

# A Set-Theoretic Approach to ABox Reasoning Services

Domenico Cantone, Marianna Nicolosi-Asmundo,  
and Daniele Francesco Santamaria<sup>(✉)</sup>

Department of Mathematics and Computer Science,  
University of Catania, Catania, Italy  
{cantone,nicolosi,santamaria}@dmi.unict.it

**Abstract.** In this paper we consider the most common ABox reasoning services for the description logic  $\mathcal{DL}\langle 4\text{LQS}^{\text{R},\times}\rangle(\mathbf{D})$  ( $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ , for short) and prove their decidability via a reduction to the satisfiability problem for the set-theoretic fragment  $4\text{LQS}^{\text{R}}$ . The description logic  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is very expressive, as it admits various concept and role constructs and data types that allow one to represent rule-based languages such as SWRL.

Decidability results are achieved by defining a generalization of the conjunctive query answering (CQA) problem that can be instantiated to the most widespread ABox reasoning tasks. We also present a KE-tableau based procedure for calculating the answer set from  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -knowledge bases and higher order  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries, thus providing means for reasoning on several well-known ABox reasoning tasks. Our calculus extends a previously introduced KE-tableau based decision procedure for the CQA problem.

## 1 Introduction

Recently, results from Computable Set Theory have been applied to knowledge representation for the semantic web in order to define and reason about description logics and rule languages. Such a study is motivated by the fact that Computable Set Theory is a research field full of interesting decidability results and that there exists a natural translation map between some set-theoretic fragments and description logics and rule languages.

In particular, the decidable four-level stratified fragment of set theory  $4\text{LQS}^{\text{R}}$ , involving variables of four sorts, pair terms, and a restricted form of quantification over variables of the first three sorts (cf. [4]), has been used in [3] to represent the description logic  $\mathcal{DL}\langle 4\text{LQS}^{\text{R}}\rangle(\mathbf{D})$  ( $\mathcal{DL}_{\mathbf{D}}^4$ , for short). The logic  $\mathcal{DL}_{\mathbf{D}}^4$  admits concept constructs such as full negation, union and intersection of concepts, concept domain and range, existential quantification and min cardinality on the left-hand side of inclusion axioms. It also supports role constructs such as role chains on the left hand side of inclusion axioms, union, intersection, and

---

This work has been partially supported by the Polish National Science Centre research project DEC-2011/02/A/HS1/00395.

complement of abstract roles, and properties on roles such as transitivity, symmetry, reflexivity, and irreflexivity. As briefly shown in [3],  $\mathcal{DL}_{\mathbf{D}}^4$  is particularly suitable to express a rule language such as the Semantic Web Rule Language (SWRL), an extension of the Ontology Web Language (OWL). It admits data types, a simple form of concrete domains that are relevant in real world applications. In [3], the consistency problem for  $\mathcal{DL}_{\mathbf{D}}^4$ -knowledge bases has been proved decidable by means of a reduction to the satisfiability problem for  $4\text{LQS}^{\mathbf{R}}$ , whose decidability has been established in [4]. It has also been shown that, under not very restrictive constraints, the consistency problem for  $\mathcal{DL}_{\mathbf{D}}^4$ -knowledge bases is **NP**-complete. Such a low complexity result is motivated by the fact that existential quantification cannot appear on the right-hand side of inclusion axioms. Nonetheless,  $\mathcal{DL}_{\mathbf{D}}^4$  turns out to be more expressive than other low complexity logics such as OWL RL and therefore it is suitable for representing real world ontologies. For example, the restricted version of  $\mathcal{DL}_{\mathbf{D}}^4$  mentioned above allows one to express several ontologies, such as *Ontoceramic* [9], for the classification of ancient pottery.

