [11] 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 [9].

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.

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

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

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

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

$$\exists\text{hasParent}.\{z\} \sqcap \exists\text{hasParent}.\exists\text{married}.\{z\} \sqsubseteq C. \tag{4}$$

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 $\mathcal{SROIQ}$.
2. We establish the worst-case complexity of reasoning in $\mathcal{SROIQV}$ to be N2ExpTime-complete, and thus not harder than for $\mathcal{SROIQ}$.

3. We define $\mathcal{SROELV}_n$ ($n \geq 0$) as a new family of DLs with nominal schemas for which reasoning is possible in polynomial time.

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 $\mathcal{SROELV}_n$ 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 $\mathcal{SROIQV}$ may seem encouraging, we believe that the tractable ontology languages $\mathcal{SROELV}_n$ are the most promising candidates for implementations.

This paper is a condensed presentation of the main results of [13] where we develop all results for the slightly more general case of DLs with Boolean role constructors and concept products [18]. Moreover, [13] also explains how DL-safe rules (and hence DLs with nominal schemas) can be used to express OWL RL ontologies, and provides an extended discussion of related approaches which include description graphs, existential rules and tuple-generating dependencies (TGDs) in Datalog, and DL Rules.

In this paper, we introduce the syntax and semantics of nominal schemas for $\mathcal{SROIQV}$, and establish the worst-case complexity of reasoning in Section 2. The DLs $\mathcal{SROELV}_n$ are introduced in Section 3, and their tractability is established in Section 4. In Section 5 we show how DL-safe rules can be expressed with nominal schemas, and Section 6 concludes.

2 Nominal Schemas in $\mathcal{SROIQ}$

We start by introducing *nominal schemas* as an extension of existing description logics. We first generally introduce the feature for $\mathcal{SROIQ}$ to obtain the very expressive DL $\mathcal{SROIQV}$.

A signature of $\mathcal{SROIQV}$ is a tuple $\Sigma = \langle N_I, N_C, N_R, N_V \rangle$ of mutually disjoint sets of *individual names*, *concept names*, *role names*, and *variables*. Variables can be used like individuals in nominal expressions, and concept expressions of $\mathcal{SROIQV}$ are thus defined as follows:

$$\begin{aligned}\mathbf{C} ::= \top \mid \bot \mid N_C \mid \{N_I\} \mid \{N_V\} \mid \neg\mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \\ \exists\mathbf{R}.\mathbf{C} \mid \forall\mathbf{R}.\mathbf{C} \mid \exists\mathbf{S}.\mathsf{Self} \mid \leqslant k\,\mathbf{S}.\mathbf{C} \mid \geqslant k\,\mathbf{S}.\mathbf{C}\end{aligned}$$

where k is a natural number, and $\mathbf{R}$ ($\mathbf{S}$) is a (simple) $\mathcal{SROIQ}$ role as usual. We use U to denote the universal role. The common axiom types are defined as usual, but with the extended set of concept expressions. Herein, we restrict our attention to $\mathcal{SROIQV}$ knowledge bases with regular RBoxes, which are defined as in $\mathcal{SROIQ}$.

Axiom (4) above is an example of a $\mathcal{SROIQV}$ TBox axiom, 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. We note that it would also be straightforward to introduce nominal schemas into the normative RDF syntax for OWL 2 [16]. 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). Attaching the semantics of nominal schemas to "reserved" variable URIs would allow the reuse of existing tools for representation, manipulation, parsing, and serialization.

The semantics of $\mathcal{SROIQV}$ is defined by interpreting variables as placeholders for *named* individuals, i.e. elements of the interpretation domain that are represented by individual names in N_I. This can be accomplished by using a suitably restricted form of variable assignment. Equivalently, one can eliminate nominal schemas by replacing them with the (finitely many) nominals that they can represent, and apply the standard $\mathcal{SROIQ}$ semantics to the result [13].

Definition 1. *The* grounding $\mathsf{ground}(\alpha)$ *of a $\mathcal{SROIQV}$ axiom α 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 $\mathcal{SROIQV}$ knowledge base KB, we define* $\mathsf{ground}(KB) \coloneqq \bigcup_{\alpha \in KB} \mathsf{ground}(\alpha)$.

