1-1-2010

Nominal Schemas for Integrating Rules and Ontologies

Frederick Maier

Adila A. Krisnadhi

Pascal Hitzler
pascal.hitzler@wright.edu

Nominal Schemas for Integrating Rules and Ontologies

Technical Report, Kno.e.sis Center, Wright State University, 2010

Frederick Maier, Adila A. Krisnadhi, Pascal Hitzler
Kno.e.sis Center, Wright State University, Dayton, OH 45435, U.S.A.
{fred,adila,pascal}@knoesis.org

This is a very early report on the topic. It has subsequently been revised and substantially extended, leading to the publication [15], available from
`http://korrekt.org/page/A_Better_Uncle_For_OWL`

ABSTRACT

We propose a description-logic style extension of OWL DL, which includes DL-safe variable SWRL and seamlessly integrates datalog rules. Our language also sports a tractable fragment, which we call ELP 2, covering OWL EL, OWL RL, most of OWL QL, and variable restricted datalog.

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 quest for a satisfactory integration of Ontologies and Rules for the Semantic Web is still ongoing [7]. In the aftermath of the 2004 W3C Recommendation for the Web Ontology Language OWL [17], the problem was discussed, in a nutshell, by referring to the *uncle* rule, expressible in datalog as

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

which could not be modeled at all in that 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 [21], the 2009 revision of the OWL W3C Recommendation, in fact indirectly incorporates datalog rules of a certain form. This covers the *uncle* rule mentioned above, and, more generally, many rules with a *tree-shaped body* [13]. Decidability is nevertheless retained. However, many rules, such as

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

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

On the other hand, there are approaches that are 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.[1] This is the case for the Semantic Web Rule Language SWRL, [9, 8] which is the most straightforward rule extension of OWL, and which implicitly covers all of first-order predicate logic.[2] In order to retain decidability, the most prominently discussed idea[3] is to restrict the applicability of rules to *known individuals* in the knowledge base, i.e. to explicitly mentioned logical constants. Rules that are understood in this sense are called *DL-safe* [8, 20], and the combination of OWL DL and DL-safe SWRL is indeed decidable.[4]

[1]In the case of nonmonotonic semantics for rules, this may lead to languages which are not even semi-decidable, because some of these rule languages are not semi-decidable. We will not deal with nonmonotonic semantics very much in this paper, but we will briefly discuss them in Section 5.

[2]The combination of RIF rule bases and OWL DL ontologies is also undecidable [4], even for RIF Core [2].

[3]Other approaches, such as [5], achieve decidability by limiting the semantic interaction between the DL part and the rules part of the language. I.e., they provide a *loose* integration. While this is useful in some contexts, it seems evident that a tight integration would be preferred in general. A conceptually different series of work based on the idea of extending datalog with existential quantifiers uses syntactic restrictions to achieve decidability and favorable computational complexities [3].

[4]Hybrid MKNF [10, 19] is also essentially based on the same idea, although the semantics is spelled out differently.

A generalization of DL-safe rules has been introduced in [14],[5] in the form of *DL-safe variables* as part of the definition of the tractable ELP language. Rather than restricting *all* variables in a (DL-safe) rule such that they can bind only to known individuals, DL-safe variables endow the ontology engineer with the means to explicitly specify the variables to be treated this way.

In this paper, we expand on the above idea and improve on it in several ways. The key technical innovation is the introduction of *nominal schemas* as new elements of the DL syntax. While the semantic intuition behind nominal schemas is the same as that behind DL-safe variables, the key 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 syntax paradigm what has hitherto only been achieved by hybrid approaches.

To give an initial example, consider again the rule (1) extended by the axioms $(\exists \text{married}.\exists \text{parentOf}.\{\text{mary}\})(\text{john})$ and $\text{parentOf}(\text{john}, \text{mary})$. First-order logic semantics (i.e., SWRL semantics) would yield, that Mary belongs to the class C. Interpreting the rule (1) as DL-safe, however, does not allow this conclusion. If we impose that the variable z in rule (1) is DL-safe while x and y are not, then the desired conclusion follows, but the rule (1) interpreted as such cannot be easily expressed in DL syntax,[6] and so the knowledge base remains hybrid.

In contrast, using nominal schemas, rule (1) with z DL-safe can be expressed as

$$\exists \text{parentOf}.\{z\} \sqcap \exists \text{married}.\exists \text{parentOf}.\{z\} \sqsubseteq \exists U.(\{z\} \sqcap C), \quad (2)$$

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

The main contribution of this paper is the introduction of nominal schemas into DL syntax and the corresponding definition of a DL called $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$, which extends OWL DL and possesses the following very pleasing features: (1) It is a description logic (and thus can be defined concisely without reference to rules). (2) It is decidable (worst-case computational complexity of reasoning is in N3ExpTime). (3) It encompasses DL-safe SWRL and thus DL-safe datalog. (4) It sports a tractable (polynomial time) fragment ELP 2, which contains OWL EL (i.e. $\mathcal{EL}^{++}$), OWL RL (i.e. DLP 2), most of OWL QL (i.e. DL-Lite$_\mathcal{R}$; inverse roles are not completely covered), DL-safe datalog (and thus DL-safe RIF-Core), and 5-variable restricted (non DL-safe) datalog.

The work presented herein is heavily inspired by the work on ELP [14], which also explains why we call the tractable sublanguage ELP 2. We also incorporate the rather neat alternative definition of ELP given in [11] together with notational choices from [12]. The key difference between ELP 2 and ELP as presented in these publications is that ELP is defined as a *hybrid* language composed of DL axioms and rule axioms, and determining whether a knowledge base is in ELP can, in general, only be made by performing syntactic transformations and subsequently checking whether some standing conditions are met. In contrast, ELP 2 (and, in fact, $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$) can be defined without resorting to rules or such transformations. Indeed, we urge the reader to verify, from the definition given in [14], that rule (1) (with z DL-safe) is not in ELP, which requires the checking of several rather involved conditions. In contrast, it follows easily from our definitions (given later) that axiom (2) is in ELP 2.

The paper is structured as follows. In Section 2 we introduce the syntax and semantics of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ and give a complexity analysis. We introduce ELP 2 in Section 3 and show its polynomiality and that it encompasses OWL EL. In Section 4 we clarify the relation of datalog and SWRL to $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ and ELP 2, and that the latter contains OWL RL and most of OWL QL. Section 5 contains conclusions and a discussion of future research.

The paper is relatively self-contained, but due to the depth of the discussion, we assume some knowledge by the reader of different notations for logics around OWL and rules, as well as background on DLs and their relationships to OWL. This background material can be found, e.g., in [6].

2. NOMINAL SCHEMAS FOR OWL

In [23] it was shown that the description logic $\mathcal{SROIQ}$, the logic underlying OWL 2 DL, can be extended with boolean constructors ($\neg$, $\sqcap$, $\sqcup$) on simple roles without thereby increasing the complexity of the standard reasoning tasks. The resulting logic was called $\mathcal{SROIQB}_s$. Concept products $C \times D$ and a universal role U (which can in turn be expressed as $\top \times \top$, where $\top$ is the universal concept) were also briefly discussed: while not formally part of $\mathcal{SROIQB}_s$, expressions involving both can be reduced to those using only the basic role constructors. A logic (which we'll call $\mathcal{SROIQB}_s(\times)$) involving all features was described in [11]. Again, including all of these features does not increase the worst-case complexity of reasoning.

Here, we extend $\mathcal{SROIQB}_s$ with nominal schemas, forming the logic $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$. We mainly follow and extend the presentation found in [11] and give a complete account of the syntax and semantics of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$.

A *signature* Σ is a tuple $\langle N_I, N_C, N_R, N_V \rangle$, 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 of terms $\mathbf{T}$ of Σ is $N_I \cup N_V$. 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*. We also specify a *universal role* U, not in N_R. For each $a \in N_I$, $\{a\}$ is a *nominal*, and for each $v \in N_V$, $\{v\}$ is a *nominal schema*. O denotes the set of nominals, and V denotes the set of nominal schemas.

From now on, we assume a given signature Σ and so omit further references to it.

[5] See also [11].

[6] Arguments like this are always difficult, since languages of the same computational complexity are always translatable into each other. As usual, we appeal to an intuitively "simple" notion of "expressibility" in this context.

