

# *SR<sub>OIQ</sub>* Syntax Approximation by Using Nominal Schemas

Cong Wang, David Carral, and Pascal Hitzler

Kno.e.sis Center, Wright State University, Dayton, OH, U.S.A.

**Abstract.** Nominal schemas is a recently introduced extension of description logics which makes it possible to express rules which generalize DL-safe ones. A tractable description logic,  $\mathcal{ELROV}_n$ , has been identified. This leads us to the question: can we improve approximate reasoning results by employing nominal schemas? In this paper, we investigate how to approximately cast *SR<sub>OIQ</sub>* into  $\mathcal{ELROV}_n$ . Using a datalog-based tractable algorithm, a preliminary evaluation shows that our approach can indeed do approximate *SR<sub>OIQ</sub>*-reasoning with a high recall.

## 1 Introduction

Reasoning with large or complex terminology is computationally difficult and is one of the bottlenecks for Semantic Web applications. Most reasoning tasks for ontologies underlying OWL [11] are intractable. Even with small ontologies, sound and complete reasoning is practically infeasible, in particular for applications where quick responses are critical.

This fundamental insight that expressive ontology reasoning is often necessarily of high computational complexity has triggered a line of research which aims at utilizing approximate algorithms, i.e. algorithms which are (provably) not sound and complete, but which nevertheless provide answers which are good enough for practical purposes [6,9,10,24]. This general idea of *approximate reasoning* is not new and to a certain extent had been studied already before the advent of the Semantic Web [4,25,26]. But the Semantic Web effort with its increased requirements for scalability has recently put this into a focus which this branch of reasoning research has never had before [7,8,12,20,21,22,23,27].

One of the prominent general approaches to approximate reasoning is known as *language weakening*. Language weakening refers to the idea of rewriting a knowledge base into a language which can be handled more efficiently. Obviously, if the target language has a lower complexity class, this rewriting in general cannot be done without a loss, resulting in an approximate reasoning procedure. In order to limit loss in the translation, it is of advantage if the target language be as expressive as possible while still being of low computational complexity, and hence languages which push expressivity while retaining tractability are natural choices for a language weakening approach.

In this paper, we use  $\mathcal{ELROV}_n$  for approximate reasoning over *SR<sub>OIQ</sub>* using language weakening.  $\mathcal{ELROV}_n$  is essentially a tractable extension of  $\mathcal{EL}^{++}$  [2],

**Table 1.** Normal forms of  $\mathcal{SROIQ}$  TBox axioms.  $A, B$  and  $C$  are atomic concept or negations of atomic concepts.

$A \sqsubseteq \perp$	$\perp \sqsubseteq C$	$A \sqsubseteq C$	$A \sqcap B \sqsubseteq C$	$A \sqsubseteq B \sqcup C$	
$\exists R.A \sqsubseteq C$	$A \sqsubseteq \exists R.C$	$\forall R.A \sqsubseteq C$	$A \sqsubseteq \forall R.C$	$A \sqsubseteq \{a\}$	$\{a\} \sqsubseteq A$
$\geq nR.A \sqsubseteq C$	$\leq nR.A \sqsubseteq C$	$A \sqsubseteq \leq nR.C$	$A \sqsubseteq \geq nR.C$		

a.k.a. OWL 2 EL [18], by nominal schemas [17].<sup>1</sup> As such,  $\mathcal{ELROV}_n$  incorporates DL-safe Datalog under Herbrand semantics [14]. We have recently described an efficient procedure to reasoning with  $\mathcal{ELROV}_n$  [5] on which we base the evaluations in this paper.

The plan of this paper is as follows. In Section 2 we recall the languages  $\mathcal{SROIQ}$  and  $\mathcal{ELROV}_n$ . In Section 3 we describe our approximate compilation of  $\mathcal{SROIQ}$  into  $\mathcal{ELROV}_n$ . In Section 4 we recall our  $\mathcal{ELROV}_n$  reasoning approach from [5]. In Section 5 we describe our implementation and evaluation results. In Section 6 we conclude.

## 2 Preliminaries

In this section, we introduce the description logics (DLs)  $\mathcal{SROIQ}$  and  $\mathcal{ELROV}_n$ . The latter includes the new constructor from [17], nominal schemas, which we use to approximate some features of  $\mathcal{SROIQ}$ .