In [7], the description logic  $\mathcal{DL}\langle 4\text{LQS}^{\mathbf{R},\times}\rangle(\mathbf{D})$  ( $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ , for short), extending  $\mathcal{DL}_{\mathbf{D}}^4$  with Boolean operations on concrete roles and with the product of concepts, has been introduced and the *Conjunctive Query Answering* (CQA) problem for  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  has been proved decidable via a reduction to the CQA problem for  $4\text{LQS}^{\mathbf{R}}$ , whose decidability follows from that of  $4\text{LQS}^{\mathbf{R}}$  (see [4]). CQA is a powerful way to query ABoxes, particularly relevant in the context of description logics and for real world applications based on semantic web technologies, as it provides mechanisms for interacting with ontologies and data. The CQA problem for description logics has been introduced in [1,2] (further references on the problem can be found in [8]). Finally, we mention also a terminating KE-tableau based procedure that, given a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -query  $Q$  and a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -knowledge base  $\mathcal{KB}$  represented in set-theoretic terms, determines the answer set of  $Q$  with respect to  $\mathcal{KB}$ . KE-tableau systems [10] allow the construction of trees whose distinct branches define mutually exclusive situations, thus preventing the proliferation of redundant branches, typical of semantic tableaux.

In this paper we extend the results presented in [7] by considering also the main ABox reasoning tasks for  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ , such as *instance checking* and *concept retrieval*, and study their decidability via a reduction to the satisfiability problem for  $4\text{LQS}^{\mathbf{R}}$ . Specifically, we define *Higher Order* (HO)  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries admitting variables of three sorts: individual and data type variables, concept variables, and role variables. HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries can be instantiated to any of the ABox reasoning tasks we are considering in the paper. Then, we define the *Higher Order Conjunctive Query Answering* (HOCQA) problem for  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  and prove its decidability by reducing it to the HOCQA problem for  $4\text{LQS}^{\mathbf{R}}$ . Decidability of the latter problem follows from that of the satisfiability problem for  $4\text{LQS}^{\mathbf{R}}$ .  $4\text{LQS}^{\mathbf{R}}$  representation of  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -knowledge bases is defined according to [7].  $4\text{LQS}^{\mathbf{R}}$  turns out to be naturally suited for the HOCQA problem since HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries are easily translated into  $4\text{LQS}^{\mathbf{R}}$ -formulae. In particular, individual and data type variables are mapped into  $4\text{LQS}^{\mathbf{R}}$  variables of sort 0, concept variables into  $4\text{LQS}^{\mathbf{R}}$  variables of sort 1, and role variables

into 4LQS<sup>R</sup> variables of sort 3. Finally, we present an extension of the KE-tableau presented in [7], which provides a decision procedure for the HOCQA task for  $\mathcal{DL}_{\mathcal{D}}^{4,\times}$ .

## 2 Preliminaries

### 2.1 The Set-Theoretic Fragment 4LQS<sup>R</sup>

It is convenient to first introduce the syntax and semantics of the more general four-level quantified language 4LQS. Then we provide some restrictions on the quantified formulae of 4LQS to characterize 4LQS<sup>R</sup>. The interested reader can find more details in [4] together with the decision procedure for the satisfiability problem for 4LQS<sup>R</sup>.

4LQS involves four collections,  $\mathcal{V}_i$ , of variables of sort  $i = 0, 1, 2, 3$ , respectively. These will be denoted by  $X^i, Y^i, Z^i, \dots$  (in particular, variables of sort 0 will also be denoted by  $x, y, z, \dots$ ). In addition to variables, 4LQS involves also *pair terms* of the form  $\langle x, y \rangle$ , for  $x, y \in \mathcal{V}_0$ .

4LQS-*quantifier-free atomic formulae* are classified as:

- level 0:  $x = y, \quad x \in X^1, \quad \langle x, y \rangle = X^2, \quad \langle x, y \rangle \in X^3$ ;
- level 1:  $X^1 = Y^1, \quad X^1 \in X^2$ ;
- level 2:  $X^2 = Y^2, \quad X^2 \in X^3$ .

4LQS-*purely universal formulae* are classified as:

- level 1:  $(\forall z_1) \dots (\forall z_n) \varphi_0$ , where  $z_1, \dots, z_n \in \mathcal{V}_0$  and  $\varphi_0$  is any propositional combination of quantifier-free atomic formulae of level 0;
- level 2:  $(\forall Z_1^1) \dots (\forall Z_m^1) \varphi_1$ , where  $Z_1^1, \dots, Z_m^1 \in \mathcal{V}_1$  and  $\varphi_1$  is any propositional combination of quantifier-free atomic formulae of levels 0 and 1, and of purely universal formulae of level 1;
- level 3:  $(\forall Z_1^2) \dots (\forall Z_p^2) \varphi_2$ , where  $Z_1^2, \dots, Z_p^2 \in \mathcal{V}_2$  and  $\varphi_2$  is any propositional combination of quantifier-free atomic formulae and of purely universal formulae of levels 1 and 2.