A *simple atomic role* is any element of $N_R^s \cup \{R^- \mid R \in N_R^s\}$. A *non-simple atomic role* is any element of $N_R^n \cup \{R^- \mid R \in N_R^n\}$. The set $\mathbf{R}_n$ of *non-simple role expressions* is the smallest set that contains every non-simple atomic role and every *concept product* $C \times D$, where C, D are nominal-schema free elements of the set $\mathbf{C}$, as defined in Definition 1. The set $\mathbf{R}_s$ of *simple role expressions* is the smallest set that contains U, every simple atomic role, all expressions of the form $\neg R$, $R \sqcap S$ and $R \sqcup S$ for $R, S \in \mathbf{R}_s$, and every concept product. The set $\mathbf{R}$ of *role expressions* is the union of $\mathbf{R}_s$ and $\mathbf{R}_n$. The set $\mathbf{R}_{C\times D} \subseteq \mathbf{R}$ is the set of all concept products.

We define Inv as a function from roles to roles, as follows: For each $R, S \in N_R$, define $\mathrm{Inv}(R) = R^-$, $\mathrm{Inv}(R^-) = R$, $\mathrm{Inv}(R \sqcap S) = \mathrm{Inv}(R) \sqcap \mathrm{Inv}(S)$, $\mathrm{Inv}(R \sqcup S) = \mathrm{Inv}(R) \sqcup \mathrm{Inv}(S)$, $\mathrm{Inv}(\neg R) = \neg \mathrm{Inv}(R)$; $\mathrm{Inv}(U) = U$; for each concept product $C \times D$, $\mathrm{Inv}(C \times D) = D \times C$.

Definition 1. The set $\mathbf{C}$ of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ *concept descriptions* (or, *concepts*) is the smallest set such that

1. $\{\top, \bot\} \cup N_C \cup O \cup V \subseteq \mathbf{C}$;
2. if $C, D \in \mathbf{C}$ and k is an nonnegative integer, then the following are also $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ concepts: $\neg C$, $C \sqcap D$, $C \sqcup D$, $\exists R.C$, $\forall R.C$, $\exists S.\mathsf{Self}$, $\leq k\, S.C$ and $\geq k\, S.C$, where $R \in \mathbf{R}$ and $S \in \mathbf{R}_s$.

A concept description C is *nominal-schema free* if no occurrence of any element of V appears in C.

A generalized *role inclusion axiom (RIA)* is any statement of the form $S_1 \circ \cdots \circ S_k \sqsubseteq R$ where R and each S_i are in $\mathbf{R}$ and if $R \notin \mathbf{R}_n$, then $k = 1$ and $S_1 \in \mathbf{R}_s$. An *RBox axiom* is either an RIA, or a statement of the form $\mathrm{Ref}(R)$ (reflexive role), $\mathrm{Asy}(R)$ (asymmetric role), $\mathrm{Dis}(R_1, R_2)$ (role disjointness), where $R_{(i)} \in \mathbf{R}$. A *TBox axiom* (*general concept inclusion axiom, GCI*) is any expression $C \sqsubseteq D$, where $C, D \in \mathbf{C}$. A TBox axiom is *nominal-schema free* if no nominal schema occurs in it. An *ABox axiom* is any expression of the form $C(a)$, $R(a, b)$, $\neg S(a, b)$, and $a \not\approx b$ where C is a nominal-schema free concept description in $\mathbf{C}$, $R \in \mathbf{R}$, $S \in \mathbf{R}_s \cup \mathbf{R}_{C\times D}$, and $a, b \in N_I$. A *knowledge base* is a set of RBox, TBox, and ABox axioms. A knowledge base is *nominal-schema free* if all of its TBox axioms are so.

An RBox is *regular* if there is a strict partial order $\prec$ on $\mathbf{R}$ such that

- if $R \notin \{S, \mathrm{Inv}(S)\}$, then $S \prec R$ iff $\mathrm{Inv}(S) \prec R$; and
- every RIA is of one of the forms $R \circ R \sqsubseteq R$, $\mathrm{Inv}(R) \sqsubseteq R$, $R \circ S_1 \circ \ldots S_k \sqsubseteq R$, $S_1 \circ \cdots \circ S_k \circ R \sqsubseteq R$, or $S_1 \circ \cdots \circ S_k \sqsubseteq R$, where R and each S_i is in $\mathbf{R}$, and $S_i \prec R$ for each S_i.

A knowledge base is a $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ knowledge base iff its RBox is regular, and for each statement Irr(S), $\mathrm{Asy}(S)$, $\mathrm{Dis}(S, T)$ in the RBox, $S, T \in \mathbf{N}_s$.

Axiom (2) from the previous section is an example of a $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ axiom. The $\{z\}$ appearing there is a nominal schema. Intuitively, each nominal schema appearing in an axiom is universally quantified, and in any interpretation, ranges only over named individuals in the domain of discourse. Importantly, the semantics described below assumes that axioms do not share nominals schemas; i.e., if a schema appears in one axiom, then it appears in no other. This simplifies the presentation somewhat.

An *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ for a signature Σ consists of a domain of discourse $\Delta^\mathcal{I} \neq \emptyset$ and a function $\cdot^\mathcal{I}$ which maps N_C, N_R, N_I, $\top$, and $\bot$ to elements, sets, and relations of $\Delta^\mathcal{I}$ as shown in Table 1. A *variable assignment* Z for an interpretation $\mathcal{I}$ is a function from N_V to $\Delta^\mathcal{I}$ such that for each $v \in N_V$, $Z(v) = a^\mathcal{I}$ for some $a \in N_I$. For any interpretation $\mathcal{I}$, assignment Z, and $C_{(i)} \in \mathbf{C}$, $R_{(i)} \in N_R$, $t_{(i)} \in \mathbf{T}$, the function $\cdot^{\mathcal{I},Z}$ is defined as shown in Table 1. For A a $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ axiom, and S a set of such axioms, $\mathcal{I}$ and Z *satisfy* A, written $\mathcal{I}, Z \models A$, *iff* the corresponding condition shown in Table 1 holds. $\mathcal{I}$ and Z *satisfy* S ($\mathcal{I}, Z \models S$) *iff* $\mathcal{I}, Z \models A$ for each A in S. $\mathcal{I}$ is a model of S if ($\mathcal{I}, Z \models S$) for each assignment Z for $\mathcal{I}$.

The logic $\mathcal{SROIQB}_s$ is obtained from $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ by disallowing nominal schemas; concept products are already covered by $\mathcal{SROIQB}_s$ since they can be simulated using role negations [23]. The logic $\mathcal{SROIQ}$ is in turn obtained from $\mathcal{SROIQB}_s$ by disallowing Boolean role constructors. In Section 4, we show that $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ is also expressive enough to encompass DL-safe datalog (and thus DL-safe SWRL and DL-safe RIF-Core).

We note that it is very straightforward to introduce nominal schemas into the normative RDF syntax for OWL 2 [22]. 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).

2.1 Complexity of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$

As shown here, every $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ knowledge base can be converted into an equisatisfiable $\mathcal{SROIQB}_s$ knowledge base simply by grounding the nominal schemas—i.e., by uniformly replacing nominal schemas with elements of O. Since reasoning in $\mathcal{SROIQB}_s$ is N2ExpTime-complete [11, 23], it follows that it is decidable in $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$.

Let C be a concept containing $m \geq 1$ nominal schemas $\{v_1\}, \{v_2\}, \ldots, \{v_m\}$ and $(a_1, \ldots, a_m)$ be a tuple of m individual names. A *grounding of* C *w.r.t.* $(a_1, \ldots, a_m)$ (denoted using $C\theta$, where $\theta = [v_1/a_1, \ldots, v_m/a_m]$) is the concept that is obtained from C by substituting all occurrences of nominal schema $\{v_i\}$ with nominal $\{a_i\}$ for every $i = 1, \ldots, m$. If A is an axiom, then $A\theta$ is an axiom obtained from A by an analogous substitution. If A is free of nominal schemas, then $A = A\theta$. A' is the set of all groundings of A constructible from N_I. If S is a set of axioms, then let S' be $\bigcup\{A' | A \in S\}$. Observe that S' is expressible in $\mathcal{SROIQB}_s$.

Grounding provides an alternative way of looking at nominal schemas, namely as a type of syntactic sugar or macro for the set of corresponding groundings. In general, however, there should be smarter algorithmizations than through a naive grounding, a matter to which we return in Section 3.

