Description Logic Rules

Markus Krötzsch and **Sebastian Rudolph** and **Pascal Hitzler**[1]

Abstract. We introduce *description logic (DL) rules* as a new rule-based formalism for knowledge representation in DLs. As a fragment of the Semantic Web Rule Language SWRL, DL rules allow for a tight integration with DL knowledge bases. In contrast to SWRL, however, the combination of DL rules with expressive description logics remains decidable, and we show that the DL $\mathcal{SROIQ}$ – the basis for the ongoing standardisation of OWL 2 – can completely internalise DL rules. On the other hand, DL rules capture many expressive features of $\mathcal{SROIQ}$ that are not available in simpler DLs yet. While reasoning in $\mathcal{SROIQ}$ is highly intractable, it turns out that DL rules can be introduced to various lightweight DLs without increasing their worst-case complexity. In particular, DL rules enable us to significantly extend the tractable DLs $\mathcal{EL}^{++}$ and DLP.

1 INTRODUCTION

The development of description logics (DLs) has been driven by the desire to push the expressivity bounds of these knowledge representation formalisms while still maintaining decidability and implementability. This has lead to very expressive DLs such as $\mathcal{SHOIN}$, the logic underlying the Web Ontology Language OWL DL, $\mathcal{SHOIQ}$, and more recently $\mathcal{SROIQ}$ [6] which is the basis for the ongoing standardisation of OWL 2[2] as the next version of the Web Ontology Language. On the other hand, more lightweight DLs for which most common reasoning problems can be implemented in (sub)polynomial time have also been sought, leading, e.g., to the tractable DL $\mathcal{EL}^{++}$ [2].

Another popular paradigm of knowledge representation are rule-based formalisms – ranging from logic programming to deductive databases. Similar to DLs, the expressivity and complexity of rule languages has been studied extensively [4], and many decidable and tractable formalisms are known. Yet, reconciling DLs and rule languages is not easy, and many works have investigated this problem.

In this paper, we introduce *DL rules* as an expressive new rule language for combining DLs with first-order rules in a rather natural way that admits tight integration with existing DL systems. Since DLs can be considered as fragments of function-free first-order logic with equality, an obvious approach is to combine them with first-order Horn-logic rules. This is the basis of the *Semantic Web Rule Language SWRL* [7], proposed as a rule extension to OWL. However, reasoning becomes undecidable for the combination of OWL and SWRL, and thus more restricted rule languages have been investigated. A prominent example are *DL-safe rules* [9], which restrict the applicability of rules to a finite set of named individuals to retain decidability. Another basic approach is to identify the Horn-logic rules directly expressible in OWL DL (i.e. $\mathcal{SHOIN}$), and this fragment has been called *Description Logic Programs DLP* [5].

DL rules can be characterised as a decidable fragment of SWRL, which corresponds to a large class of SWRL rules indirectly expressible in $\mathcal{SROIQ}$. They are based on the observation that DLs can express only tree-like interdependencies of variables. For example, the concept expression $\exists\,\mathsf{worksAt}.\mathsf{University} \sqcap \exists\,\mathsf{supervises}.\mathsf{PhDStudent}$ that describes all people working at a university and supervising some PhD student corresponds to the following first-order formula:

$$\exists y.\exists z.\mathsf{worksAt}(x,y) \wedge \mathsf{University}(y) \wedge \mathsf{supervises}(x,z) \wedge \mathsf{PhDStudent}(z)$$

Here variables form the nodes of a tree with root x, where edges are given by binary predicates. Intuitively, DL rules are exactly those SWRL rules, where premises (rule bodies) consist of one or more of such tree-shaped structures. One could, for example, formulate the following rule:

$$\mathsf{worksAt}(x,y) \wedge \mathsf{University}(y) \wedge \mathsf{supervises}(x,z) \wedge \mathsf{PhDStudent}(z) \rightarrow \mathsf{profOf}(x,z)$$

Since SWRL allows the use of DL concept expressions in rules, we obtain $\mathcal{SROIQ}$ rules, $\mathcal{EL}^{++}$ rules, or DLP rules as extensions of the respective DLs. It turns out that DL rules are indeed "syntactic sugar" in $\mathcal{SROIQ}$, but we argue that a rule-based presentation is significantly simpler in many cases, in particular since rules as the above require the introduction of auxiliary vocabulary to be expressed in $\mathcal{SROIQ}$. On the other hand, DL rules truly extend the expressivity in many simpler DLs, and we show that the favourable complexity properties of the light-weight DLs $\mathcal{EL}^{++}$ and DLP are preserved when adding DL rules.

After providing preliminary definitions in Section 2, we introduce DL rules in Section 3. Section 4 shows how DL rules can be internalised in $\mathcal{SROIQ}$, while Section 5 employs a novel reasoning algorithm to process $\mathcal{EL}^{++}$ rules directly. Finally, Section 6 introduces DLP 2 and shows the tractability of this DL-based rule language. Proofs are omitted for reasons of space; they can be found in [1].