4LQS-formulae are all the propositional combinations of quantifier-free atomic formulae of levels 0, 1, 2, and of purely universal formulae of levels 1, 2, 3.

The variables  $z_1, \dots, z_n$  are said to occur *quantified* in  $(\forall z_1) \dots (\forall z_n) \varphi_0$ . Likewise,  $Z_1^1, \dots, Z_m^1$  and  $Z_1^2, \dots, Z_p^2$  occur quantified in  $(\forall Z_1^1) \dots (\forall Z_m^1) \varphi_1$  and in  $(\forall Z_1^2) \dots (\forall Z_p^2) \varphi_2$ , respectively. A variable occurs *free* in a 4LQS-formula  $\varphi$  if it does not occur quantified in any subformula of  $\varphi$ . For  $i = 0, 1, 2, 3$ , we denote with  $\text{Var}_i(\varphi)$  the collections of variables of level  $i$  occurring free in  $\varphi$  and we put  $\text{Vars}(\varphi) := \bigcup_{i=0}^3 \text{Var}_i(\varphi)$ .

A substitution  $\sigma := \{x/y, X^1/Y^1, X^2/Y^2, X^3/Y^3\}$  is the mapping  $\varphi \mapsto \varphi\sigma$  such that, for any given 4LQS-formula  $\varphi$ ,  $\varphi\sigma$  is the 4LQS-formula obtained from  $\varphi$  by replacing the free occurrences of the variables  $x_i$  in  $\mathbf{x}$  (for  $i = 1, \dots, n$ ) with the corresponding  $y_i$  in  $\mathbf{y}$ , of  $X_j^1$  in  $\mathbf{X}^1$  (for  $j = 1, \dots, m$ ) with  $Y_j^1$  in  $\mathbf{Y}^1$ , of

$X_k^2$  in  $\mathbf{X}^2$  (for  $k = 1, \dots, p$ ) with  $Y_k^2$  in  $\mathbf{Y}^2$ , and of  $X_h^3$  in  $\mathbf{X}^3$  (for  $h = 1, \dots, q$ ) with  $Y_h^3$  in  $\mathbf{Y}^3$ , respectively. A substitution  $\sigma$  is *free* for  $\varphi$  if the formulae  $\varphi$  and  $\varphi\sigma$  have exactly the same occurrences of quantified variables. The *empty substitution*, denoted with  $\epsilon$ , satisfies  $\varphi\epsilon = \varphi$ , for every 4LQS-formula  $\varphi$ .

A 4LQS-*interpretation* is a pair  $\mathcal{M} = (D, M)$ , where  $D$  is a non-empty collection of objects (called *domain* or *universe* of  $\mathcal{M}$ ) and  $M$  is an assignment over the variables in  $\mathcal{V}_i$ , for  $i = 0, 1, 2, 3$ , such that:

$$MX^0 \in D, \quad MX^1 \in \mathcal{P}(D), \quad MX^2 \in \mathcal{P}(\mathcal{P}(D)), \quad MX^3 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(D))),$$

where  $X^i \in \mathcal{V}_i$ , for  $i = 0, 1, 2, 3$ , and  $\mathcal{P}(s)$  denotes the powerset of  $s$ .

Pair terms are interpreted *à la* Kuratowski, and therefore we put

$$M\langle x, y \rangle := \{\{Mx\}, \{Mx, My\}\}.$$

Quantifier-free atomic formulae and purely universal formulae are evaluated in a standard way according to the usual meaning of the predicates ‘ $\in$ ’ and ‘ $=$ ’. The interpretation of quantifier-free atomic formulae and of purely universal formulae is given in [4].

Finally, compound formulae are interpreted according to the standard rules of propositional logic. If  $\mathcal{M} \models \varphi$ , then  $\mathcal{M}$  is said to be a 4LQS-*model* for  $\varphi$ . A 4LQS-formula is said to be *satisfiable* if it has a 4LQS-model. A 4LQS-formula is *valid* if it is satisfied by all 4LQS-interpretations.