Table 1: Semantics of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$.

atomic concept	A	$A^{\mathcal{I},Z} = A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	V	$V^{\mathcal{I},Z} = V^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual	a	$a^{\mathcal{I},Z} = a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
variable	v	$v^{\mathcal{I},Z} = Z(v)$
top	$\top$	$\top^{\mathcal{I},Z} = \top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
bottom	$\bot$	$\bot^{\mathcal{I},Z} = \bot^{\mathcal{I}} = \varnothing$
nominal (schema)	$\{t\}$	$\{t\}^{\mathcal{I},Z} = \{t^{\mathcal{I},Z}\}$
existential restriction	$\exists R.C$	$(\exists R.C)^{\mathcal{I},Z} = \{x \mid (\exists y)[(x,y) \in R^{\mathcal{I},Z} \wedge y \in C^{\mathcal{I},Z}]\}$
universal/value restriction	$\forall R.C$	$(\forall R.C)^{\mathcal{I},Z} = \{x \mid (\forall y)[(x,y) \in R^{\mathcal{I},Z} \to y \in C^{\mathcal{I},Z}]\}$
self restriction	$\exists R.\mathsf{Self}$	$(\exists R.\mathsf{Self})^{\mathcal{I},Z} = \{x \mid (x,x) \in R^{\mathcal{I},Z}\}$
concept complement	$\neg C$	$(\neg C)^{\mathcal{I},Z} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},Z}$
concept conjunction	$C_1 \sqcap C_2$	$(C_1 \sqcap C_2)^{\mathcal{I},Z} = C_1^{\mathcal{I},Z} \cap C_2^{\mathcal{I},Z}$
concept disjunction	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I},Z} = C_1^{\mathcal{I},Z} \cup C_2^{\mathcal{I},Z}$
qualified number restrictions	$\leq n\, R.C$	$(\leq n\, R.C)^{\mathcal{I},Z} = \{x \mid \lvert(x,y) \in R^{\mathcal{I},Z} \wedge y \in C^{\mathcal{I},Z}\rvert \leq n\}$
	$\geq n\, R.C$	$(\geq n\, R.C)^{\mathcal{I},Z} = \{x \mid \lvert(x,y) \in R^{\mathcal{I},Z} \wedge y \in C^{\mathcal{I},Z}\rvert \geq n\}$
universal role	U	$U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	R^-	$(R^-)^{\mathcal{I},Z} = \{(x,y) \mid (y,x) \in R^{\mathcal{I},Z}\}$
concept product	$C_1 \times C_2$	$(C_1 \times C_2)^{\mathcal{I},Z} = \{(x,y) \mid x \in C_1^{\mathcal{I},Z} \wedge y \in C_2^{\mathcal{I},Z}\}$
role negation	$\neg R$	$(\neg R)^{\mathcal{I},Z} = \{(x,y) \mid (x,y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \wedge (x,y) \notin R^{\mathcal{I},Z}\}$
role conjunction	$R_1 \sqcap R_2$	$(R_1 \sqcap R_2)^{\mathcal{I},Z} = R_1^{\mathcal{I},Z} \cap R_2^{\mathcal{I},Z}$
role disjunction	$R_1 \sqcup R_2$	$(R_1 \sqcup R_2)^{\mathcal{I},Z} = R_1^{\mathcal{I},Z} \cup R_2^{\mathcal{I},Z}$
role chain	$R_1 \circ \ldots \circ R_n$	$(R_1 \circ \ldots \circ R_n)^{\mathcal{I},Z} = R_1^{\mathcal{I},Z} \circ \ldots \circ R_n^{\mathcal{I},Z}$ where '$\circ$' denotes the usual composition of binary relations
unique name axiom	$t_1 \not\approx t_2$	$t_1^{\mathcal{I},Z} \neq t_2^{\mathcal{I},Z}$
concept instance	$C(t)$	$t^{\mathcal{I},Z} \in C^{\mathcal{I},Z}$
role instance	$R(t_1, t_2)$	$(t_1^{\mathcal{I},Z}, t_2^{\mathcal{I},Z}) \in R^{\mathcal{I},Z}$
	$\neg R(t_1, t_2)$	$(t_1^{\mathcal{I},Z}, t_2^{\mathcal{I},Z}) \notin R^{\mathcal{I},Z}$
concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I},Z} \subseteq C^{\mathcal{I},Z}$
role inclusion	$V \sqsubseteq W$	$V^{\mathcal{I},Z} \subseteq W^{\mathcal{I},Z}$
	$R_1 \circ \ldots \circ R_n \sqsubseteq R$	$(R_1 \circ \ldots \circ R_n)^{\mathcal{I},Z} \subseteq R^{\mathcal{I},Z}$
concept product (left)	$C \times D \sqsubseteq R$	$(C \times D)^{\mathcal{I},Z} \subseteq R^{\mathcal{I},Z}$
concept product (right)	$R \sqsubseteq C \times D$	$R^{\mathcal{I},Z} \sqsubseteq (C \times D)^{\mathcal{I},Z}$
role reflexivity	$\mathrm{Ref}(R)$	$R^{\mathcal{I},Z}$ is reflexive
role asymmetry	$\mathrm{Asy}(V)$	$V^{\mathcal{I},Z}$ is antisymmetric and irreflexive (i.e, asymmetric)
role disjointness	$\mathrm{Dis}(V, W)$	$V^{\mathcal{I},Z} \cap W^{\mathcal{I},Z} = \emptyset$

$\mathcal{I}$ an interpretation over Σ; Z an assignment for $\mathcal{I}$; $A \in N_C$; $C_{(i)}, D \in \mathbf{C}$; $V \in N_R$; $R_{(i)} \in \mathbf{R}$; $a \in N_I$; $v \in N_V$; $t_{(i)} \in \mathbf{T}$.

THEOREM 1. *$\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ is decidable.*

PROOF. Let S be a $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ knowledge base. We show S is satisfiable if and only if S' is.

(LR) Let $\mathcal{I}$ be a model of S, and suppose $A \in S'$. If $A \in S$, then obviously $\mathcal{I}$ is a model of A. So suppose $A \notin S$. Then A must be of the form $(C \sqsubseteq D)\theta$ where (i) $C \sqsubseteq D \in S$; (ii) there are m nominal schemas $\{v_1\}, \ldots, \{v_m\}$ in $C \sqsubseteq D$, $m \geq 1$; (iii) $\theta = [v_1/a_1, \ldots, v_m/a_m]$ for some individual names $a_1, \ldots, a_m$ in N_I. Let Z be an assignment for $\mathcal{I}$ such that $Z(v_i) = a_i$ for each $i = 1, \ldots, m$. Then $(C\theta)^{\mathcal{I},Z} = C^{\mathcal{I},Z}$ and $(D\theta)^{\mathcal{I},Z} = D^{\mathcal{I},Z}$. Because $\mathcal{I}$ is a model of S, $C^{\mathcal{I},Z} \subseteq D^{\mathcal{I},Z}$, and so $(C\theta)^{\mathcal{I},Z} \subseteq (D\theta)^{\mathcal{I},Z}$, i.e., $\mathcal{I}, Z \models A$. Since $C\theta$ and $D\theta$ are nominal-schema free, $(C\theta)^{\mathcal{I},W} = (C\theta)^{\mathcal{I},Z}$ and $(D\theta)^{\mathcal{I},W} = (D\theta)^{\mathcal{I},Z}$ for every assignment W for $\mathcal{I}$, and so $\mathcal{I}, W \models A$ for every assignment W for $\mathcal{I}$, i.e., $\mathcal{I}$ is a model of A. Generalizing on A, $\mathcal{I}$ is a model of S'.