A signature  $\Sigma = \langle \Sigma_I, \Sigma_C, \Sigma_R, \Sigma_S \rangle$  consists of mutually disjoint finite sets of atomic roles *role names*  $\Sigma_R$ , atomic concepts  $\Sigma_C$ , and individuals *individual*  $\Sigma_I$ , together with a distinguished subset  $\Sigma_S \subseteq \Sigma_R$  of *simple atomic roles*. The set of roles (over  $\Sigma$ ) is  $\mathbf{R} := \Sigma_R \cup \{R^- \mid R \in \Sigma_R\}$ ; the set of *simple roles* is  $\mathbf{S} := \Sigma_S \cup \{S^- \mid S \in \Sigma_S\}$ . A *role chain* is an expression of the form  $R_1 \cdot \dots \cdot R_n$  with  $n \geq 1$  and each  $R_i \in \mathbf{R}$ . The function  $\text{inv}(\cdot)$  is defined on roles by  $\text{inv}(R) := R^-$  and  $\text{inv}(R^-) := R$  where  $R \in \mathbf{R}$ , and extended to role chains by  $\text{inv}(R_1 \cdot \dots \cdot R_n) := \text{inv}(R_n) \cdot \dots \cdot \text{inv}(R_1)$ .

The set  $\mathbf{C}$  of  $\mathcal{SROIQ}$  concepts (over  $\Sigma$ ) is defined recursively as follows:

$$\mathbf{C} := \Sigma_C \mid \{\Sigma_I\} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \neg \mathbf{C} \mid \exists \mathbf{R}.\mathbf{C} \mid \forall \mathbf{R}.\mathbf{C} \mid \geq n\mathbf{S}.\mathbf{C} \mid \leq n\mathbf{S}.\mathbf{C} \mid \exists \mathbf{S}.\mathbf{Self}$$

A TBox is a finite set of general concept inclusions (GCIs) of the form  $C \sqsubseteq D$  where  $C, D \in \mathbf{C}$ . A  $\mathcal{SROIQ}$  TBox can be normalized such that it only contains the normal forms in Table 1 [1].

Satisfiability checking of  $\mathcal{SROIQ}$  ontologies is in  $\text{N2ExpTime}$  [13]. Given a disjunctive assertion  $(C \sqcup D)(s)$ , the tableau algorithm [13] nondeterministically guesses that either  $C(s)$  or  $D(s)$  holds, which can give rise to exponential behavior. Although the *absorption* technique and the *hypertableaux* approach [19] reduce the cost of this nondeterminism, it is still a considerable performance bottleneck.

<sup>1</sup> It was called  $\mathcal{SROELV}_n$  in [17].

$\mathcal{SROIQ}$  defines simple roles and role regularity to ensure decidability [13]. However, since we will later approximately cast  $\mathcal{SROIQ}$  into  $\mathcal{ELROV}_n$ , which is free of these restrictions, we do not have to concern ourselves with them for the purposes of this paper.  $\mathcal{ELROV}_n$  extends  $\mathcal{EL}^{++}$  with nominal schemas (see [5,17] for details). To deal with the new constructor, we extend the signature to  $\Sigma = \langle \Sigma_I, \Sigma_C, \Sigma_R, \Sigma_V \rangle$ , where  $\Sigma_V$  is a set of variables. A *nominal schema* is a concept of the form  $\{x\}$  where  $x \in \Sigma_V$ . Semantically, these variables can only bind to known individuals. The  $n$  in  $\mathcal{ELROV}_n$  is a global bound on the number of different nominal schemas which can occur in any axiom in a knowledge base—this restriction guarantees tractability. The set of  $\mathbf{C}$  of  $\mathcal{ELROV}_n$  concepts is defined as follows:

$$\mathbf{C} := \Sigma_C \{ \{ \Sigma_I \} \mid \{ \Sigma_V \} \} \mathbf{C} \sqcap \mathbf{C} \mid \exists \mathbf{R}. \mathbf{C} \mid \exists \mathbf{S}. \mathbf{Self}$$

To give an example, consider the first-order rule

$$R_1(x, y) \wedge R_2(y, z) \wedge R_3(x, z) \rightarrow R(x, z)$$

which cannot be translated faithfully into  $\mathcal{SROIQ}$ . By limiting the variable  $z$  in the sense that it can bind only to known individuals (such variables are called *DL-safe* [16]), we can express this rule in  $\mathcal{ELROV}_n$  as

$$\exists R_1. \exists R_2. \{z\} \sqcap \exists R_3. \{z\} \sqsubseteq \exists R. \{z\}.$$

If  $a_1, \dots, a_k$  are all the known individuals in the knowledge base, then this axiom can also be expressed using the  $k$   $\mathcal{SROIQ}$ -axioms

$$\exists R_1. \exists R_2. \{a_i\} \sqcap \exists R_3. \{a_i\} \sqsubseteq \exists R. \{a_i\}$$

where  $i$  ranges from 1 to  $k$ . This kind of conversion, called *full* or *naive grounding*, of nominal schemas into classical description logics is, however, computationally infeasible [5] even for  $\mathcal{ELROV}_n$ , which is of PTime complexity [17]. In [5], we thus presented a datalog-based algorithm for  $\mathcal{ELROV}_n$  which avoids full grounding, and have also shown experimentally that the algorithm is efficient.