2 PRELIMINARIES

In this section, we recall the definition of the expressive description logic $\mathcal{SROIQ}$ [6]. We assume that the reader is familiar with description logics [3]. As usual, the DLs considered in this paper are based on three disjoint sets of *individual names* N_I, *concept names* N_C, and *role names* N_R containing the *universal role* $U \in \mathsf{N}_R$.

Definition 1 *A $\mathcal{SROIQ}$ Rbox for N_R is based on a set* $\mathbf{R}$ *of* atomic roles *defined as* $\mathbf{R} := \mathsf{N}_R \cup \{R^- \mid R \in \mathsf{N}_R\}$*, where we set* $\mathrm{Inv}(R) := R^-$ *and* $\mathrm{Inv}(R^-) := R$ *to simplify notation. In the sequel, we will use the symbols R, S, possibly with subscripts, to denote atomic roles.*

A generalised role inclusion axiom *(RIA) is a statement of the form $S_1 \circ \ldots \circ S_n \sqsubseteq R$, and a set of such RIAs is a generalised* role hierar-

[1] Universität Karlsruhe (TH), Germany, [mak|sru|phi]@aifb.uni-karlsruhe.de

[2] OWL 2 is the forthcoming W3C recommendation updating OWL, based on the OWL 1.1 member submission, cf. `http://www.w3.org/2007/OWL`.

Syntax	Semantics
R^-	$\{\langle x,y\rangle \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid \langle y,x\rangle \in R^\mathcal{I}\}$
U	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
$\top$	$\Delta^\mathcal{I}$
$\bot$	$\emptyset$
$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
$\{a\}$	$\{a^\mathcal{I}\}$
$\forall R.C$	$\{x \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in R^\mathcal{I}$ implies $y \in C^\mathcal{I}\}$
$\exists R.C$	$\{x \in \Delta^\mathcal{I} \mid$ for some $y \in \Delta^\mathcal{I}$, $\langle x,y\rangle \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\}$
$\exists S.\mathsf{Self}$	$\{x \in \Delta^\mathcal{I} \mid \langle x,x\rangle \in S^\mathcal{I}\}$
$\leq n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \leq n\}$
$\geq n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \geq n\}$

Figure 1. Semantics of concept constructors in $\mathcal{SROIQ}$ for an interpretation $\mathcal{I}$ with domain $\Delta^\mathcal{I}$.

chy. *A role hierarchy is* regular *if there is a strict partial order* $\prec$ *on* $\mathbf{R}$ *such that*

- $S \prec R$ *iff* $\mathrm{Inv}(S) \prec R$, *and*
- *every RIA is of one of the forms:*

$$R \circ R \sqsubseteq R, \qquad R^- \sqsubseteq R, \qquad S_1 \circ \ldots \circ S_n \sqsubseteq R,$$
$$R \circ S_1 \circ \ldots \circ S_n \sqsubseteq R, \qquad S_1 \circ \ldots \circ S_n \circ R \sqsubseteq R$$

such that $R \in \mathsf{N}_R$ *is a (non-inverse) role name, and* $S_i \prec R$ *for* $i = 1, \ldots, n$.

The set of simple roles *for some role hierarchy is defined inductively:*

- *If a role R occurs only on the right-hand-side of RIAs of the form* $S \sqsubseteq R$ *such that S is simple, then R is also simple.*
- *The inverse of a simple role is simple.*

A role assertion *is a statement of the form* $\mathsf{Ref}(R)$ *(*reflexivity*),* $\mathsf{Asy}(S)$ *(*asymmetry*), or* $\mathsf{Dis}(S, S')$ *(*role disjointness*), where S and S′ are simple. A* $\mathcal{SROIQ}$ Rbox *is the union of a set of role assertions together and a role hierarchy. A* $\mathcal{SROIQ}$ *Rbox is regular if its role hierarchy is regular.*

Definition 2 *Given a* $\mathcal{SROIQ}$ *Rbox* $\mathcal{R}$*, the set of* concept expressions $\mathbf{C}$ *is defined as follows:*

- $\mathsf{N}_C \subseteq \mathbf{C}$, $\top \in \mathbf{C}$, $\bot \in \mathbf{C}$,
- *if* $C, D \in \mathbf{C}$, $R \in \mathbf{R}$, $S \in \mathbf{R}$ *a simple role,* $a \in \mathsf{N}_I$*, and n a non-negative integer, then* $\neg C$, $C \sqcap D$, $C \sqcup D$, $\{a\}$, $\forall R.C$, $\exists R.C$, $\exists S.\mathsf{Self}$, $\leq n\, S.C$, *and* $\geq n\, S.C$ *are also concept expressions.*

Throughout this paper, the symbols C, D will be used to denote concept expressions. A $\mathcal{SROIQ}$ Tbox *is a set of* general concept inclusion axioms *(GCIs) of the form* $C \sqsubseteq D$.

An individual assertion *can have any of the following forms:* $C(a)$, $R(a,b)$, $\neg R(a,b)$, $a \not\approx b$, *with* $a, b \in \mathsf{N}_I$ *individual names,* $C \in \mathbf{C}$ *a concept expression, and* $R, S \in \mathbf{R}$ *roles with S simple. A* $\mathcal{SROIQ}$ Abox *is a set of individual assertions.*

A $\mathcal{SROIQ}$ knowledge base KB *is the union of a regular Rbox* $\mathcal{R}$*, and an Abox* $\mathcal{A}$ *and Tbox* $\mathcal{T}$ *for* $\mathcal{R}$.