(RL) Let $\mathcal{I}$ be a model of S', and suppose $A \in S$. If $A \in S'$, then $\mathcal{I}$ models A. So suppose $A \notin S'$. A must have the form $C \sqsubseteq D$ and must contain nominal schemas, $\{v_1\}$, $\ldots$, $\{v_m\}$ for some $m \geq 1$. Let Z be an assignment for $\mathcal{I}$. Then there exist individual names $a_1, \ldots, a_m$ in N_I such that $Z(v_i) = a_i$ for each $i = 1, \ldots, m$. The assignment Z guarantees that $C^{\mathcal{I},Z} = (C\theta)^{\mathcal{I},Z}$ and $D^{\mathcal{I},Z} = (D\theta)^{\mathcal{I},Z}$, where $\theta = [v_1/a_1, \ldots, v_m/a_m]$. $A\theta$ is a valid grounding of A and so appears in S'. Since $\mathcal{I}$ models S', $\mathcal{I}, Z \models A\theta$, and so $(C\theta)^{\mathcal{I},Z} \subseteq (D\theta)^{\mathcal{I},Z}$. I.e., $\mathcal{I}, Z \models A$. Generalizing on Z and A, we conclude that $\mathcal{I}$ is a model of S. $\square$

The naive grounding of a $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ knowledge base will yield an exponentially larger $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ knowledge base that is free of nominal schemas if the number of occurrences of different nominal schemas in the axioms of the knowledge base is unbounded. However, if we have a fixed global bound of the number of occurrences of different nominal schemas in each axiom, i.e., by imposing such a restriction as a part of the language, then the naive grounding only yields a polynomially larger knowledge base. Given that reasoning in $\mathcal{SROIQB}_s$ is N2ExpTime-complete, we obtain the following corollary.

THEOREM 2. *Reasoning for $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ can be done in N3ExpTime, and it is N2ExpTime-hard. Moreover, let $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\}_n)$ be the logic that is obtained from $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ by requiring that every axiom in the knowledge base may only have at most n different nominal schemas where n is a fixed positive integer. Then reasoning in $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\}_n)$ is N2ExpTime-complete.*

3. A TRACTABLE FRAGMENT

The introduction of nominal schemas does not only give rise to an integration of OWL and rules. It also provides us with the possibility of defining a tractable fragment, i.e., a sublanguage having polynomial time complexity. It is rather pleasing that this tractable fragment completely contains two of the three designated tractable profiles of OWL 2 [18] as well as variable restricted datalog. The third tractable OWL 2 profile is also almost completely contained.

This fragment is defined here, and subsequent sections show the desired containment relationships. The definition is very heavily inspired by the work on ELP [11, 14]. The essential differences have already been discussed in Section 1.

If n is an arbitrary but fixed positive integer, then the logic $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$, defined below, is a tractable fragment of $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$—the integer restricts the number of occurrences of nominal schemas. Every $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$ signature is a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ signature, and the sets N_C, N_R, N_I, etc., are defined as they were previously. Concepts and roles for $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ are defined below.

Definition 2. The set $\mathbf{C}$ of $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ concept descriptions is the smallest set such that

1. $\{\top, \bot\} \cup N_C \cup O \cup V \subseteq \mathbf{C}$,
2. if $C, D \in \mathbf{C}$, $R \in N_R$, then $C \sqcap D$, $\exists R.C$ and $\exists R.\text{Self}$ are in $\mathbf{C}$.

A $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ RBox axiom is any expression having one of the forms:

1. $R_1 \circ R_2 \sqsubseteq R$, where R and each $R_i \in N_R$.
2. $R_1 \sqcap R_2 \sqsubseteq R$, where R and each $R_i \in N_R$.
3. $C \times D \sqsubseteq R$, where $C, D \in \mathbf{C}$ and $R \in N_R$.
4. $R \sqsubseteq C \times D$, where $C, D \in \mathbf{C}$ and $R \in N_R$.

A $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ TBox axiom is any expression $C \sqsubseteq D$, where $C, D \in \mathbf{C}$. A $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ ABox axiom is any expression $C(a)$ or $R(a, b)$, where $C \in N_C$, $R \in N_R$, $a, b \in N_I$.

In order to arrive at a polynomial time fragment, we have to avoid the exponential blowup caused by grounding of nominal schema as in Theorem 2. The following notion allows us to do this in a non-trivial way.

Definition 3. An occurrence of a nominal schema $\{x\} \in V$ in an axiom A is called *safe*, if this occurrence is within a concept description D of A such that

1. D contains no other occurrence of any nominal schema,
2. the occurrence of D is within a subformula of the form $\{a\} \sqcap \exists R.D$, with $\{a\}$ a nominal.

We write $S(a, \exists R.D, x)$ to denote the subformula of the form $\{a\} \sqcap \exists R.D$ which contains a safe occurrence of a nominal schema $\{x\}$ as just described. We call $S(a, \exists R.D, x)$ a *safe environment* for this occurrence of the nominal schema $\{x\}$.

Definition 4. If KB is a knowledge base and R a role name, then $\text{ran}(R)$ is the set of concept descriptions $D \in \mathbf{C}$ such that $\{R_1 \sqsubseteq R_2, \ldots, R_n \sqsubseteq R_{n+1}\} \subseteq KB$, where $n \geq 1$, $R_1 = R$, and $R_{n+1} \sqsubseteq C \times D \in KB$.
A $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base is a knowledge base which satisfies the following restrictions:

1. if $\exists S.\text{Self}$ appears in KB, then $S \in N_R^s$;
2. if $R_1 \sqcap R_2$ appear in KB, then R_1, $R_2 \in N_R^s$;
3. if $R_1 \circ R_2 \sqsubseteq S$ is in KB, $\text{ran}(S) \subseteq \text{ran}(R_2)$;
4. if $R_1 \sqcap R_2 \sqsubseteq S$ is in KB, $\text{ran}(S) \subseteq \text{ran}(R_1) \cup \text{ran}(R_2)$;
5. no nominal schema appears in any role expression;
6. in any axiom $B \sqsubseteq C$, there are at most n nominal schemas appearing more than once in non-safe form, and all remaining nominal schemas appear only in B.[7]

Item 6 merits some further explanations, as this condition is central to the expressive power of $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ and also to the question why it is of polynomial time complexity. Item 6 in fact entails the following: There are at most n nominal schemas in $B \sqsubseteq C$ such that, after replacing all occurrences of these by nominals (through grounding), the resulting axiom $B' \sqsubseteq C'$ contains nominal schemas only in B'. Furthermore, for each nominal schema in B', all occurrences, except possibly one occurrence, are safe.

Another perspective on item 6 can be given as follows. In order to check whether an axiom $B \sqsubseteq C$ satisfies item 6, do the following. (1) Erase all safe occurrences in B (e.g., by substituting an arbitrary atomic class name), resulting in an axiom $B' \sqsubseteq C$. (2) Subsequently, erase all nominal schemas in B' which occur only once, resulting in an axiom $B'' \sqsubseteq C$. (3) $B \sqsubseteq C$ satisfies item 6 if, and only if, $B'' \sqsubseteq C$ contains at most n nominal schemas (possibly occurring several times).

It is very easily verified that axiom (2) can be expressed in $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_1)$. This also shows that the language $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_1)$ is *not* contained in ELP.

Definition 5. $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$ knowledge bases are also called *ELP 2 knowledge bases*

The choice for the number 5 as the global bound lies in the fact that OWL 2 RL can be embedded in the 5-variable fragment of datalog. We return to this topic in Section 4.2. Below, we show that ELP 2 contains OWL 2 EL.

THEOREM 3. *$\mathcal{SROEL}(\sqcap, \times)$ (and, thus, essentially OWL 2 EL) is a fragment of $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$.*

PROOF. Syntactically, every valid $\mathcal{SROEL}(\sqcap, \times)$ expression is a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$ expression. This can be seen by comparing the presentation here with that found in [12]. Similarly, variable assignments become irrelevant, and so the semantics for $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$ collapses to that of $\mathcal{SROEL}(\sqcap, \times)$. □

[7] This requirement can be relaxed somewhat. However the definition then becomes more awkward, which is why we decided to stick with the presented one. We will further comment on this below.

3.1 A polynomial algorithm

We show here that $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$, for any fixed n, is indeed a tractable logic.

Let K be a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base. We will transform K into a $\mathcal{SROEL}(\sqcap, \times)$ knowledge base K'. This will be done in three subsequent steps.

(1) We know, that in any axiom $A \in K$, there are at most n nominal schemas $\{x_1\}, \ldots, \{x_n\}$ which appear more than once in non-safe form. The first step of the transformation is obtained by inserting into K' all of the groundings of each axiom $A \in K$, as described in Section 2.1—however, the groundings are done only with respect to the nominal schemas $\{x_1\}, \ldots, \{x_n\}$; all other nominal schemas are untouched. The number of new axioms introduced this way is bounded from above by y^{n+1}, where y is the maximum of the number of axioms and the number of individuals in K. Hence, this grounding is polynomial.