A DL interpretation $\mathcal{I}$ is a model *of a $\mathcal{SROIQV}$ axiom α, written $\mathcal{I} \models \alpha$, if and only if $\mathcal{I}$ is a model of the knowledge base* $\mathsf{ground}(\alpha)$. *Satisfaction and entailment of $\mathcal{SROIQV}$ axioms and knowledge bases is defined as usual.*

Note that grounding does not affect the structural restrictions of simplicity and regularity. Definition 1 provides a direct approach for reasoning with $\mathcal{SROIQV}$, though not necessarily a very practical one given that each $\mathcal{SROIQV}$ axiom represents an exponential number of $\mathcal{SROIQ}$ axioms obtained by grounding. However, this observation already yields an upper bound for the complexity of reasoning with $\mathcal{SROIQV}$ that is exponentially larger than that of $\mathcal{SROIQ}$, i.e. N3ExpTime. In the remainder of this section, we 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 $\mathcal{SROIQ}$ [8].

We first recall the complexity proof of [8] which is 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 [17]. 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 $\mathcal{C}^2$.

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 $R_{(i)}, S_1, S_2 \in N_R$ with S_1, S_2 simple, and $C \equiv D$ is short for $\{C \sqsubseteq D, D \sqsubseteq C\}$. This normalization can be done in linear time; see [8] 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 [3] using nondeterministic finite automata (NFA) to represent RIAs that entail non-simple roles. Suitable NFA for $\mathcal{SROIQ}$ were defined in [4,7]. 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 [4] and the accompanying technical report.

Fact 1 *Consider a $\mathcal{SROIQ}$ knowledge base KB. For each (possibly inverse) non-simple role $R \in \mathbf{R}$, there is an NFA $\mathcal{A}_R$ over the alphabet N_R such that for every model $\mathcal{I}$ of KB, and for every word $S_1 \dots S_n$ accepted by $\mathcal{A}_R$:*

$$\text{If } \langle \delta_i, \delta_{i+1} \rangle \in S_i^{\mathcal{I}} \text{ for all } i = 1, \dots, n, \text{ then } \langle \delta_1, \delta_{n+1} \rangle \in R^{\mathcal{I}}.$$

Moreover, let $\prec$ denote a strict linear order that witnesses regularity of KB as required in [4]. For each non-simple $R \in N_R$, the number of states of $\mathcal{A}_R$ is bounded exponentially in the depth *of KB that is defined as:*

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

It suffices to construct the respective NFA for non-simple roles. 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 $R_1 \circ \dots \circ R_n \sqsubseteq R$ 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 1. *The problem of deciding the satisfiability of $\mathcal{SROIQV}$ knowledge bases is* N2ExpTime*-complete.*

Proof. The depth of KB is only affected by RBox axioms. 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 $\mathcal{SROIQ}$, one can apply the transformation steps (1)–(3). This yields a $\mathcal{C}^2$ theory T that is equisatisfiable to $\mathsf{ground}(KB)$ [8] and thus to KB. 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 [17], thus deciding satisfiability of KB in N2ExpTime.

Hardness follows since $\mathcal{SROIQV}$ includes $\mathcal{SROIQ}$, for which deciding satisfiability is N2ExpTime-hard [8]. □

3 A Tractable Fragment

The result that reasoning in $\mathcal{SROIQV}$ has the same worst-case complexity as $\mathcal{SROIQ}$ is encouraging, yet we are far from a practical reasoning procedure for this DL. In particular, Theorem 1 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. In [13], we also further discuss the relationship to the tractable profiles of OWL 2.

Concretely, we define DLs $\mathcal{SROELV}_n$ 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 $\mathcal{SROEL}(\times)$, introduced as an extension of OWL EL [12]. In essence, $\mathcal{SROEL}(\times)$ is an extension of $\mathcal{EL}$ with $\top$, $\bot$, nominals, complex role inclusions, Self, and concept products [18]. Here, we only need the special concept product $\top \times \top$, denoted as the universal role U. In particular, we also omit range restrictions $R \sqsubseteq \top \times C$ since they do not contribute to our treatment.