We further recall the semantics of $\mathcal{SROIQ}$ knowledge bases.

Definition 3 *An interpretation* $\mathcal{I}$ *consists of a set* $\Delta^\mathcal{I}$ *called* domain *(the elements of it being called* individuals*) together with a function* $\cdot^\mathcal{I}$ *mapping*

- *individual names to elements of* $\Delta^\mathcal{I}$,
- *concept names to subsets of* $\Delta^\mathcal{I}$, *and*
- *role names to subsets of* $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

The function $\cdot^\mathcal{I}$ *is inductively extended to role and concept expressions as shown in Table 1. An interpretation* $\mathcal{I}$ satisfies *an axiom* φ *if we find that* $\mathcal{I} \models \varphi$*:*

- $\mathcal{I} \models S \sqsubseteq R$ *if* $S^\mathcal{I} \subseteq R^\mathcal{I}$,
- $\mathcal{I} \models S_1 \circ \ldots \circ S_n \sqsubseteq R$ *if* $S_1^\mathcal{I} \circ \ldots \circ S_n^\mathcal{I} \sqsubseteq R^\mathcal{I}$ *(*$\circ$ *being overloaded to denote the standard composition of binary relations here),*
- $\mathcal{I} \models \mathsf{Ref}(R)$ *if* $R^\mathcal{I}$ *is a reflexive relation,*
- $\mathcal{I} \models \mathsf{Asy}(R)$ *if* $R^\mathcal{I}$ *is antisymmetric and irreflexive,*
- $\mathcal{I} \models \mathsf{Dis}(R, S)$ *if* $R^\mathcal{I}$ *and* $S^\mathcal{I}$ *are disjoint,*
- $\mathcal{I} \models C \sqsubseteq D$ *if* $C^\mathcal{I} \subseteq D^\mathcal{I}$.

An interpretation $\mathcal{I}$ satisfies *a knowledge base* KB *(we then also say that* $\mathcal{I}$ *is a* model *of* KB *and write* $\mathcal{I} \models$ KB*) if it satisfies all axioms of* KB. *A knowledge base* KB *is* satisfiable *if it has a model. Two knowledge bases are* equivalent *if they have exactly the same models, and they are* equisatisfiable *if either both are unsatisfiable or both are satisfiable.*

Further details on $\mathcal{SROIQ}$ can be found in [6]. We have omitted here several syntactic constructs that can be expressed indirectly, especially role assertions for transitivity, reflexivity of simple roles, and symmetry.

3 DESCRIPTION LOGIC RULES

In this section, we formally introduce *DL rules* as a syntactic fragment of first-order logic.

Definition 4 *Consider some description logic* $\mathcal{L}$ *with concept expressions* $\mathbf{C}$*, individual names* N_I*, atomic roles* $\mathbf{R}$ *(possibly including inverse roles), and let* $\mathbf{V}$ *be a countable set of first-order variables. Given terms* $t, u \in \mathsf{N}_I \cup \mathbf{V}$*, a* concept atom (role atom) *is a formula of the form* $C(t)$ $(R(t,u))$ *with* $C \in \mathbf{C}$ $(R \in \mathbf{R})$.

To simplify notation, we will often use finite sets S of (role and concept) atoms for representing the conjunction $\bigwedge S$. *Given such a set S of atoms and terms* $t, u \in \mathsf{N}_I \cup \mathbf{V}$*, a* path *from t to u in S is a non-empty sequence* $R_1(x_1, x_2), \ldots, R_n(x_n, x_{n+1}) \in S$ *where* $x_1 = t$, $x_i \in \mathbf{V}$ *for* $2 \leq i \leq n$, $x_{n+1} = u$, *and* $x_i \neq x_{i+1}$ *for* $1 \leq i \leq n$. *A term t in S is* initial *(resp.* final*) if there is no path to t (resp. no path starting at t).*

Given sets B and H of atoms, and a set $\mathbf{x} \subseteq \mathbf{V}$ *of all variables in* $B \cup H$*, a* description logic rule (DL rule) *is a formula* $\forall \mathbf{x}. \bigwedge B \rightarrow \bigwedge H$ *such that*

(R1) for any $u \in \mathsf{N}_I \cup \mathbf{V}$ *that is not initial in B, there is a path from exactly one initial* $t \in \mathsf{N}_I \cup \mathbf{V}$ *to u in B,*
(R2) for any $t, u \in \mathsf{N}_I \cup \mathbf{V}$*, there is at most one path in B from t to u,*
(R3) if H contains an atom C(t) or R(t, u), then t is initial in B.

Here $\forall \mathbf{x}$ *for* $\mathbf{x} = \{x_1, \ldots, x_n\}$ *abbreviates an arbitrary sequence* $\forall x_1 \ldots \forall x_n$. *Since we consider only conjunctions with all variables quantified, we will often simply write* $B \rightarrow H$ *instead of* $\forall \mathbf{x}. \bigwedge B \rightarrow \bigwedge H$. *A* rule base RB *for some DL* $\mathcal{R}$ *is a set of DL rules for* $\mathcal{R}$.