We are now left with nominal schemas in axioms $B \sqsubseteq H$ which occur only in B and which occur at most once in non-safe form in B.[8]

(2) Introduce a new class name O. For all individuals b occurring in the knowledge base, add axiom $\{b\} \sqsubseteq O$ to K'. For any axiom of K' and any nominal schema $\{x\}$ occurring only once in this axiom, replace $\{x\}$ by O. Note that these nominal schemas can occur only on left-hand sides of axioms of the form $B \sqsubseteq H$. This transformation is obviously polynomial in the size of the knowledge base.

(3) Let A $(= B \sqsubseteq H)$ be an axiom of K' containing a safe occurrence of a nominal schema $\{x\}$. Let $S_i(a_i, \exists R_i.D_i, x)$ be all the safe environments for all safe occurrences of $\{x\}$ in B, and let $C_{x,A}$ be a new atomic concept. Do the following.

1. For each individual b in K', add to K' the axiom
$$\prod_i (\exists U.S_i(a_i, \exists R_i.D_i, b)) \sqsubseteq \exists U.(\{b\} \sqcap C_{x,A}).$$
2. In B, first add $\exists U.C_{x,A}$ as a conjunct, resulting in $B \sqcap \exists U.C_{x,A}$. Then replace each $S_i(a_i, \exists R_i.D_i, x)$ by $\{a_i\}$. Finally, replace the (single) remaining (non-safe) occurrence of $\{x\}$, if there is any, by $C_{x,A}$.

The number of new axioms added by this step is bounded from above by y^2, where y is the maximum of the number of axioms and the number of individuals in the original knowledge base. Hence, the step is polynomial.

The knowledge base resulting from steps (1) to (3) is nominal schema-free and thus a $\mathcal{SROEL}(\sqcap, \times)$ knowledge base. The arguments just given also prove Theorem 4 below. Theorem 5, a key result, will be proven in Section 4.1, after some preparations.

[8]As mentioned earlier, the requirements concerning occurrences of nominal schemas, as given in item 6 of Definition 4, can be relaxed. In fact, it is only required that after the above mentioned grounding, the resulting knowledge base is such that, in each axiom $B \sqsubseteq H$, nominal schemas occur only in B and occur at most once in non-safe form in B.

THEOREM 4. *The transformation is polynomial* wrt *the size of the knowledge base.*

THEOREM 5. *The transformation preserves satisfiability.*

THEOREM 6. *Reasoning in $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$, for any fixed n, is possible in polynomial time in the size of the knowledge base. This encompasses satisfiability checking, instance retrieval, and computing class subsumptions.*

PROOF. This follows from the facts that the transformation yields a $\mathcal{SROEL}(\sqcap, \times)$ knowledge base polynomial in size relative to the size of the input (Theorem 4), the transformation is satisfiability preserving (Theorem 5), and reasoning in $\mathcal{SROEL}(\sqcap, \times)$ is polynomial. $\square$

4. DL-SAFE DATALOG

We now turn to the issue of expressing DL-safe datalog rules in our new languages. We first recall definitions for datalog.

$\Sigma = \langle N_I, N_P, N_V \rangle$ is a *datalog signature*, where N_I, N_P, and N_V are finite and pairwise disjoint sets of *individual names*, *predicates*, and *variables*. The set of terms $\mathbf{T}$ of Σ is $N_I \cup N_V$. A datalog *atom* is any expression of the form $P(t_1, \ldots, t_n)$, where $P \in N_P$ and each $t_i \in \mathbf{T}$.

As before, we often assume a single signature Σ and omit explicit references to it. We also assume that each predicate has an associated arity, and that $\top$ and $\bot$ are unary. Below, we assume only unary and binary predicates appear in N_P, and so N_P can be viewed as consisting of disjoint sets of concept and role predicates.

If B is a finite and nonempty set of atoms and H is an atom, then $B \to H$ is a *datalog rule*. A *datalog rule base* is a set RB of datalog rules. For a (fixed) positive integer n, an *n-variable datalog rule base* is a datalog rule base where each rule contains at most n distinct variables.

Definition 6. Interpretations ($\mathcal{I}$), assignments (Z), and the function $.^{\mathcal{I},Z}$ are defined for datalog terms and atoms as they are in $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$. The interpretation of $\top$ and $\bot$ is fixed, as it is in any DL. Satisfaction for atoms, rules, and rule bases is defined below. Let $\mathcal{I}$ be an interpretation and Z a variable assignment.

1. If $P(t_1, \ldots, t_n)$ is an atom,
   - $\mathcal{I}, Z \models P(t_1, \ldots, t_n)$ *iff* $(t_1^{\mathcal{I},Z}, \ldots, t_n^{\mathcal{I},Z}) \in P^{\mathcal{I}}$; otherwise $\mathcal{I}, Z \not\models P(t_1, \ldots, t_n)$.
2. If S is a set of atoms,
   - $\mathcal{I}, Z \models S$ *iff* $\mathcal{I}, Z \models A$ for each $A \in S$.
   - $\mathcal{I}, Z \not\models S$ *iff* $\mathcal{I}, Z \not\models A$ for at least one $A \in S$.
3. If $B \to H$ is a datalog rule, $\mathcal{I} \models B \to H$, *iff* $\mathcal{I}, Z \models H$ or $\mathcal{I}, Z \not\models B$ for each variable assignment Z.
4. If RB is a datalog rule base, then $\mathcal{I}$ satisfies RB *iff* it satisfies each rule in RB.

$\mathcal{I}$ is a *model* of an atom, rule, or rule base if it satisfies it.

Definition 6 defines a first-order logic semantics for datalog. It should be noted that a ground atom is entailed by a datalog rule base under this first-order logic semantics if, and only if, it is entailed by it under the Herbrand semantics [16]. This means, in turn, that the set of ground atoms entailed by a datalog rule base is the same whether we use the first-order logic semantics, the Herbrand semantics, or the semantics resulting from interpreting all rules in the rule-base as DL-safe. If we look at other than ground entailments, this is no longer the case. However, for datalog, usually only ground entailments are considered.

4.1 Embedding into $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$

If RB is a datalog rule base defined over a signature $\Sigma = \langle N_I, N_P, N_V \rangle$, then RB can be embedded into an equisatisfiable $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base $trans(RB)$ over a signature Σ'. The embedding obviously also works for $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$. Below, let $N_{P,i}$, $i \in \{1, 2\}$ be the unary and binary predicates of N_P.

Definition 7. If $\Sigma = \langle N_I, N_P, N_V \rangle$:

1. $\Sigma' = \langle N_I, N_C, N_R, N_V \rangle$, where
   (a) $N_C = N_{P,1}$;
   (b) $N_R = N_{P,2} \cup \{U\}$ (where U is the universal role).
2. for each unary or binary atom A over Σ,
   (a) if A is $R(t, u)$, where $t, u \in \mathbf{T}$, then $trans(A)$ is $\exists U.(\{t\} \sqcap \exists R.\{u\})$.
   (b) if A is $C(t)$, where $t \in \mathbf{T}$, then $trans(A)$ is $\exists U.(\{t\} \sqcap C)$.
3. for each set $X = \{A_1, \ldots, A_n\}$ of binary and unary atoms, $trans(X) = trans(A_1) \sqcap \ldots \sqcap trans(A_n)$.
4. for each $B \to H \in RB$, $trans(r) = trans(B) \to trans(H)$.
5. $trans(RB) = \{trans(r) | r \in RB\}$.

Observe that no role inclusion axioms appear in $trans(RB)$, and so all roles can be considered simple. Since this is so, $trans(RB)$ is a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base, where n is the maximum number of nominal schemas appearing in any rule of $trans(RB)$.

Every interpretation for RB or its translation can be taken as one for its counterpart, provided U is interpreted in the usual fashion. As this is so, we assume a common domain of discourse $\Delta^{\mathcal{I}}$.