### 3 Approximation

For our approximation of  $\mathcal{SROIQ}$  by  $\mathcal{ELROV}_n$ , we use a number of different techniques, some of which are borrowed from existing literature. The key ideas are as follows.

- We rewrite mincardinality restrictions into maxcardinality restrictions or approximate using an existential.
- We rewrite universal quantification into existential quantification.
- We approximate maxcardinality restrictions using functionality.
- We approximate inverse roles and functionality using nominal schemas.
- We approximate negation using class disjointness.

---

**Algorithm 1** Approximation Algorithm

---

```
1: normalize the  $\mathcal{SROIQ}$  KB into normal forms;
2: for each concept  $C$  do
3:   introduce a fresh concept  $neg(C)$ ;
4:   add axiom  $C \sqcap neg(C) \sqsubseteq \perp$ ;
5: end for
6: for each role  $R^-$  appearing in KB do
7:   introduce a fresh role  $inv(R)$ ;
8:   add  $\{x\} \sqcap \exists R.\{y\} \sqsubseteq \{y\} \sqcap \exists inv(R).\{x\}$ ;
9: end for
10: for each axiom  $a$  in TBox do
11:   if  $a$  is of type  $A \sqsubseteq C$  then
12:     add axiom  $neg(A) \sqsubseteq neg(C)$ ;
13:   else if  $a$  is of type  $A \sqsubseteq B \sqcup C$  then
14:     add axiom  $neg(B) \sqcap neg(C) \sqsubseteq neg(A)$ ;
15:   else if  $a$  is of type  $A \sqsubseteq \forall R.C$  then
16:     add axiom  $\exists R.neg(C) \sqsubseteq neg(A)$ ;
17:     add axiom  $\exists inv(R).A \sqsubseteq C$  and  $\{x\} \sqcap \exists R.\{y\} \sqsubseteq \{y\} \sqcap \exists inv(R).\{x\}$ ;
18:   else if  $a$  is of type  $\forall R.A \sqsubseteq C$  then
19:     add axiom  $neg(C) \sqsubseteq \exists R.neg(A)$ ;
20:   else if  $a$  is of type  $C \sqsubseteq \geq nR.A$  then
21:     add axiom  $C \sqsubseteq \exists R.A$ ;
22:   else if  $a$  is of type  $C \sqsubseteq \leq nR.A$  then
23:     add axiom  $C \sqcap \exists R.(\{z1\} \sqcap A) \sqcap \exists R.(\{z2\} \sqcap A) \sqsubseteq \exists U.(\{z1\} \sqcap \{z2\})$ ;
24:   else if  $a$  is of type  $\leq nR.A \sqsubseteq C$  then
25:     add axiom  $neg(C) \sqsubseteq \exists R.A$ ;
26:   else if  $a$  is of type  $\geq nR.A \sqsubseteq C$  then
27:     add axiom  $neg(C) \sqcap \exists R.(\{z1\} \sqcap A) \sqcap \exists R.(\{z2\} \sqcap A) \sqsubseteq \exists U.(\{z1\} \sqcap \{z2\})$ ;
28:   else
29:     add axiom  $a$ ;
30:   end if
31: end for
```

---

– We approximate disjunction using conjunction.

A pseudocode description is given in Algorithm 1, we explain the relevant parts in more detail below. Role chain axioms are left untouched, as are axioms which can already directly be expressed in  $\mathcal{ELROV}_n$ . We drop the soundness proof, since one can easily find out that our approach is sound but incomplete.

### 3.1 Approximation of Inverse Role and Functionality

Since  $\mathcal{ELROV}_n$  can express DL-safe Datalog rules, all rule-like axioms in  $\mathcal{SROIQ}$  can be approximated easily in  $\mathcal{ELROV}_n$ .

For role inclusion axioms of the form  $R \sqsubseteq S^-$ , the first-order logic rule is  $R(x, y) \rightarrow S(y, x)$ . By restricting the variables to nominals, we obtain  $nom(x) \wedge nom(y) \wedge R(x, y) \rightarrow S(y, x)$ , where  $nom(x)$  is defined by the collection of facts

$nom(a_i)$  for each individual  $a_i$ . The latter rule can be expressed by means of the nominal schema axiom,

$$\{x\} \sqcap \exists R.\{y\} \sqsubseteq \{y\} \sqcap \exists S.\{x\}$$

where  $x$  and  $y$  are nominal schemas. This axiom will be later translated into datalog rule,

$$nom(x), nom(y), triple(x, R, y) \rightarrow triple(y, S, x)$$

where we can clearly see that the rule expresses the inverse role with restricting variable bounded to known individuals.