We are now ready to present the fragment 4LQS<sup>R</sup> of 4LQS of our interest. This is the collection of the formulae  $\psi$  of 4LQS fulfilling the restrictions:

1. for every purely universal formula  $(\forall Z_1^1) \dots (\forall Z_m^1)\varphi_1$  of level 2 occurring in  $\psi$  and every purely universal formula  $(\forall z_1) \dots (\forall z_n)\varphi_0$  of level 1 occurring negatively in  $\varphi_1$ ,  $\varphi_0$  is a propositional combination of quantifier-free atomic formulae of level 0 and the condition

$$\neg\varphi_0 \rightarrow \bigwedge_{i=1}^n \bigwedge_{j=1}^m z_i \in Z_j^1$$

is a valid 4LQS-formula (in this case we say that  $(\forall z_1) \dots (\forall z_n)\varphi_0$  is *linked to the variables*  $Z_1^1, \dots, Z_m^1$ );

2. for every purely universal formula  $(\forall Z_1^2) \dots (\forall Z_p^2)\varphi_2$  of level 3 in  $\psi$ :
  - every purely universal formula of level 1 occurring negatively in  $\varphi_2$  and not occurring in a purely universal formula of level 2 is only allowed to be of the form

$$(\forall z_1) \dots (\forall z_n) \neg \left( \bigwedge_{i=1}^n \bigwedge_{j=1}^n \langle z_i, z_j \rangle = Y_{ij}^2 \right),$$

- with  $Y_{ij}^2 \in \mathcal{V}^2$ , for  $i, j = 1, \dots, n$ ;
- purely universal formulae  $(\forall Z_1^1) \dots (\forall Z_m^1) \varphi_1$  of level 2 may occur only positively in  $\varphi_2$ .<sup>1</sup>

Restriction 1 has been introduced for technical reasons related to the decidability of the satisfiability problem for the fragment, while restriction 2 allows one to define binary relations and several operations on them (see [4] for details).

The semantics of  $4\mathcal{LQS}^R$  plainly coincides with that of  $4\mathcal{LQS}$ .

## 2.2 The Logic $\mathcal{DL}\langle 4\mathcal{LQS}^{R;\times} \rangle(\mathbf{D})$

The description logic  $\mathcal{DL}\langle 4\mathcal{LQS}^{R;\times} \rangle(\mathbf{D})$  (which, as already remarked, will be more simply referred to as  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$ ) is an extension of the logic  $\mathcal{DL}\langle 4\mathcal{LQS}^R \rangle(\mathbf{D})$  presented in [3], where Boolean operations on concrete roles and the product of concepts are introduced. In addition to other features,  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$  admits also *data types*, a simple form of concrete domains that are relevant in real-world applications. In particular, it treats derived data types by admitting data type terms constructed from data ranges by means of a finite number of applications of the Boolean operators. Basic and derived data types can be used inside inclusion axioms involving concrete roles.

Data types are introduced through the notion of *data type map*, defined according to [12] as follows. Let  $\mathbf{D} = (N_D, N_C, N_F, \cdot^{\mathbf{D}})$  be a data type map, where  $N_D$  is a finite set of data types,  $N_C$  is a function assigning a set of constants  $N_C(d)$  to each data type  $d \in N_D$ ,  $N_F$  is a function assigning a set of facets  $N_F(d)$  to each  $d \in N_D$ , and  $\cdot^{\mathbf{D}}$  is a function assigning a data type interpretation  $d^{\mathbf{D}}$  to each data type  $d \in N_D$ , a facet interpretation  $f^{\mathbf{D}} \subseteq d^{\mathbf{D}}$  to each facet  $f \in N_F(d)$ , and a data value  $e_d^{\mathbf{D}} \in d^{\mathbf{D}}$  to every constant  $e_d \in N_C(d)$ . We shall assume that the interpretations of the data types in  $N_D$  are non-empty pairwise disjoint sets.