Lemma 1. *Let A be a binary or unary atom in RB. For any interpretation $\mathcal{I}$ and assignment Z,*

1. *$\mathcal{I}, Z \models A$ iff $trans(A)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$ and*
2. *$\mathcal{I}, Z \not\models A$ iff $trans(A)^{\mathcal{I},Z} = \varnothing$.*

Proof. For part (1), suppose $\mathcal{I}, Z \models A$ and let $d \in \Delta^{\mathcal{I}}$. If $A = C(t)$, then $t^{\mathcal{I},Z} \in C^{\mathcal{I}}$. Since $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$ and $t^{\mathcal{I},Z} \in \{t\}^{\mathcal{I},Z}$, it follows that $d \in (\exists U.(\{t\} \sqcap C))^{\mathcal{I},Z}$. If instead $A = R(t, u)$, then $(t^{\mathcal{I},Z}, u^{\mathcal{I},Z}) \in R^{\mathcal{I}}$. Since $u^{\mathcal{I},Z} \in \{u\}^{\mathcal{I},Z}$, it must be that $t^{\mathcal{I},Z} \in (\exists R.\{u\})^{\mathcal{I},Z}$. Since $t^{\mathcal{I},Z} \in \{t\}^{\mathcal{I},Z}$ and $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$, it follows that $d \in (\exists U.(\{t\} \sqcap \exists R.\{u\}))^{\mathcal{I},Z}$. Generalizing on d, $trans(A)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$.

Now suppose $trans(A)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$. If $trans(A)$ is $\exists U.(\{t\} \sqcap C)$, then for any $d \in \Delta^{\mathcal{I}}$, it follows that $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$ and $t^{\mathcal{I},Z} \in C^{\mathcal{I},Z}$. And so $\mathcal{I}, Z \models C(t)$. If $trans(A)$ is $\exists U.(\{t\} \sqcap \exists R.\{u\})^{\mathcal{I},Z}$, then it follows that for any $d \in \Delta^{\mathcal{I}}$, $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$ and $t^{\mathcal{I},Z} \in (\exists R.\{u\})^{\mathcal{I},Z}$. From this, $(t^{\mathcal{I},Z}, u^{\mathcal{I},Z}) \in R^{\mathcal{I},Z}$, and so $\mathcal{I}, Z \models R(t, u)$.

For (2), suppose $\mathcal{I}, Z \not\models A$. Suppose for a proof by contradiction that there is a $d \in \Delta^{\mathcal{I}}$ such that $d \in trans(A)^{\mathcal{I},Z}$. If $A = C(t)$, then $t^{\mathcal{I},Z} \notin C^{\mathcal{I}}$. Since $d \in trans(A)^{\mathcal{I},Z}$, it follows that $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$ and $t^{\mathcal{I},Z} \in \{t\}^{\mathcal{I},Z}$ and $t^{\mathcal{I},Z} \in C^{\mathcal{I}}$. If instead $A = R(t, u)$, then $(t^{\mathcal{I},Z}, u^{\mathcal{I},Z}) \notin R^{\mathcal{I}}$. Since $d \in trans(A)^{\mathcal{I},Z}$, it follows that $(d, t^{\mathcal{I},Z}) \in U^{\mathcal{I},Z}$ and $(t^{\mathcal{I},Z}, u^{\mathcal{I},Z}) \in R^{\mathcal{I},Z}$. Either way, a contradiction occurs, and so $trans(A)^{\mathcal{I},Z} = \varnothing$.

Now suppose $trans(A)^{\mathcal{I},Z} = \varnothing^{\mathcal{I}}$. If $\mathcal{I}, Z \models A$, then from the first part of the proof, $trans(A)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$. And so it must be that $\mathcal{I}, Z \not\models A$. $\square$

The following is the key result of the embedding.

Theorem 7. *RB is satisfiable* iff *$trans(RB)$ is.*

Proof. Let $\mathcal{I}$ be a model of RB and Z an assignment for $\mathcal{I}$. Suppose $B \to H \in RB$. If $\mathcal{I}, Z \models H$, then $trans(H)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$ by Lemma 1. Similarly, if $\mathcal{I}, Z \not\models B$, then there exists a $B_i \in B$ such that $\mathcal{I}, Z \not\models B_i$. Again by Lemma 1, $trans(B_i)^{\mathcal{I},Z} = \varnothing$, and so $trans(B)^{\mathcal{I},Z} = \varnothing$. Either way, $\mathcal{I}, Z \models trans(B \to H)$. Generalizing on Z and $B \to H$, $\mathcal{I}$ models $trans(RB)$.

Now let $\mathcal{I}$ be a model of $trans(RB)$ and Z an assignment for $\mathcal{I}$. Suppose $B \to H \in RB$ and $\mathcal{I}, Z \models B$. For each $A \in B$, $\mathcal{I}, Z \models A$. By Lemma 1, $trans(A)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$ for each $A \in B$. From this, it follows that $trans(B)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$. Since $\mathcal{I}, Z \models trans(B \to H)$, it must be that $trans(H)^{\mathcal{I},Z} = \Delta^{\mathcal{I}}$, and so $\mathcal{I}, Z \models H$ by Lemma 1. As such, $\mathcal{I}, Z \models B \to H$. Generalizing on Z and $B \to H$, $\mathcal{I}$ models RB. $\square$

The key insight from Theorem 7 is that RB and $trans(RB)$ yield the same ground (i.e., ABox) entailments: RB entails a ground atom A *iff* $RB \cup \{A \to \bot\}$ is unsatisfiable.

The following theorem is more tricky. While it shows that $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge bases can be combined with n-variable datalog rule bases without jeopardizing polynomiality, it is important to realize that the datalog rules are semantically interpreted in a DL-safe way, through the use of nominal schemas in the transformation. Hence, the set of entailments of such a combined knowledge base differs from the set of entailments which the hybrid knowledge base would have under a straightforward first-order logic semantics. Entailment under such a first-order semantics, however, would be undecidable.

Theorem 8. *If KB is a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base and RB an n-variable datalog rule base, then $KB' = KB \cup trans(RB)$ is a $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge base.*

PROOF. Individually, both KB and $trans(RB)$ are clearly $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge bases. Examination of the restrictions placed on $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ knowledge bases reveals that KB' satisfies them all. E.g., no role chains, RIAs, or role conjunctions appear in $trans(RB)$; roles from KB in $trans(RB)$ only appear in existential restrictions, and so whether they are simple or not is irrelevant. $\square$

We are now in a position to provide a proof for Theorem 5.

PROOF OF THEOREM 5. We have to show that each of the transformation steps (1) to (3) preserves satisfiability. For step (1), this follows from the same arguments used to prove Theorem 1.

For step (2), let K be the knowledge base before the transformation, and let K' be the knowledge base after the transformation. Given a model M of K, a model M' of K' is obtained by defining $O^{M'}$ to be $(N_I)^M$. Given a model of K', a model for K is obtained by restricting the signature (i.e., removing O).

For step (3), again let K be the knowledge base before the transformation, and let K' be the knowledge base after the transformation. Now note that the axiom

$$\prod_i (\exists U.S_i(a_i, \exists R_i.D_i, b)) \sqsubseteq \exists U.(\{b\} \sqcap C_{x,A})$$

as given in step (3) of the transformation is similar in structure to the results of embedding datalog rules as discussed above. This insight guides the rest of the proof.

For any interpretation $\mathcal{I}$ for K and any concept $C_{x,A}$, we extend $\mathcal{I}$ to $\mathcal{I}'$ by defining $C_{x,A}^{\mathcal{I}'}$ to be the set of $b^{\mathcal{I}}$ ($b \in N_I$) such that $\prod_i (\exists U.S_i(a_i, \exists R_i.D_i, b))^{\mathcal{I}}$ is not empty.[9] Below, $Cj(A, x, b)$ is shorthand for such a conjunction. We assume a common domain of discourse Δ for both $\mathcal{I}$ and $\mathcal{I}'$.

(LR) Assume M models K and $A \in K$. If A contains no nominal schemas (this holds for every RIA and ABox assertion), M is a model of A *iff* M' is. So suppose A is the TBox axiom $B \sqsubseteq H$ such that $S_1(a_1, \exists R_1.D_1, v) \ldots S_n(a_n, \exists R_n.D_n, v)$ are the safe environments for $\{v\}$ in B. Let Z be a variable assignment. The transformation produces axioms of the below two forms:

1. $\prod_i (\exists U.S_i(a_i, \exists R_i.D_i, b)) \sqsubseteq \exists U.(\{b\} \sqcap C_{v,A})$, $b \in N_I$.
2. $\exists U.C_{v,A} \sqcap B^* \sqsubseteq H$

where $C_{v,A}$ is a new atomic concept and B^* is B with each $S_i(a_i, \exists R_i.D_i, v)$ replaced with $\{a_i\}$, and the one non-safe occurrence (if present) of $\{v\}$ is replaced with $C_{v,A}$. No axiom of form 1 contains any nominal schema.