Similarly, for a functionality axiom  $C \sqsubseteq \leq 1R.D$ , we can cast it into

$$C \sqcap \exists R.(\{z1\} \sqcap D) \sqcap \exists R.(\{z2\} \sqcap D) \sqsubseteq \exists U.(\{z1\} \sqcap \{z2\})$$

where  $U$  is the universal role. This axiom will be translated into two datalog rules:

$$nom(z1), nom(z2), inst(x, C), inst(x, D), triple(x, R, z1), triple(x, R, z2) \rightarrow inst(z1, z2)$$

$$nom(z1), nom(z2), inst(x, C), inst(x, D), triple(x, R, z1), triple(x, R, z2) \rightarrow inst(z2, z1)$$

Briefly, it means if there are two triples  $triple(x, R, z1)$  and  $triple(x, R, z2)$ , then  $z1$  and  $z2$  must be same. (See details of translation in [5].)

Since  $A \sqsubseteq \forall R.C$  is the same as  $\exists R^-.A \sqsubseteq C$ , we can approximate  $A \sqsubseteq \forall R.C$  by adding

$$\exists inv(R).A \sqsubseteq C$$

and

$$\{x\} \sqcap \exists R.\{y\} \sqsubseteq \{y\} \sqcap \exists inv(R).\{x\}.$$

Furthermore, for each axiom  $A \sqsubseteq \leq nR.C$ , we reduce it to  $A \sqsubseteq \leq 1R.C$ , such that it can be approximated through the nominal schema axiom

$$A \sqcap \exists R.(\{x\} \sqcap C) \sqcap \exists R.(\{y\} \sqcap C) \sqsubseteq \exists U.(\{x\} \sqcap \{y\}).$$

### 3.2 Approximation of Negation and Disjunction

Our approach for approximating negation is derived from [23]. In brief, we add a fresh concept  $neg(C)$  for each concept  $C$  in KB, and add the axiom  $neg(C) \sqcap C \sqsubseteq \perp$  to express that the negation of  $C$  and  $C$  are disjoint. Furthermore, we rewrite the following axioms by using their De Morgan equivalent axioms and replace  $\neg C$  by the fresh concept  $neg(C)$ .

- (1)  $A \sqsubseteq B \sqcup C \Rightarrow \neg B \sqcap \neg C \sqsubseteq \neg A \Rightarrow neg(C) \sqsubseteq neg(A)$
- (2)  $A \sqsubseteq \forall R.C \Rightarrow \exists R.\neg C \sqsubseteq \neg A \Rightarrow \exists R.neg(C) \sqsubseteq neg(A)$
- (3)  $\forall R.A \sqsubseteq C \Rightarrow \neg C \sqsubseteq \exists R.\neg A \Rightarrow neg(C) \sqsubseteq \exists R.neg(A)$
- (4)  $\leq nR.A \sqsubseteq C \Rightarrow \neg C \sqsubseteq > nR.A \Rightarrow neg(C) \sqsubseteq > nR.A$
- (5)  $\geq nR.A \sqsubseteq C \Rightarrow \neg C \sqsubseteq < nR.A \Rightarrow neg(C) \sqsubseteq < nR.A$

**Table 2.** Evaluation ontologies for our algorithm

Ontology	Classes	Annotation P.	Data P.	Object P.
Rex <sup>3</sup>	552	10	0	6
Spatial <sup>4</sup>	106	13	0	13
Xenopus <sup>5</sup>	710	19	0	5

Note that we can always reduce  $C \sqsubseteq_{\geq} nR.A$  to  $C \sqsubseteq \exists R.A$ . Then, for the last two axioms (4) and (5), we reduce them to  $neg(C) \sqsubseteq \exists R.A$  and  $neg(C) \sqsubseteq < 1R.A$ . Following the ideas in [12,27], for  $A \sqsubseteq B \sqcup C$ , it can be reduced to  $A \sqsubseteq B \sqcap C$ , i.e.,  $A \sqsubseteq B$  and  $A \sqsubseteq C$ , falling into unsound but complete results. We will attempt to combine this idea with the approach in the paper. Briefly, combining unsound results and incomplete results to achieve higher precise and recall.

## 4 Reasoning over $\mathcal{ELROV}_n$

We briefly recall the algorithm for reasoning over  $\mathcal{ELROV}_n$  presented in [5], and the evaluation results presented therein. The algorithm actually imposes some restrictions on  $\mathcal{ELROV}_n$  which are described in detail in [5] and which cause no problem for our approximation approach.