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 [11] 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. 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 [11], 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. For instance, 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 2. *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 $\mathcal{SROIQV}$ 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 3. *Let $n \geq 0$. A $\mathcal{SROELV}_n$* concept *is a $\mathcal{SROIQV}$ concept that may contain $\top$, $\bot$, $\sqcap$, $\exists$, Self, the universal role, nominals and nominal schemas, but which does not contain $\sqcup$, $\neg$, $\forall$, ${\leqslant}k$, ${\geqslant}k$, or inverse roles.*

A $\mathcal{SROELV}_n$ TBox axiom *is a $\mathcal{SROIQV}$ TBox axiom α that uses $\mathcal{SROELV}_n$ concepts only, and where at most n nominal schemas are not safe for α. An* RBox axiom *of $\mathcal{SROELV}_n$ is an RBox axiom of $\mathcal{SROIQV}$ that does not contain inverse roles. A $\mathcal{SROELV}_n$* knowledge base *is a $\mathcal{SROIQV}$ knowledge base that contains only $\mathcal{SROELV}_n$ axioms.*

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. When viewing nominal schemas as a way of augmenting DL expressivity in existing applications, it seems plausible that this number remains small. Axiom (4) is an example of a $\mathcal{SROELV}_1$ axiom.

4 Reasoning with $\mathcal{SROELV}_n$

If n is constant, the problem of checking satisfiability in $\mathcal{SROELV}_n$ 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 $\mathcal{SROEL}(\times)$ [12].

Let KB be a $\mathcal{SROELV}_n$ knowledge base. We define a $\mathcal{SROEL}(\times)$ 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 2. *Given a $\mathcal{SROELV}_n$ 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. □

As shown in [13], a $\mathcal{SROELV}_n$ knowledge base *KB* is satisfiable if and only if $\mathsf{ground}^+(KB)$ is satisfiable. 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 $\mathcal{SROEL}(\times)$ from [12], we thus obtain the following result. Hardness for P follows from the hardness of $\mathcal{SROEL}(\times)$.

Theorem 3. *If KB is a $\mathcal{SROELV}_n$ knowledge base of size s, satisfiability of KB can be reduced to instance retrieval w.r.t. a set of Datalog rules of size proportional to s^n and at most 4 variables per rule. If n is constant, the problem is* P*-complete.*

5 DL-Safe Rules

An interesting feature of nominal schemas is that they can be used to express arbitrary DL-safe rules [14]. 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. 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 $N_I \cup N_V$.

Definition 4. *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 conjunction 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. *A 0-variable rule is a* ground rule. *The* grounding $\mathsf{ground}(B \to H)$ *of a DL-safe rule $B \to H$ is the set of all rules that can be obtained by uniformly replacing variables in $B \to H$ with individual names of the signature.*

An interpretation $\mathcal{I}$ satisfies *a ground DL-safe rule $B \to H$, written $\mathcal{I} \models B \to H$, if either $\mathcal{I} \models H$ or $\mathcal{I} \not\models B$. An interpretation $\mathcal{I}$* satisfies *a DL-safe rule $B \to H$ if it satisfies all rules in $\mathsf{ground}(B \to H)$. A set of rules is satisfied if all of its elements are. Models, satisfiability, and entailment are defined as usual.*

Since DL-safe rules use the same models as $\mathcal{SROIQV}$, it is easy 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.

Definition 5. *A syntactic transformation dl from atoms and DL-safe rules to $\mathcal{SROIQV}$ concepts and TBox axioms is defined as follows. For a unary atom $A(t)$ and binary atom $R(s,t)$, we set*

$$\mathsf{dl}(A(t)) \coloneqq \exists U.(\{t\} \sqcap A) \qquad \text{and} \qquad \mathsf{dl}(R(s,t)) \coloneqq \exists U.(\{s\} \sqcap \exists R.\{t\}).$$

Given a DL-safe rule $B \to H$, we set $\mathsf{dl}(B \to H) \coloneqq \bigsqcap_{F \in B} \mathsf{dl}(F) \sqsubseteq \mathsf{dl}(H)$, and for a set of DL-safe rules RB we define $\mathsf{dl}(RB) \coloneqq \bigcup_{B \to H \in RB} \mathsf{dl}(B \to H)$.