The semantics of DL rules in the context of a description logic knowledge base is given by interpreting both the rules and knowledge base as first-order theories in the usual way, and applying the

standard semantics of predicate logic. This has been discussed in the context of SWRL in [7], and we will not repeat the details here.

Note that Definition 4 ensures that role atoms in rule bodies essentially form a (set of) directed trees, starting at initial elements. Using the well-known equivalence of formulae $\{p \rightarrow q_1 \wedge q_2\}$ and $\{p \rightarrow q_1, p \rightarrow q_2\}$, one can transform any rule into an equivalent set of rules without conjunctions in rule heads. Since this can be done in linear time, we will assume without loss of generality that all DL rules are of this form.

Moreover, since all DLs considered in this work support nominals, we will assume without loss of generality that all terms in rules are variables. Indeed, any atom $C(a)$ with $a \in \mathsf{N}_I$ can be replaced by $C(x) \wedge \{a\}(x)$ for some new variable $x \in \mathbf{V}$. In the presence of inverse roles, role atoms with individual names can be replaced by concept atoms as follows: $R(x, a)$ becomes $\exists R.\{a\}(x)$, $R(a, y)$ becomes $\exists \mathrm{Inv}(R).\{a\}(y)$, and $R(a, b)$ becomes $\exists R.\{a\}(x) \wedge \{b\}(x)$. A similar transformation is possible for rule heads, where generated concept atoms $\{a\}(x)$ are again added *to the rule body*.

Before considering the treatment of DL rules in concrete DLs, we highlight some relevant special applications of DL rules.

Concept products Rules of the form $C(x) \wedge D(y) \rightarrow R(x, y)$ can encode *concept products* (sometimes written $C \times D \sqsubseteq R$) asserting that all elements of two classes must be related [10]. Examples include statements such as $\mathsf{Elephant}(x) \wedge \mathsf{Mouse}(y) \rightarrow \mathsf{biggerThan}(x, y)$ or $\mathsf{Alkaline}(x) \wedge \mathsf{Acid}(y) \rightarrow \mathsf{neutralises}(x, y)$.

Local reflexivity, universal role Rules of the forms $C(x) \rightarrow R(x, x)$ and $R(x, x) \rightarrow C(x)$ can replace the $\mathcal{SROIQ}$ Tbox expression $C \sqsubseteq \exists R.\mathsf{Self}$ and $\exists R.\mathsf{Self} \sqsubseteq C$. The universal role U of $\mathcal{SROIQ}$ can be defined as $\top(x) \wedge \top(y) \rightarrow U(x, y)$. Hence, a DL that permits such rules does not need to explicitly introduce those constructs.

Qualified RIAs DL rules of course can express arbitrary role inclusion axioms, but they also can state that a RIA applies only to instances of certain classes. Examples include $\mathsf{Woman}(x) \wedge \mathsf{hasChild}(x, y) \rightarrow \mathsf{motherOf}(x, y)$ and $\mathsf{trusts}(x, y) \wedge \mathsf{Doctor}(y) \wedge \mathsf{recommends}(y, z) \wedge \mathsf{Medicine}(z) \rightarrow \mathsf{buys}(x, z)$.

4 DL RULES IN $\mathcal{SROIQ}$

In this section, we show how knowledge bases of such rules can be completely internalised into the DL $\mathcal{SROIQ}$. Since $\mathcal{SROIQ}$ supports inverse roles, it turns out that one can relax condition (R1) of DL rules as follows:

(R1') for any $u \in \mathsf{N}_I \cup \mathbf{V}$ that is not initial in B, there is a path from one *or more* initial elements $t \in \mathsf{N}_I \cup \mathbf{V}$ to u in B.

On the other hand, we need to adopt the notions of *regularity* and *simplicity* to DL rule bases in $\mathcal{SROIQ}$, which again restricts the permissible rule bases:

Definition 5 *Consider a rule base* RB *and a knowledge base* KB *for* $\mathcal{SROIQ}$*. The set of* simple roles *of* KB ∪ RB *is the smallest set of atomic roles containing every role R for which the following conditions hold:*

- *If R or* $\mathrm{Inv}(R)$ *occur on the right-hand-side of some RIA of* KB*, then this RIA is of the form* $S \sqsubseteq R$ *or* $S \sqsubseteq \mathrm{Inv}(R)$*, and S is simple.*
- *If R or* $\mathrm{Inv}(R)$ *occur in some rule head of the form* $R(x, y)$ *or* $\mathrm{Inv}(R)(x, y)$ *in* RB*, then the according rule body is of the form* $S(x, y)$ *with S simple, or of the form* $C(x)$ *where* $x = y$*.*

Note that this is indeed a proper inductive definition, where roles that do not occur on the right of either RIAs or rules form the base case. The extended knowledge base KB ∪ RB *is* admissible *for* $\mathcal{SROIQ}$ *if all roles* $S_{(i)}$ *occurring in concept (sub)expressions of the form* $\leq n\, S.C$, $\geq n\, S.C$, $\exists S.\mathsf{Self}$, *and* $\mathsf{Dis}(S_1, S_2)$*, and in role atoms of the form* $S(x, x)$ $(x \in \mathbf{V})$ *are simple.*