Let  $\mathbf{R}_A, \mathbf{R}_D, \mathbf{C}, \mathbf{I}$  be denumerable pairwise disjoint sets of abstract role names, concrete role names, concept names, and individual names, respectively. We assume that the set of abstract role names  $\mathbf{R}_A$  contains a name  $U$  denoting the universal role.

(a)  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$ -data types, (b)  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$ -concepts, (c)  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$ -abstract roles, and (d)  $\mathcal{DL}_{\mathbf{D}}^{4;\times}$ -concrete role terms are constructed according to the following syntax rules:

- (a)  $t_1, t_2 \longrightarrow dr \mid \neg t_1 \mid t_1 \sqcap t_2 \mid t_1 \sqcup t_2 \mid \{e_d\}$ ,
- (b)  $C_1, C_2 \longrightarrow A \mid \top \mid \perp \mid \neg C_1 \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \{a\} \mid \exists R. Self \mid \exists R. \{a\} \mid \exists P. \{e_d\}$ ,
- (c)  $R_1, R_2 \longrightarrow S \mid U \mid R_1^- \mid \neg R_1 \mid R_1 \sqcup R_2 \mid R_1 \sqcap R_2 \mid R_{C_1} \mid R_{|C_1} \mid R_{C_1} \mid c_2 \mid id(C) \mid C_1 \times C_2$ ,
- (d)  $P_1, P_2 \longrightarrow T \mid \neg P_1 \mid P_1 \sqcup P_2 \mid P_1 \sqcap P_2 \mid P_{C_1} \mid P_{t_1} \mid P_{C_1 t_1}$ ,

<sup>1</sup> Definitions of positive and of negative occurrences of a formula within another formula can be found in [4].

where  $dr$  is a data range for  $\mathbf{D}$ ,  $t_1, t_2$  are data type terms,  $e_d$  is a constant in  $N_C(d)$ ,  $a$  is an individual name,  $A$  is a concept name,  $C_1, C_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concept terms,  $S$  is an abstract role name,  $R, R_1, R_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role terms,  $T$  is a concrete role name, and  $P, P_1, P_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concrete role terms. We remark that data type terms are introduced in order to represent derived data types.

A  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -knowledge base is a triple  $\mathcal{K} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$  such that  $\mathcal{R}$  is a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $RBox$ ,  $\mathcal{T}$  is a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $TBox$ , and  $\mathcal{A}$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $ABox$ . These are defined as follows.

A  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $RBox$  is a collection of statements of the following forms:  $R_1 \equiv R_2$ ,  $R_1 \sqsubseteq R_2$ ,  $R_1 \dots R_n \sqsubseteq R_{n+1}$ ,  $\text{Sym}(R_1)$ ,  $\text{Asym}(R_1)$ ,  $\text{Ref}(R_1)$ ,  $\text{Irref}(R_1)$ ,  $\text{Dis}(R_1, R_2)$ ,  $\text{Tra}(R_1)$ ,  $\text{Fun}(R_1)$ ,  $R_1 \equiv C_1 \times C_2$ ,  $P_1 \equiv P_2$ ,  $P_1 \sqsubseteq P_2$ ,  $\text{Dis}(P_1, P_2)$ ,  $\text{Fun}(P_1)$ , where  $R_1, R_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role terms,  $C_1, C_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract concept terms, and  $P_1, P_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concrete role terms. Any expression of the type  $w \sqsubseteq R$ , where  $w$  is a finite string of  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role terms and  $R$  is an  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role term, is called a *role inclusion axiom (RIA)*.

A  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $TBox$  is a set of statements of the following types:

- $C_1 \equiv C_2$ ,  $C_1 \sqsubseteq C_2$ ,  $C_1 \sqsubseteq \forall R_1.C_2$ ,  $\exists R_1.C_1 \sqsubseteq C_2$ ,  $\geq_n R_1.C_1 \sqsubseteq C_2$ ,  $C_1 \sqsubseteq \leq_n R_1.C_2$ ,
- $t_1 \equiv t_2$ ,  $t_1 \sqsubseteq t_2$ ,  $C_1 \sqsubseteq \forall P_1.t_1$ ,  $\exists P_1.t_1 \sqsubseteq C_1$ ,  $\geq_n P_1.t_1 \sqsubseteq C_1$ ,  $C_1 \sqsubseteq \leq_n P_1.t_1$ ,

where  $C_1, C_2$  are  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concept terms,  $t_1, t_2$  data type terms,  $R_1$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role term,  $P_1$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concrete role term. Any statement of the form  $C \sqsubseteq D$ , with  $C, D$   $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concept terms, is a *general concept inclusion axiom*.