The algorithm itself is based on results presented in [15]. Following this approach, for every  $\mathcal{ELROV}_n$  knowledge base  $KB$  we can construct a Datalog program  $P_{KB}$  that can be used for reasoning over  $KB$ . The Datalog program  $P_{KB}$  contains facts which are translated from all the DL normal forms (Figure 1) and rules (Figure 2). [5] contains a correctness proof.

The evaluation reported in [5] was performed using the Java-based Datalog reasoner IRIS<sup>2</sup> [3], and we compared it to a full grounding approach for which we also used IRIS. We used suitable ontologies from the TONES repository, see Table 2 for some basic metrics, and artificially added named individuals and axioms using nominal schemas. Results are listed in Table 3. In our approach, the number of nominal schemas per axioms had almost no effect on the runtime, thus indicating that the approach performs very well indeed.

<sup>2</sup> <http://iris-reasoner.org/>

<sup>3</sup> <http://obo.cvs.sourceforge.net/checkout/obo/obo/ontology/physicochemical/rex.obo>

<sup>4</sup> <http://obo.cvs.sourceforge.net/checkout/obo/obo/ontology/anatomy/caro/spatial.obo>

<sup>5</sup> [http://obo.cvs.sourceforge.net/checkout/obo/obo/ontology/anatomy/gross\\_anatomy/animal\\_gross\\_anatomy/frog/xenopus\\_anatomy.obo](http://obo.cvs.sourceforge.net/checkout/obo/obo/ontology/anatomy/gross_anatomy/animal_gross_anatomy/frog/xenopus_anatomy.obo)

**Table 3.** Evaluation, IRIS reasoning time listed only (no pre-processing, no load time), in ms. The "No ns" column refers to the running with no nominal schemas, while  $k$  ns refers to the use of  $k$  nominal schemas in an axiom. Times in brackets are for full grounding, for comparison. If not listed, full grounding was OOM (Out of Memory)

Ontology	number of individuals	no ns	1 ns	2 ns	3 ns	4 ns	5 ns
Rex	100	263	263 (321)	267 (972)	273	275	259
	1000	480	518 (1753)	537 (OOM)	538	545	552
	10000	2904	2901 (133179)	3120 (OOM)	3165	3192	3296
Spatial	100	22	191 (222)	201 (1163)	198	202	207
	1000	134	417 (1392)	415 (OOM)	421	431	432
	10000	1322	1792 (96437)	1817 (OOM)	1915	1888	1997
Xenopus	100	62	332 (383)	284 (1629)	311	288	280
	1000	193	538 (4751)	440 (OOM)	430	456	475
	10000	1771	2119 (319013)	1843 (OOM)	1886	2038	2102

## 5 Implementation and Evaluation

We realized the implementation based on the  $\mathcal{ELR}\mathcal{O}\mathcal{V}_n$  datalog-based reasoner [5]. All experiments were conducted on a laptop with a 2.4GHz Intel Core™ i7-3630QM processor and 8GB RAM operated by Windows 7 64-bit system with Java VM v.1.7.0. We set time out of 1 hour and Java heap space of 1GB. The ontologies were chose from Oxford Ontologies Repository <sup>6</sup>, in Table Table 4. To evaluate the performance in practice, we also compared with mainstream reasoners Pellet 2.3.0<sup>7</sup>, FaCT++ 1.6.2<sup>8</sup> and HermiT 1.3.7<sup>9</sup>. The reasoning task is classification, therefore recall equals the number of subsumption relations between concepts divides its correct number. Since our approach needs some individual to fire the datalog rules, we add one unique dummy individual for each concepts if the testing ontology does not contain individuals. Therefore, we can check subsumption relations by tracking those dummy individuals.

The experiment, Table 5 , shows our approach has good recalls but fails when conducting very large ontologies. The reason is that IRIS reasoner has a difficulty to run with large number of rules or facts. However, with a quicker datalog reasoner or a more efficient reasoner that supports nominal schemas, we believe it will achieve a better result. Also, since the number of rules (Figure 2) are fixed, we do not need a full powerful Datalog reasoner. We can specifically program the rules to improve the efficiency.