An extended knowledge base KB ∪ RB *is* regular *if there is a strict partial order* $\prec$ *on* $\mathbf{R}$ *such that*

- $S \prec R$ *iff* $\mathrm{Inv}(S) \prec R$*,*
- *the role box of* KB *is regular w.r.t.* $\prec$*, and*
- *for any rule* $B \rightarrow R(x, y)$*, each* $S(z, v) \in B$ *satisfies one of the following:*
  - $S \prec R$*, or*
  - *there is no path from v to y, or*
  - $S = R$*, there is no other* $R(z', v') \in B$ *with a path from* v' *to y, and we find that: either* $x = z$ *and there is no* $C(x) \in B$*, or* $y = v$ *and there is no* $C(y) \in B$*.*

Note that RIAs in regular $\mathcal{SROIQ}$ knowledge bases are allowed to have two special forms for transitivity and symmetry, which we do omit for the definition of regularity in DL rules to simplify notation. Since S in $S(x, x)$ is simple, we can replace such role atoms by concept atoms $C(x)$ where C is a new concept name for which a new axiom $C \equiv \exists S.\mathsf{Self}$ is added. We will thus assume that no role atoms of this form occur in admissible knowledge bases.

One can now show that checking the satisfiability of extended $\mathcal{SROIQ}$ knowledge bases that are admissible and regular is decidable, and has the same worst-case complexity as reasoning in $\mathcal{SROIQ}$. This is achieved by a polynomial transformation of rule bases into $\mathcal{SROIQ}$ axioms. The first step of doing this is to replace "dead branches" of the tree-shaped query body by DL concepts.

Lemma 6 *Any DL rule* $B \rightarrow H$ *for* $\mathcal{SROIQ}$ *can be transformed into a semantically equivalent rule* $B' \rightarrow H$ *such that all paths in* B' *are contained in a single maximal path. If* $H = R(x, y)$*, then y is the final element of that maximal path, and if* $H = C(x)$ *then there are no paths in B. A rule with these properties is called* linearised*.*

As an example, the DL rule that was given in the introduction can be simplified to yield (using " , " instead of "∧" for brevity):

$$\exists \mathsf{worksAt.University}(x), \mathsf{supervises}(x,z), \mathsf{PhDStudent}(z) \rightarrow \mathsf{profOf}(x,z)$$

The above transformation allows us to reduce tree-shaped rules to rules of only linear structure that are much more similar to RIAs in $\mathcal{SROIQ}$. But while all role atoms now belong to a single maximal path, rules might still contain disconnected concept atoms, such as, e.g., in the rule $R(x, y) \wedge C(z) \rightarrow S(x, y)$.

It can be shown that DL rules in $\mathcal{SROIQ}$ can be internalised.

Theorem 7 *Consider a rule base* RB *and a knowledge base* KB *for* $\mathcal{SROIQ}$*, such that* RB∪KB *is admissible. There is a* $\mathcal{SROIQ}$ *knowledge base* $\mathrm{KB}_{\mathrm{RB}}$ *that can be computed in time polynomial in the size of* RB*, such that* KB ∪ RB *and* KB ∪ $\mathrm{KB}_{\mathrm{RB}}$ *are equisatisfiable.*

Moreover, if KB ∪ RB *is regular, then* KB ∪ $\mathrm{KB}_{\mathrm{RB}}$ *is also regular.*

Considering again our introductory example, we arrive at the following $\mathcal{SROIQ}$ axioms (where S_1, S_2 are new auxiliary roles):

$$S_1 \circ \mathsf{supervises} \circ S_2 \sqsubseteq \mathsf{profOf}$$

$$\exists \mathsf{worksAt.University} \equiv \exists S_1.\mathsf{Self} \qquad \mathsf{PhDStudent} \equiv \exists S_2.\mathsf{Self}$$

Based on Theorem 7, we conclude that the problem of checking the

satisfiability of $\mathcal{SROIQ}$ knowledge bases extended with DL rules is decidable, as long as the extended knowledge base is admissible and regular. Since the internalisation is possible in polynomial time, the worst-case complexity for this problem is the same as for checking satisfiability of $\mathcal{SROIQ}$ knowledge bases.

5 DL RULES IN $\mathcal{EL}^{++}$

In this section, we investigate DL rules for the DL $\mathcal{EL}^{++}$ [2], for which many typical inference problems can be solved in polynomial time. As $\mathcal{EL}^{++}$ cannot internalise DL rules, they constitute a true extension of expressivity. We therefore take a different approach than in $\mathcal{SROIQ}$: instead of considering rule bases as an auxiliary set of axioms that is successively reduced and internalised, we introduce DL rules as core expressive mechanism to which all other $\mathcal{EL}^{++}$ axioms can be reduced. While $\mathcal{EL}^{++}$ rule bases offer many expressive features formerly unavailable in $\mathcal{EL}^{++}$, we show that the complexity of core inference problems remains tractable. We simplify our presentation by omitting concrete domains from $\mathcal{EL}^{++}$ – they are not affected by our extension and can be treated as shown in [2].