Let P be a form 1 axiom (with $b \in N_I$ replacing v). Suppose $x \in Cj(A, v, b)^{M',Z}$. According to the specification of $C_{v,A}^{M'}$, $b^{M'} \in C_{v,A}^{M',Z}$. Clearly, $(x, b^{M'}) \in U^{M',Z}$, and so $x \in (\exists U.(\{b\} \sqcap C_{v,A}))^{M',Z}$. Generalizing on P, M' and Z satisfy each form 1 axiom.

Suppose P has form 2 and $x \in (\exists U.C_{v,A} \sqcap B^*)^{M',Z}$. Then there is a $d \in N_I$ such that $(x, d^{M'}) \in U^{M',Z}$ and $d^{M'} \in C_{v,A}^{M',Z}$. From this, $Cj(A, v, d)^{M',Z} \neq \varnothing$. If there is an occurrence of $C_{v,A}$ in B^*, then the following holds:

> (∗) There is an $e \in N_I$ with $e^{M'} \in C_{v,A}^{M',Z}$ such that $x \in (\exists U.C_{v,A} \sqcap B^{*\prime})^{M',Z}$, where $B^{*\prime}$ is obtained from B^* by replacing $C_{v,A}$ by $\{e\}$.

To show (∗), we distinguish two cases. (1) If $B^* = C_{v,A} \sqcap D$ for some concept description D, then $x \in C_{v,A}^{M',Z}$. By definition of the extension of $C_{v,A}$ under M', there must be an $e \in N_I$ with $x = e^{M'}$, and so $x \in (\exists U.C_{v,A} \sqcap \{e\} \sqcap D)^{M',Z}$. I.e., $x \in (\exists U.C_{v,A} \sqcap B^{*\prime})^{M',Z}$. (2) Otherwise, note that B^* is of the form

$$B^* = D_1 \sqcap \exists R_1.(D_2 \sqcap \exists R_2.(\ldots (D_k \sqcap \exists R_k.(D_{k+1} \sqcap C_{v,A})) \ldots)),$$

where all D_j are concept descriptions. Since we have $x \in (B^*)^{M',Z}$, there must be a selection of elements $y_0, \ldots, y_k \in \Delta^{M'}$ such that $x = y_0$ and $(y_{j-1}, y_j) \in R_j^{M',Z}$ for all $j = 1, \ldots, k$. By definition of the extension of $C_{v,A}$ under M', there must thus be an $e \in N_I$ with $y_k = e^{M'} \in C_{v,A}^{M',Z}$, and (∗) follows in this case. This concludes the proof of (∗).

Now note that $x \in (B^*)^{M',Z}$ and thus $x \in (B^{*\prime})^{M',Z}$. Let Z' be an assignment such that $Z'(u) = Z(u)$ for all variables $u \neq v$ and such that $Z'(v) = e^M$ if $C_{v,A}$ occurs in B^*, and $Z'(v) = d^M$ otherwise. We induct on the depth of the expression tree of $B^{*\prime}$, showing that for any sub-concept C^* of $B^{*\prime}$, and for any $u \in \Delta$, if $u \in (C^*)^{M',Z}$, then $u \in C^{M,Z'}$, where C is the subformula of B from which C^* is constructed.

For the base case (the leaves, where $\exists R.Self$ is considered a leaf), if $C^* = C$, then $\{v\}$ does not appear in C, and so $(C^*)^{M',Z} = C^{M,Z'}$. So suppose $C^* \neq C$. Since C^* is a leaf, we must distinguish two cases. (i) If $C^* = \{e\}$ (coming from $C_{v,A}$ via (∗)), then $u \in C^{M,Z'}$ by definition of Z'. (ii) In the remaining case, the leaf has the form $\{a\}$, where C is of the form $\{a\} \sqcap \exists R.\{v\}$. As such, the portion of $Cj(A, v, c)$ corresponding to C is just $\exists U.(\{a\} \sqcap \exists R.\{c\})$. It follows that $u = a^{M'} = a^M$ and $\exists U.(\{a\} \sqcap \exists R.\{c\})^{M,Z'} \neq \varnothing$ (because $Cj(A, v, c)^{M',Z} \neq \varnothing$ and $Cj(A, v, c)^{M',Z} = Cj(A, v, c)^{M,Z'}$). Let $z \in \exists U.(\{a\} \sqcap \exists R.\{c\})^{M,Z'}$. Then $(z, a^M) \in U^{M,Z'}$ and $a^M \in \exists R.\{c\})^{M,Z'}$. Since $u = a^M$, $u \in (\{a\} \sqcap \exists R.\{c\})^{M,Z'}$. It follows that $u \in (\{a\} \sqcap \exists R.\{v\})^{M,Z'}$, i.e., $u \in C^{M,Z'}$.

For the induction, it must be that C is of the form $C_1 \sqcap C_2$ or $\exists R.C_1$. If the former, then $u \in ((C_1 \sqcap C_2)^*)^{M',Z}$ and so $u \in (C_1^*)^{M',Z}$ and $u \in (C_2^*)^{M',Z}$. By inductive hypothesis, $u \in C_1^{M,Z'}$ and $u \in C_2^{M,Z'}$, and so $u \in (C_1 \sqcap C_2)^{M,Z'}$. If the latter, then there exists a $w \in \Delta$ such that $(u, w) \in R^{M',Z}$

[9] Replacing v with b yields an expression containing *no* nominal schemas, and so variable assignments can be ignored.

and $w \in (C_1^*)^{M',Z}$. By inductive hypothesis, $w \in C_1^{M,Z'}$. From this, $u \in \exists R.C_1^{M,Z'}$. Either way, $u \in C^{M,Z'}$.

Given the above induction, $x \in B^{M,Z'}$. Since M models $B \sqsubseteq H$, $x \in H^{M,Z'}$. Since H contains neither nominal schemas nor $C_{v,A}$, $H^{M,Z'} = H^{M,Z}$, and so $x \in H^{M',Z}$. Given this, M' and Z satisfy P.

Generalizing on Z and then A, M' models K'.

(RL) Let M' be a model of K' and M its restriction to the terms appearing in K. Let $A \in K$; we assume that A is a concept inclusion axiom of the form $B \sqsubseteq H$. Let Z be an assignment and suppose that $x \in B^{M,Z}$. Observe that $B^{M,Z} = B^{M',Z}$ and furthermore that $Z(v) = c^M = c^{M'}$ for some $c \in N_I$. We may assume *wlog* that $\perp$ does not appear in B, since otherwise A would be trivially modelled by every interpretation $\mathcal{I}$. Since this is so, and since $x \in B^{M',Z}$ it must be that each $S_i(a_i, \exists R_i.D_i, c)^{M',Z}$ is nonempty. As such, there is a $z \in \Delta$ such that $z \in S_i(a_i, \exists R_i.D_i, c)^{M',Z}$. It follows that $x \in (\exists U.(S_i(a_i, \exists R_i.D_i, c)))^{M',Z}$. Generalizing on S_i, $x \in Cj(A, v, c)^{M',Z}$. From this, $x \in \exists U.(\{c\} \sqcap C_{v,A})^{M',Z}$, and so $x \in \exists U.(C_{v,A})^{M',Z}$.

We now induct on the degree of B^*, showing that for any subconcept C^* of B^* and any $u \in \Delta$, if $u \in C^{M',Z}$, then $u \in (C^*)^{M',Z}$. For the base case, If $C^* = C$, then there's nothing to prove. For $C^* \neq C$ we consider the two possible cases. If C^* is $\{a\}$, corresponding to the safe environment $C = \{a\} \sqcap \exists R.\{v\}$, then $u \in (\{a\} \sqcap \exists R.\{v\})^{M',Z}$, and so clearly $u \in (C^*)^{M',Z}$. If instead C^* is $C_{v,A}$, corresponding to $C = \{v\}$, then $u = v^{M',Z} = c^{M'}$. Recall that $x \in Cj(A, v, c)^{M',Z}$, i.e., $Cj(A, v, c)^{M',Z}$ is not empty. As such, $c^{M'} \in C_{v,A}^{M',Z}$, and so $u \in (C^*)^{M',Z}$.