The function dl transforms rules into $\mathcal{SROELV}_n$ 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 or regularity are violated. In consequence, $\mathsf{dl}(RB)$ is a $\mathcal{SROELV}_n$ knowledge base if RB is a set of n-variable rules. The following result of [13] is not hard to show:

Theorem 4. *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.*

Importantly, this result confirms that nominal schemas are powerful enough to express arbitrary DL-safe rules. The use of nominal schemas, however, in $\mathcal{SROIQV}$ is more general than the extension of $\mathcal{SROIQ}$ with DL-safe rules, since the latter correspond to a special form of $\mathcal{SROIQV}$ axioms only. Combining Theorem 3 with the observation that $\mathsf{dl}(RB)$ is linear in the size of RB, we can state the following:

Theorem 5. *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 $\mathcal{SROELV}_n$ knowledge base, is* P*-complete.*

6 Conclusions and Future Work

We have introduced nominal schemas as an extension to description logics, giving DLs sufficient expressivity to incorporate rule-based modeling. In particular, the use of nominal schemas supports the integration of DL-safe rules into DL knowledge bases. 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 $\mathcal{SROIQ}$, and that versatile tractable sublanguages exist. Whether and how these theoretical results can be put into efficient reasoning algorithms is an open research question. Two different approaches to addressing this problem appear viable. On the one hand, nominal schemas could be implemented by modifying/extending existing $\mathcal{SROIQ}$ implementations that have good support for nominals, such as the OWL 2 reasoner *HermiT* [15]. 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 description logics could be implemented using rule-based procedures as proposed for $\mathcal{SROEL}$ [12]. 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. Specifically, the approach in [12] introduces new constant symbols for eliminating existentials, and DL-safe variables must not be 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 opportunity for enhancing the expressive power of ontologies without giving up on established tools, formats, 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.

References

1. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D. (eds.): RIF Core Dialect. W3C Recommendation (22 June 2010), available at http://www.w3.org/TR/rif-core/

2. de Bruijn, J.: RIF RDF and OWL Compatibility. W3C Recommendation (22 June 2010), available at http://www.w3.org/TR/rif-rdf-owl/
3. Demri, S., Nivelle, H.: Deciding regular grammar logics with converse through first-order logic. J. of Logic, Language and Information 14(3), 289–329 (2005)
4. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
5. Horrocks, I., Patel-Schneider, P., Bechhofer, S., Tsarkov, D.: OWL Rules: A proposal and prototype implementation. J. of Web Semantics 3(1), 23–40 (2005)
6. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language. W3C Member Submission (21 May 2004), see http://www.w3.org/Submission/SWRL/
7. Horrocks, I., Sattler, U.: Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. Artificial Intelligence 160(1), 79–104 (2004)
8. Kazakov, Y.: $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In: Brewka, G., Lang, J. (eds.) Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08). pp. 274–284. AAAI Press (2008)
9. Krötzsch, M.: Description Logic Rules, Studies on the Semantic Web, vol. 008. IOS Press/AKA (2010)
10. Krötzsch, M., Rudolph, S., Hitzler, P.: Description logic rules. In: Ghallab, M., et al. (eds.) Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008. pp. 80–84. IOS Press (2008)
11. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable rules for OWL 2. In: Sheth, A., et al. (eds.) Proceedings of the 7th International Semantic Web Conference (ISWC-08). Lecture Notes in Computer Science, vol. 5318, pp. 649–664. Springer (2008)
12. Krötzsch, M.: Efficient rule-based inferencing for OWL EL. In: Walsh, T. (ed.) Proc. 22nd Int. Conf. on Artificial Intelligence (IJCAI'11). IJCAI (2011), to appear
13. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Proc. 20th Int. Conf. on World Wide Web (WWW'11). pp. 645–654. ACM (2011)
14. Motik, B., Sattler, U., Studer, R.: Query answering for OWL DL with rules. J. of Web Semantics 3(1), 41–60 (2005)
15. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. J. of Artificial Intelligence Research 36(1), 165–228 (2009)
16. Patel-Schneider, P., Motik, B. (eds.): 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/
17. Pratt-Hartmann, I.: Complexity of the two-variable fragment with counting quantifiers. J. of Logic, Language and Information 14, 369–395 (2005)
18. Rudolph, S., Krötzsch, M., Hitzler, P.: Cheap Boolean role constructors for description logics. In: Hölldobler, S., et al. (eds.) Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08). LNAI, vol. 5293, pp. 362–374. Springer (2008)