Definition 8 *An atomic role of $\mathcal{EL}^{++}$ is a (non-inverse) role name. An $\mathcal{EL}^{++}$ Rbox is a set of generalised role inclusion axioms, and an $\mathcal{EL}^{++}$ Tbox is a $\mathcal{SROIQ}$ Tbox that contains only the following concept constructors: $\sqcap$, $\exists$, $\top$, $\bot$, as well as nominal classes $\{a\}$. An $\mathcal{EL}^{++}$* rule base *is a set of DL rules for $\mathcal{EL}^{++}$ that do not contain atoms of the form $R(x, x)$ in the body.*

Note that we do not have any requirement for regularity or simplicity of roles in the context of $\mathcal{EL}^{++}$. It turns out that neither is relevant for obtaining decidability or tractability. The case of $R(x, x)$ in bodies is not addressed by the below algorithm, and an according extension is left to future work. Since it is obvious that both concept and role inclusion axioms can directly be expressed by DL rules, we will consider only $\mathcal{EL}^{++}$ rule bases without any additional $\mathcal{EL}^{++}$ knowledge base axioms. We can restrict our attention to $\mathcal{EL}^{++}$ rules in a certain normal form:

Definition 9 *An $\mathcal{EL}^{++}$ rule base* RB *is in* normal form *if all concept atoms in rule bodies are either concept names or nominals, all variables in a rule's head also occur in its body, and all rule heads are of one of the following forms:*

$$A(x) \qquad \exists R.A(x) \qquad R(x, y)$$

where $A \in \mathsf{N}_C \cup \{\{a\} \mid a \in \mathsf{N}_I\} \cup \{\top, \bot\}$ and $R \in \mathsf{N}_R$. A set $\mathcal{B}$ of basic concept expressions *for* RB *is defined as $\mathcal{B} := \{C \mid C \in \mathsf{N}_C, C \text{ occurs in } \mathrm{RB}\} \cup \{\{a\} \mid a \in \mathsf{N}_I, a \text{ occurs in } \mathrm{RB}\} \cup \{\top, \bot\}$.*

Proposition 10 *Any $\mathcal{EL}^{++}$ rule base can be transformed into an equisatisfiable $\mathcal{EL}^{++}$ rule base in normal form. The transformation can be done in polynomial time.*

When checking satisfiability of $\mathcal{EL}^{++}$ rule bases, we can thus restrict to rule bases in the above normal form. A polynomial algorithm for checking class subsumptions in $\mathcal{EL}^{++}$ knowledge bases has been given in [2], and it was shown that other standard inference problems can easily be reduced to that problem. We now present a new algorithm for checking satisfiability of $\mathcal{EL}^{++}$ rule bases, and show its correctness and tractability. Clearly, subsumption checking can be reduced to this problem: given a new individual $a \in \mathsf{N}_I$, the rule base $\mathrm{RB} \cup \{C(a), \{a\}(x) \sqcap D(x) \to \bot(x)\}$ is unsatisfiable iff RB entails $C \sqsubseteq D$. Instance checking in turn is directly reducible to subsumption checking in the presence of nominals.

Algorithm 11 *The algorithm proceeds by computing two sets: a set $\mathcal{E}$ of inferred "domain elements", and a set $\mathcal{S}$ of relevant subclass inclusion axioms that are entailed by* RB*. The elements of $\mathcal{E}$ are represented by basic concept expressions of* RB*, i.e. $\mathcal{E} \subseteq \mathcal{B}$, and the inclusion axioms in $\mathcal{S}$ are of the form $C \sqsubseteq D$ or $C \sqsubseteq \exists R.D$, where $C, D \in \mathcal{E}$. Thus $\mathcal{E}$ and $\mathcal{S}$ are polynomially bounded by the size of* RB*.*

Initially, we set $\mathcal{E} := \{\{a\} \mid \{a\} \in \mathcal{B}\} \cup \{\top\}$ and $\mathcal{S} := \emptyset$. Now a DL rule is applied whenever we find that there is a match *with the rule body. Given a rule $B \to H$, a match θ is a mapping from all variables in B to elements of $\mathcal{E}$, such that the following hold:*

- *for every $C(y) \in B$, $\theta(y) \sqsubseteq C \in \mathcal{S}$, and*
- *for every $R(y, z) \in B$, $\theta(y) \sqsubseteq \exists R.\theta(z) \in \mathcal{S}$.*

The algorithm now proceeds by applying the following rules until no possible rule application further modifies the set $\mathcal{E}$ or $\mathcal{S}$:

(EL1) If $C \in \mathcal{E}$, then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq C, C \sqsubseteq \top\}$.
(EL2) If there is a rule $B \to E(x) \in$ RB, and if there is a match θ for B with $\theta(x) = \theta_x$, then $\mathcal{S} := \mathcal{S} \cup \{\theta_x \sqsubseteq E\}$. In this case, if $E = C$ or $E = \exists R.C$, then $\mathcal{E} := \mathcal{E} \cup \{C\}$.
(EL3) If there is a rule $B \to R(x, y) \in$ RB, and if there is a match θ for B with $\theta(x) = \theta_x$ and $\theta(y) = \theta_y$, then $\mathcal{S} := \mathcal{S} \cup \{\theta_x \sqsubseteq \exists R.\theta_y\}$.
(EL4) If $\{C \sqsubseteq \{a\}, D \sqsubseteq \{a\}, D \sqsubseteq E\} \subseteq \mathcal{S}$ then $\mathcal{S} := \mathcal{S} \cup \{C \sqsubseteq E\}$.