For the induction, we consider the two viable forms of the C corresponding to C^*. If C has form $C_1 \sqcap C_2$, then $u \in C_1^{M',Z}$ and $u \in C_2^{M',Z}$, and so by inductive hypothesis $u \in (C_1^*)^{M',Z}$ and $u \in (C_2^*)^{M',Z}$. As such, $u \in ((C_1 \sqcap C_2)^*)^{M',Z}$. If C is instead of the form $\exists R.C_1$, then there exists a $w \in \Delta$ such that $(u, w) \in R^{M',Z}$ and $w \in (C_1)^{M',Z}$. By inductive hypothesis, $w \in (C_1^*)^{M',Z}$. From this, $u \in ((\exists R.C_1)^*)^{M',Z}$.

Since $x \in B^{M',Z}$, we conclude that $x \in (B^*)^{M',Z}$, and therefore $x \in (\exists U.C_{v,A} \sqcap B^*)^{M',Z}$. As M' models K', we may conclude $x \in H^{M',Z}$, and since $H^{M',Z} = H^{M,Z}$, $x \in H^{M,Z}$.

And so $M, Z \models A$. Generalizing on Z, M models A. Generalizing on A, M models K. $\square$

4.2 Embedding DLP 2 and DL-Lite$_\mathcal{R}$

For the following, note that DLP 2 [13] is essentially OWL 2 RL [18]. We show that it can be embedded into ELP 2.

A DLP role is any $\mathcal{SROIQ}$ role expression. A DLP *body concept description* is a $\mathcal{SROIQ}$ concept description constructed from elements of $N_C \cup O \cup \{\top, \perp\}$ and $\exists$ and $\sqcap$. A DLP *head concept description* is a $\mathcal{SROIQ}$ concept description constructed from elements of $N_C \cup O \cup \{\top, \perp\}$ and $\forall$, $\sqcap$, and expressions of the form $\leq 1R.C$, where R is a $\mathcal{SROIQ}$ role expression and C is a body concept description. A DLP RBox axiom is any expression of the form $R \sqsubseteq S$ or $R \circ R \sqsubseteq R$, where R,S are DLP roles. A DLP TBox axiom is any expression of the form $C \sqsubseteq D$, where C is a DLP body concept and D is a DLP head concept. An DLP ABox axiom is any expression of the form $D(a)$ or $R(a, b)$, where D is a DLP head concept, R is a DLP role, and $a, b \in N_I$. A DLP knowledge base is a set of Rbox, TBox, and ABox axioms.

Let $\mathbf{T}$ be a set of terms as defined earlier. If C is a DLP concept description and $t \in \mathbf{T}$, then $C(t)$ is a *concept atom*. If R is a DLP role and $t, u \in \mathbf{T}$, then $R(t, u)$ is a *role atom*.

Given a set S of (concept and role) atoms, and $t, u \in \mathbf{T}$, a *path* from t to u in S is a nonempty sequence $R_1(x_1, x_2)$, $\ldots$, $R_1(x_n, x_{n+1})$, where $x_1 = t$, $x_{n+1} = u$, and $x_i \in N_V$ for all $2 \leq i \leq n$. A term t is *initial* in S if there is no path to t in S; t is *final* in S if there is no path starting with t in S.

A *DL rule* is any expression of the form $B \to H$, where B and H are sets of atoms and:

1. for any term u not initial in B, there is a path in B from exactly one initial term to u;
2. for any terms t, u, there is exactly one path from t to u in B;
3. if $C(t)$ or $R(t, u)$ are in H, then t is initial in B.

A *DL rule base* is any set of DL rules. A *DLP rule base* is a DL rule base such that every concept atom in B is a body concept, and every concept in H is a head concept. A *DLP 2 knowledge base* is a DLP knowledge base together with a DLP rule base. Axioms of the form $Dis(R, S)$, and $Asy(R)$ are also allowed, where R and S are DLP roles.

Theorem 9. *Any DLP 2 knowledge base can be converted into an equisatisfiable $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$ knowledge base in polynomial time.*

Proof. As shown in [13], a DLP 2 knowledge base KB can be converted in polynomial time to a datalog rule base RB in which 1) each rule of RB has at most 5-variables, and 2) KB is satisfiable if and only if RB is. Each predicate in the rule base RB is either unary or binary, and so the conversions of Section 4 apply. According to Theorem 7, RB is satisfiable if and only if $trans(RB)$ is, where $trans(RB)$ is the $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_n)$ translation of RB. We conclude that KB and $trans(RB)$ are equisatisfiable. $\square$

DL-Lite$_\mathcal{R}$ is essentially OWL 2 QL [18]. From the definition of DL-Lite$_\mathcal{R}$ (which we do not repeat here), it is immediately clear that all of the language, except inverse roles, is covered by $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$.

As for inverse roles, note that the vast majority of uses of inverse roles in existing DL-Lite$_\mathcal{R}$ knowledge bases are actually for expressing range restrictions,[10] which are also expressible in $\mathcal{SROEL}(\sqcap, \times, \{\mathbf{v}\}_5)$. For other uses of inverse

[10] D. Calvanese and R. Rosati, personal communications.

roles, we can approximate statements of the form $R \sqsubseteq S^-$ by the DL-safe rule $R(x, y) \to S(y, x)$, or more precisely by their corresponding DL axiom as described in Section 4.

We can thus say that ELP 2 covers most of OWL 2 QL.

5. CONCLUSIONS AND FUTURE WORK

We have provided a language which seamlessly integrates OWL DL and datalog (and thus SWRL and RIF-Core) rules. The language is firmly based on description logic traditions and adheres to the usual conceptual design decisions, including decidability of the language. We have also provided a sublanguage, called ELP 2, which is tractable and covers variable-restricted datalog and all major tractable profiles of OWL 2 DL (with the mentioned exception of some uses of inverse roles in OWL 2 QL).

The complexity analyses which we provided also yield key steps to a naive algorithmization of reasoning in these languages. However, there is much more work to be done in this respect, in particular regarding efficient algorithmizations and implementations.

For $\mathcal{SROIQB}_s(\times, \{\mathbf{v}\})$, naive grounding together with an extension of existing algorithms for dealing with role constructors provides a first approach to algorithmization. It is expected that system performance will be the main bottleneck in such investigations.

For ELP 2, the transformation provided in the proof of Theorem 5 provides a starting point, possibly together with reasoning approaches laid out for $\mathcal{SROEL}(\sqcap, \times)$ in [12]. However, we believe that there should be better solutions by developing algorithms which avoid the blow-up of the knowledge base through grounding, and we intend to investigate these in the near future.

Since datalog rules are expressible in ELP 2, ELP 2 may also provide a natural gateway to incorporating closed-world reasoning into description logics. This could be accomplished by applying the semantics of nonmonotonic rule-based reasoning approaches to the transformed rules.

Acknowledgements This work was supported by the National Science Foundation under award 1017225 "III: Small: TROn—Tractable Reasoning with Ontologies". The second author is funded by a Fulbright Indonesia Presidential Scholarship PhD Grant 2010.

6. 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] 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/`.

[3] Andrea Calì, Georg Gottlob, and Andreas Pieris. Query answering under non-guarded rules in datalog+/-. In Pascal Hitzler and Thomas Lukasiewicz, editors, *Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings*, volume 6333 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2010.

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

[5] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. *Artificial Intelligence*, 172(12–13):1495–1539, 2008.

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

[7] 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.

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

[9] 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. Available at `http://www.w3.org/Submission/SWRL/`.

[10] M. Knorr, J.J. Alferes, and P. Hitzler. Local closed-world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 2011. To appear.

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

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

[13] 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.

[14] 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.

[15] Markus Krötzsch, Frederick Maier, Adila A. Krisnadhi, and Pascal Hitzler. A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In S. Sadagopan, Krithi Ramamritham, Arun Kumar, M.P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proceedings of the 20th International World Wide Web Conference, WWW2011, Hyderabad, India, March/April 2011*, pages 645–654. ACM, New York, 2011.

[16] J.W. Lloyd. *Foundations of Logic Programming, 2nd Edition*. Springer, 1987.

[17] 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/`.

[18] 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/`.

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

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

[21] 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/`.

[22] 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/`.

[23] 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.