To be noticed, the approximation in this paper can be done by HermiT reasoner since HermiT can handle *DL-safe* rules and the rules can directly be

<sup>6</sup> <http://www.cs.ox.ac.uk/isg/ontologies/>

<sup>7</sup> <http://clarkparsia.com/pellet/>

<sup>8</sup> <http://owl.man.ac.uk/factplusplus/>

<sup>9</sup> <http://www.hermit-reasoner.com/>

$C(a) \mapsto \{\text{subClass}(a, D)\}$	$R(a, b) \mapsto \{\text{subEx}(a, R, b, b)\}$
$\top \sqsubseteq C \mapsto \{\text{top}(C)\}$	$A \sqsubseteq \perp \mapsto \{\text{bot}(A)\}$
$\{a\} \sqsubseteq C \mapsto \{\text{subClass}(a, C)\}$	$A \sqsubseteq \{c\} \mapsto \{\text{subClass}(A, c)\}$
$A \sqsubseteq C \mapsto \{\text{subclass}(A, C)\}$	$A \sqcap B \sqsubseteq C \mapsto \{\text{subConj}(A, B, C)\}$
$\exists R.\text{Self} \sqsubseteq C \mapsto \{\text{subSelf}(R, C)\}$	$A \sqsubseteq \exists R.\text{Self} \mapsto \{\text{supSelf}(A, R)\}$
$\exists R.A \sqsubseteq C \mapsto \{\text{subEx}(R, A, C)\}$	$A \sqsubseteq \exists R.C \mapsto \{\text{supEx}(A, R, B, \text{aux}_{A \sqsubseteq \exists R.C})\}$
$R \sqsubseteq T \mapsto \{\text{subRole}(R, T)\}$	$R \circ S \sqsubseteq T \mapsto \{\text{subRChain}(R, S, T)\}$
$R \sqsubseteq C \times D \mapsto \{\text{supProd}(R, C, D)\}$	$A \in N_C \mapsto \{\text{cls}(A)\}$
$R \in N_R \mapsto \{\text{rol}(R)\}$	$a \in N_I \mapsto \{\text{nom}(a)\}$

**Fig. 1.** Input Translation  $I_{\gamma}$

**Table 4.** Evaluation ontologies for our algorithm, the No. denotes the number order of Oxford Repository. Since the implementation does not support datatype property, any ontologies containing datatype properties are not chosen here.

No.	Ontology	expressivity	Classes	Object Properties	Individuals	TBox	RBox	ABox
00004	BAMS	<i>SHIF</i>	1110	12	0	18813	9	0
00015	DOLCE	<i>SHI</i>	37	70	0	279	76	0
00039	GALEN	<i>ALSHIF</i> <sup>+</sup>	3097	403	0	14661	111	0
00040	GO	<i>SRIQ</i>	58882	220	0	130376	104	0
00103	Gardiner	<i>ALLO</i>	176	10	187	263	0	187
00387	OBO	<i>SRLF</i>	27412	215	34	55998	87	157897

added to the input ontology in functional style. But, HerMiT doesn't have specific reasoning procedure for  $\mathcal{EL}$ -families, such that reasoning for  $\mathcal{EL}$  is not its advantage. Moreover, there are  $\mathcal{ELROV}_n$  axioms which cannot be expressed as *DL-safe* rules, e.g.,  $\exists R.\{z\} \sqsubseteq \exists T.\exists S.\{z\}$ . Moreover,

## 6 Conclusions and Future Work

We have described an approximate reasoning procedure for *SROIQ* which utilizes the tractable nominal-schemas-based  $\mathcal{ELROV}_n$  using a language weakening approach. We have also provided an experimental evaluation which shows the feasibility of this setting.

Going forward, there are several directions which we intend to explore. On the one hand, we will be looking into variants on how to obtain the weakened language, in the spirit of [27], and will attempt to further tweak and optimize our approach. On the one hand, we will be looking into incremental methods