A  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ - $ABox$  is a set of *individual assertions* of the forms:  $a : C_1$ ,  $(a, b) : R_1$ ,  $a = b$ ,  $a \neq b$ ,  $e_d : t_1$ ,  $(a, e_d) : P_1$ , with  $C_1$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concept term,  $d$  a data type,  $t_1$  a data type term,  $R_1$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role term,  $P_1$  a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concrete role term,  $a, b$  individual names, and  $e_d$  a constant in  $N_C(d)$ .

The semantics of  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is given by means of an interpretation  $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ , where  $\Delta^{\mathbf{I}}$  and  $\Delta_{\mathbf{D}}$  are non-empty disjoint domains such that  $d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$ , for every  $d \in N_D$ , and  $\cdot^{\mathbf{I}}$  is an interpretation function. The definition of the interpretation of concepts and roles, axioms and assertions is illustrated in [8, Table 1].

Let  $\mathcal{R}$ ,  $\mathcal{T}$ , and  $\mathcal{A}$  be as above. An interpretation  $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$  is a  $\mathbf{D}$ -model of  $\mathcal{R}$  (resp.,  $\mathcal{T}$ ), and we write  $\mathbf{I} \models_{\mathbf{D}} \mathcal{R}$  (resp.,  $\mathbf{I} \models_{\mathbf{D}} \mathcal{T}$ ) if  $\mathbf{I}$  satisfies each axiom in  $\mathcal{R}$  (resp.,  $\mathcal{T}$ ) according to the semantic rules in [8, Table 1]. Analogously,  $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$  is a  $\mathbf{D}$ -model of  $\mathcal{A}$ , and we write  $\mathbf{I} \models_{\mathbf{D}} \mathcal{A}$  if  $\mathbf{I}$  satisfies each assertion in  $\mathcal{A}$ , according to the semantic rules in [8, Table 1]. A  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -knowledge base  $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$  is *consistent* if there is an interpretation  $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$  that is a  $\mathbf{D}$ -model of  $\mathcal{A}$ ,  $\mathcal{T}$ , and  $\mathcal{R}$  (we write  $\mathbf{I} \models_{\mathbf{D}} \mathcal{K}$ ).

Some considerations on the expressive power of  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  are in order. As illustrated in [8, Table 1] existential quantification is admitted only on the left hand side of inclusion axioms. Thus  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is less powerful than logics such as  $\mathcal{SROIQ}(\mathbf{D})$  [11] as far as the generation of new individuals is concerned. On the other hand,  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is more liberal than  $\mathcal{SROIQ}(\mathbf{D})$  in the definition of role inclusion axioms since the roles involved are not required to be subject to any

ordering relationship, and the notion of simple role is not needed. For example, the role hierarchy presented in [11, p. 2] is not expressible in  $\mathcal{SROIQ}(\mathbf{D})$  but can be represented in  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ . In addition,  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is a powerful rule language able to express rules with negated atoms such as

$$Person(?p) \wedge \neg hasCar(?p, ?c) \implies CarlessPerson(?p).$$

Notice that rules with negated atoms are not supported by the SWRL language.

### 3 ABox Reasoning Services for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -Knowledge Base

The most important feature of a knowledge representation system is the capability of providing reasoning services. Depending on the type of the application domains, there are many different kinds of implicit knowledge that is desirable to infer from what is explicitly mentioned in the knowledge base. In particular, reasoning problems regarding ABoxes consist in querying a knowledge base in order to retrieve information concerning data stored in it. In this section we study the decidability for the most widespread ABox reasoning tasks for the logic  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  by resorting to a general problem, called *Higher Order Conjunctive Query Answering* (HOCQA), that can be instantiated to each of them.