Here we assume that $C, D, D' \in \mathcal{B}$, $E \in \mathcal{B} \cup \{\exists R.C \mid C \in \mathcal{B}\}$, and $R \in \mathsf{N}_R$. After termination, the algorithm returns "unsatisfiable" if $\bot \in \mathcal{E}$, and "satisfiable" otherwise.

The correctness of the above algorithm is shown by using the sets $\mathcal{E}$ and $\mathcal{S}$ for constructing a model, which is indeed possible whenever $\bot \notin \mathcal{E}$ (see [1] for details). $\mathcal{EL}^{++}$ has a small model property that allows us to consider at most one individual for representing the members of each class. The set $\mathcal{E}$ thus is used to record classes which must have some element, and matches θ use these class names to represent (arbitrary) individuals to which some DL rule might be applied.

Assuming that all steps of Algorithm 11 are computable in polynomial time, it is easy to see that the algorithm also terminates in polynomial time, since there are only polynomially many possible elements for $\mathcal{E}$ and $\mathcal{S}$, and each case adds new elements to either set. However, it also has to be verified that individual steps can be computed efficiently, and this is not obvious for the match-checks in (EL2) and (EL3). Indeed, finding matches in query graphs is known to be NP-complete in general, and the tree-like structure of queries is crucial to retain tractability. Moreover, even tree-like rule bodies admit exponentially many matches. But note that Algorithm 11 does not consider *all matches* but only the (polynomially many) possible values of θ_x (and θ_y). It turns out that there is indeed a algorithm that checks in polynomial time whether a match θ as in (EL2) and (EL3) exists, but without explicitly considering all possible matches.

Proposition 12 *Consider a rule of the form $B \to C(x)$ ($B \to R(x, y)$), sets $\mathcal{E}$ and $\mathcal{S}$ as in Algorithm 11, and an element $\theta_x \in \mathcal{E}$ (elements $\theta_x, \theta_y \in \mathcal{E}$). There is an algorithm that decides whether there is a match θ such that $\theta(x) = \theta_x$ ($\theta(x) = \theta_x$ and $\theta(y) = \theta_y$), running in polynomial time w.r.t. the size of the inputs.*

Theorem 13 *Algorithm 11 is a sound and complete procedure for checking satisfiability of $\mathcal{EL}^{++}$ rule bases. Satisfiability checking, instance retrieval, and computing class subsumptions for $\mathcal{EL}^{++}$ rule bases is possible in polynomial time in the size of the rule base.*

6 DLP 2

Description Logic Programs (DLP) have been proposed as a tractable knowledge representation formalism for bridging the gap between DL and (Horn) logic programming [5]. This clearly suggests further extension with DL rules, and we will see below that reasoning with this extension is still tractable. Moreover, various further features of $\mathcal{SROIQ}$ can easily be included as well, and thus we arrive at a DL rule language that might be dubbed DLP 2 in analogy to the ongoing standardisation of the extended of OWL 2 based on $\mathcal{SROIQ}$.

DLP has been defined in various ways, and a detailed syntactic characterisation is found in [11]. Essentially, however, DLP can be characterised as the fragment of $\mathcal{SHOIQ}$ that can entail neither disjunctive information nor the existence of anonymous individuals. The former condition has been extensively studied in the context of Horn description logics [8], and rather complex syntactic definitions can be given to characterise all admissible axioms of such logics. Here, we adopt a much simpler definition that focusses on the essential expressive features without encompassing all alternative syntactic forms of DLP axioms:

Definition 14 *Atomic roles of DLP are defined as in $\mathcal{SROIQ}$, including inverse roles. A* DLP body concept *is any $\mathcal{SROIQ}$ concept expression that includes only concept names, nominals, $\sqcap$, $\exists$, $\top$, and $\bot$. A* DLP head concept *is any $\mathcal{SROIQ}$ concept expression that includes only concept names, nominals, $\sqcap$, $\forall$, $\top$, $\bot$, and expressions of the form $\leq 1.C$ where C is a DLP body concept.*

A DLP knowledge base *is a set of Rbox axioms of the form $R \sqsubseteq S$ and $R \circ R \sqsubseteq R$, Tbox axioms of the form $C \sqsubseteq D$, and Abox axioms of the form $D(a)$ and $R(a,b)$, where $C \in \mathbf{C}$ is a body concept, $D \in \mathbf{C}$ is a head concept, and $a, b \in \mathsf{N}_I$ are individual names. A* DLP rule base *is a set of DL rules such that all concepts in rule bodies are body concepts, and all concepts in rule heads are head concepts.*

A DLP 2 *knowledge base consists of a DLP knowledge base that additionally might contain Rbox axioms of the form $\mathsf{Dis}(R,S)$ and $\mathsf{Asy}(R)$, together with some DLP rule base.*