$$\begin{aligned}
& \text{nom}(x) \mapsto \text{inst}(x, x) & (1) \\
& \text{nom}(x) \wedge \text{triple}(x, v, x) \mapsto \text{self}(x, v) & (2) \\
& \text{top}(z) \wedge \text{inst}(x, z') \mapsto \text{inst}(x, z) & (3) \\
& \text{bot}(z) \wedge \text{inst}(u, z) \wedge \text{inst}(x, z') \wedge \text{cls}(y) \mapsto \text{inst}(x, y) & (4) \\
& \text{subClass}(y, z) \wedge \text{inst}(x, y) \mapsto \text{inst}(x, z) & (5) \\
& \text{subConj}(y_1, y_2, z) \wedge \text{inst}(x, y_1) \wedge \text{inst}(x, y_2) \mapsto \text{inst}(x, z) & (6) \\
& \text{subEx}(v, y, z) \wedge \text{triple}(x, v, x') \wedge \text{inst}(x', y) \mapsto \text{inst}(x, z) & (7) \\
& \text{subEx}(v, y, z) \wedge \text{self}(x, v) \wedge \text{inst}(x, y) \mapsto \text{inst}(x, z) & (8) \\
& \text{supEx}(y, v, z, x') \wedge \text{inst}(x, y) \mapsto \text{triple}(x, v, x') & (9) \\
& \text{supEx}(y, v, z, x') \wedge \text{inst}(x, y) \mapsto \text{inst}(x', z) & (10) \\
& \text{subSelf}(v, z) \wedge \text{self}(x, v) \mapsto \text{inst}(x, z) & (11) \\
& \text{supSelf}(y, v) \wedge \text{inst}(x, y) \mapsto \text{self}(x, v) & (12) \\
& \text{subRole}(v, w) \wedge \text{triple}(x, v, x') \mapsto \text{triple}(x, w, x') & (13) \\
& \text{subRole}(v, w) \wedge \text{self}(x, v) \mapsto \text{self}(x, w) & (14) \\
& \text{subRChain}(u, v, w) \wedge \text{triple}(x, u, x') \wedge \text{triple}(x', v, x'') \mapsto \text{triple}(x, w, x'') & (15) \\
& \text{subRChain}(u, v, w) \wedge \text{self}(x, y) \wedge \text{triple}(x, v, x') \mapsto \text{triple}(x, w, x') & (16) \\
& \text{subRChain}(u, v, w) \wedge \text{triple}(x, u, x') \wedge \text{self}(x', v) \mapsto \text{triple}(x, w, x') & (17) \\
& \text{subRChain}(u, v, w) \wedge \text{self}(x, u) \wedge \text{self}(x, v) \mapsto \text{triple}(x, w, x) & (18) \\
& \text{supProd}(v, z_1, z_2) \wedge \text{triple}(x, v, x') \mapsto \text{inst}(x, z_1) & (19) \\
& \text{supProd}(v, z_1, z_2) \wedge \text{self}(x, v) \mapsto \text{inst}(x, z_1) & (20) \\
& \text{supProd}(v, z_1, z_2) \wedge \text{triple}(x, v, x') \mapsto \text{inst}(x', z_2) & (21) \\
& \text{supProd}(v, z_1, z_2) \wedge \text{self}(x, v) \mapsto \text{inst}(x, z_2) & (22) \\
& \text{inst}(x, y) \wedge \text{nom}(y) \wedge \text{inst}(x, z) \mapsto \text{inst}(y, z) & (23) \\
& \text{inst}(x, y) \wedge \text{nom}(y) \wedge \text{inst}(y, z) \mapsto \text{inst}(x, z) & (24) \\
& \text{inst}(x, y) \wedge \text{nom}(y) \wedge \text{triple}(z, u, x) \mapsto \text{triple}(z, u, y) & (25) \\
& \text{self}(x, y) \mapsto \text{triple}(x, y, x) & (26)
\end{aligned}$$

**Fig. 2.** Deduction Rules  $P_V$

**Table 5.** Evaluation, reasoning time of each reasoner, in ms. N/A denotes that the datalog-based reasoner corrupts with too many loading rules.

Ontology	HermiT	Fact++	Pellet	Ours	Ours Recall
BAMS	3	2	10	107	100%
DOLCE	1	1	4	53	100%
GALEN	4	2	17	7840	90.8%
GO	36	75	59	N/A	N/A
GardinerCorpus	14	6	17	89	92.3%
OBO	34	61	139	N/A	N/A

which use the approximate reasoning results as starting point and subsequently compute correct results in all or at least most cases.

*Acknowledgements* This work was supported by the National Science Foundation under award 1017225 III: Small: TROn – Tractable Reasoning with Ontologies.

## References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05). pp. 364–369. Morgan-Kaufmann Publishers, Edinburgh, UK (2005)
3. Bishop, B., Fischer, F.: IRIS – Integrated Rule Inference System. In: van Harmelen, F., Herzig, A., Hitzler, P., Lin, Z., Piskac, R., Qi, G. (eds.) ARea2008, Workshop on Advancing Reasoning on the Web: Scalability and Commonsense. Proceedings of the Workshop on Advancing Reasoning on the Web: Scalability and Commonsense Tenerife, Spain, June 2, 2008. CEUR Wrokshop Proceedings, vol. 350. CEUR-WS.org (2008)
4. Cadoli, M., Schaerf, M.: Approximate inference in default reasoning and circumscription. *Fundamenta Informaticae* 23, 123–143 (1995)
5. Carral, D., Wang, C., Hitzler, P.: Towards an efficient algorithm to reason over description logics extended with nominal schemas. Tech. rep., Kno.e.sis Center, Wright State University (2013), available at <http://www.pascal-hitzler.de/pub/elrov13.pdf>
6. Fensel, D., van Harmelen, F.: Unifying reasoning and search to web scale. *IEEE Internet Computing* 11(2), 96, 94–95 (2007)
7. Grau, B.C., Motik, B., Stoilos, G., Horrocks, I.: Completeness guarantees for incomplete ontology reasoners: Theory and practice. *J. Artif. Intell. Res. (JAIR)* 43, 419–476 (2012)
8. Groot, P., Stuckenschmidt, H., Wache, H.: Approximating description logic classification for semantic web reasoning. In: Gómez-Pérez, A., Euzenat, J. (eds.) The

- Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 – June 1, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3532, pp. 318–332. Springer (2005)
9. Hitzler, P.: Towards reasoning pragmatics. In: Janowicz, K., Raubal, M., Levashkin, S. (eds.) GeoSpatial Semantics, Third International Conference, GeoS 2009, Mexico City, Mexico, December 3–4, 2009. Proceedings. pp. 9–25. Lecture Notes in Computer Science, Springer (2009)
  10. Hitzler, P., van Harmelen, F.: A reasonable semantic web. *Semantic Web* 1(1–2), 39–44 (2010)
  11. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation (27 October 2009), available at <http://www.w3.org/TR/owl2-primer/>
  12. Hitzler, P., Vrandečić, D.: Resolution-based approximate reasoning for OWL DL. In: Gil, Y., et al. (eds.) Proceedings of the 4th International Semantic Web Conference, Galway, Ireland, November 2005. Lecture Notes in Computer Science, vol. 3729, pp. 383–397. Springer, Berlin (2005)
  13. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible *SROIQ*. In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006). pp. 57–67. AAAI Press (2006)
  14. Knorr, M., Hitzler, P., Maier, F.: Reconciling OWL and non-monotonic rules for the Semantic Web. In: De Raedt, L., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P. (eds.) ECAI 2012, 20th European Conference on Artificial Intelligence, 27–31 August 2012, Montpellier, France. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 474–479. IOS Press, Amsterdam (2012)
  15. Krötzsch, M.: Efficient inferencing for OWL EL. In: Janhunen, T., Niemelä, I. (eds.) Proc. 12th European Conf. on Logics in Artificial Intelligence (JELIA’10). LNAI, vol. 6341, pp. 234–246. Springer (2010)
  16. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable rules for OWL 2. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) Proceedings of the 7th International Semantic Web Conference (ISWC-08). Lecture Notes in Computer Science, vol. 5318, pp. 649–664. Springer (2008)
  17. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Proceedings of the 20th International Conference on World Wide Web (WWW’11). pp. 645–654. ACM (2011)
  18. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (27 October 2009), available at <http://www.w3.org/TR/owl2-profiles/>
  19. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. *J. of Artificial Intelligence Research* 36(1), 165–228 (2009)
  20. Pan, J.Z., Thomas, E.: Approximating OWL-DL ontologies. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22–26, 2007, Vancouver, British Columbia, Canada. pp. 1434–1439. AAAI Press (2007)
  21. Pan, J.Z., Thomas, E., Zhao, Y.: Completeness guaranteed approximations for OWL-DL query answering. In: Grau, B.C., Horrocks, I., Motik, B., Sattler, U. (eds.) Proceedings of the DL Home 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27–30, 2009. CEUR Workshop Proceedings, vol. 477 (2009)

22. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for TBox reasoning in R. In: Grau, B.C., Horrocks, I., Motik, B., Sattler, U. (eds.) Proceedings of the DL Home 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009. CEUR Workshop Proceedings, vol. 477 (2009)
23. Ren, Y., Pan, J.Z., Zhao, Y.: Towards soundness preserving approximation for abox reasoning of owl2. In: Haarslev, V., Toman, D., Weddell, G.E. (eds.) Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010. CEUR Workshop Proceedings, vol. 573. CEUR-WS.org (2010)
24. Rudolph, S., Tserendorj, T., Hitzler, P.: What is approximate reasoning? In: Calvanese, D., Lausen, G. (eds.) Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems (RR2008). Lecture Notes in Computer Science, vol. 5341, pp. 150–164. Springer (2008)
25. Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. *Artificial Intelligence* 74, 249–310 (1995)
26. Selman, B., Kautz, H.A.: Knowledge compilation using Horn approximations. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91). pp. 904–909 (1991)
27. Tserendorj, T., Rudolph, S., Krötzsch, M., Hitzler, P.: Approximate OWL-reasoning with Screech. In: Calvanese, D., Lausen, G. (eds.) Web Reasoning and Rule Systems, Second International Conference, RR 2008, Karlsruhe, Germany, October 31–November 1, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5341, pp. 165–180 (2008)