Note that neither regularity nor simplicity restrictions apply in DLP. It is immediate that DLP Rbox and Tbox axioms can directly by expressed by DLP rules. The same holds for Abox axioms: though we cannot use the common translation of $R(a,b)$ into $\{a\}(x) \rightarrow \exists R.\{b\}(x)$, the DLP rule $\{a\}(x) \wedge \{b\}(y) \rightarrow R(x,y)$ serves the same purpose. Hence we can restrict our further considerations to DLP 2 knowledge bases into which all knowledge base axioms other than $\mathsf{Dis}(R,S)$ and $\mathsf{Asy}(R)$ have been internalised. The core observation of this section is as follows:

Proposition 15 *Any DLP 2 knowledge base* KB *can be transformed into an equisatisfiable set of function-free first-order Horn rules with at most five variables per formula, and this transformation is possible in polynomial time w.r.t. the size of* KB.

Function-free Horn-logic with a limited number of variables per rule is known to be tractable, and we can thus conclude the tractability of DLP 2:

Theorem 16 *Satisfiability checking, instance retrieval, and computing class subsumptions for DLP 2 knowledge bases is possible in polynomial time in the size of the knowledge base.*

7 CONCLUSION

We have introduced *DL rules* as a rule-based formalism for augmenting description logic knowledge bases. For all DLs considered in this paper – $\mathcal{SROIQ}$, $\mathcal{EL}^{++}$, and DLP – the extension with DL rules does not increase the worst-case complexity. In particular, $\mathcal{EL}^{++}$ rules and the extended DLP 2 allow for polynomial time reasoning for common inference tasks, even though DL rules do indeed provide added expressive features in those cases.

The main contributions of this paper therefore are twofold. Firstly, we have extended the expressivity of two tractable DLs while preserving their favourable computational properties. The resulting formalisms of $\mathcal{EL}^{++}$ rules and DLP 2 are arguably close to being maximal tractable fragments of $\mathcal{SROIQ}$. In particular, note that the union of $\mathcal{EL}^{++}$ and DLP is no longer tractable, even when disallowing number restrictions and inverse roles: this follows from the fact that this DL contains the DL Horn-$\mathcal{FLE}$ which was shown to be ExpTime-complete in [8].

Secondly, while DL rules do not truly add expressive power to $\mathcal{SROIQ}$, our characterisation and reduction methods for DL rules provides a basis for developing ontology modelling tools. Indeed, even without any further extension, the upcoming OWL 2 standard would support all DL rules. Hence OWL-conformant tools can choose to provide rule-based user interfaces, and rule-based tools may offer some amount of OWL support. We remark that in the case of DLP and $\mathcal{EL}^{++}$, the conditions imposed on DL rules can be checked individually for each rule without considering the knowledge base as a whole. Moreover, in order to simplify rule editing, the general syntax of DL rules can be further restricted without sacrificing expressivity, e.g. by considering only chains rather than trees for rule bodies. We thus argue that DL rules can be a useful interface paradigm for many application fields.

Our treatment of rules in $\mathcal{EL}^{++}$ and DLP 2 – used only for establishing complexity bounds in this paper – can be the basis for novel rule-based reasoning algorithms for those DLs, and we leave it for future research to explore this approach.

REFERENCES

[1] Anonymous. Description logic rules. Full version with proofs, available at `http://dlrules.tripod.com/`, 2007.

[2] Franz Baader, Sebastian Brandt, and Carsten Lutz, 'Pushing the EL envelope', in *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, Edinburgh, UK, (2005). Morgan-Kaufmann Publishers.

[3] *The Description Logic Handbook: Theory, Implementation and Applications*, eds., Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, Cambridge University Press, 2007.

[4] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov, 'Complexity and expressive power of logic programming', *ACM Computing Surveys*, **33**, 374–425, (2001).

[5] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker, 'Description logic programs: combining logic programs with description logic', in *Proc. 12th Int. Conf. on World Wide Web (WWW 2003)*, pp. 48–57. ACM, (2003).

[6] Ian Horrocks, Oliver Kutz, and Ulrike Sattler, 'The even more irresistible $\mathcal{SROIQ}$', in *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006)*, pp. 57–67. AAAI Press, (June 2006).

[7] Ian Horrocks and Peter F. Patel-Schneider, 'A proposal for an OWL rules language', in *Proc. 13th Int. Conf. on World Wide Web (WWW 2004)*, eds., Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, pp. 723–731. ACM, (2004).

[8] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler, 'Complexity boundaries for Horn description logics', in *Proc. 22nd AAAI Conf. (AAAI'07)*, (2007).

[9] Boris Motik, Ulrike Sattler, and Rudi Studer, 'Query answering for OWL-DL with rules', *J. Web Sem.*, **3**(1), 41–60, (2005).

[10] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler, 'All elephants are bigger than all mice', in *Proc. 21st Int. Workshop on Description Logics (DL-08)*, (2008).

[11] Raphael Volz, *Web Ontology Reasoning with Logic Databases*, Ph.D. dissertation, Universität Karlsruhe (TH), Germany, 2004.