Let  $V_i = \{v_1, v_2, \dots\}$ ,  $V_c = \{c_1, c_2, \dots\}$ ,  $V_{ar} = \{r_1, r_2, \dots\}$ , and  $V_{cr} = \{p_1, p_2, \dots\}$  be pairwise disjoint denumerably infinite sets of variables which are disjoint from  $\mathbf{Ind}$ ,  $\bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$ ,  $\mathbf{C}$ ,  $\mathbf{R}_{\mathbf{A}}$ , and  $\mathbf{R}_{\mathbf{D}}$ . A HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -atomic formula is an expression of one of the following types:  $R(w_1, w_2)$ ,  $P(w_1, u_1)$ ,  $C(w_1)$ ,  $r(w_1, w_2)$ ,  $p(w_1, u_1)$ ,  $c(w_1)$ ,  $w_1 = w_2$ ,  $u_1 = u_2$ , where  $w_1, w_2 \in V_i \cup \mathbf{Ind}$ ,  $u_1, u_2 \in V_i \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$ ,  $R$  is a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -abstract role term,  $P$  is a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concrete role term,  $C$  is a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -concept term,  $r \in V_{ar}$ ,  $p \in V_{cr}$ , and  $c \in V_c$ . A HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -atomic formula containing no variables is said to be *ground*. A HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -literal is a HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -atomic formula or its negation. A HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive query is a conjunction of HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -literals. We denote with  $\lambda$  the empty HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive query.

Let  $v_1, \dots, v_n \in V_i$ ,  $c_1, \dots, c_m \in V_c$ ,  $r_1, \dots, r_k \in V_{ar}$ ,  $p_1, \dots, p_h \in V_{cr}$ ,  $o_1, \dots, o_n \in \mathbf{Ind} \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$ ,  $C_1, \dots, C_m \in \mathbf{C}$ ,  $R_1, \dots, R_k \in \mathbf{R}_{\mathbf{A}}$ , and  $P_1, \dots, P_h \in \mathbf{R}_{\mathbf{D}}$ . A substitution

$$\sigma := \{v_1/o_1, \dots, v_n/o_n, c_1/C_1, \dots, c_m/C_m, r_1/R_1, \dots, r_k/R_k, p_1/P_1, \dots, p_h/P_h\}$$

is a map such that, for every HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -literal  $L$ ,  $L\sigma$  is obtained from  $L$  by replacing the occurrences of  $v_i$  in  $L$  with  $o_i$ , for  $i = 1, \dots, n$ ; the occurrences of  $c_j$  in  $L$  with  $C_j$ , for  $j = 1, \dots, m$ ; the occurrences of  $r_\ell$  in  $L$  with  $R_\ell$ , for  $\ell = 1, \dots, k$ ; the occurrences of  $p_t$  in  $L$  with  $P_t$ , for  $t = 1, \dots, h$ .

Substitutions can be extended to HO  $\mathcal{DL}_D^{4,\times}$ -conjunctive queries in the usual way. Let  $Q := (L_1 \wedge \dots \wedge L_m)$  be a HO  $\mathcal{DL}_D^{4,\times}$ -conjunctive query, and  $\mathcal{KB}$  a  $\mathcal{DL}_D^{4,\times}$ -knowledge base. A substitution  $\sigma$  involving *exactly* the variables occurring in  $Q$  is a *solution for  $Q$  w.r.t.  $\mathcal{KB}$*  if there exists a  $\mathcal{DL}_D^{4,\times}$ -interpretation  $\mathbf{I}$  such that  $\mathbf{I} \models_D \mathcal{KB}$  and  $\mathbf{I} \models_D Q\sigma$ . The collection  $\Sigma$  of the solutions for  $Q$  w.r.t.  $\mathcal{KB}$  is the *higher order answer set of  $Q$  w.r.t.  $\mathcal{KB}$* . Then the *higher order conjunctive query answering problem* for  $Q$  w.r.t.  $\mathcal{KB}$  consists in finding the HO answer set  $\Sigma$  of  $Q$  w.r.t.  $\mathcal{KB}$ . We shall solve the HOCQA problem just stated by reducing it to the analogous problem formulated in the context of the fragment  $4\text{LQS}^R$  (and in turn to the decision procedure for  $4\text{LQS}^R$  presented in [4]). The HOCQA problem for  $4\text{LQS}^R$ -formulae can be stated as follows. Let  $\phi$  be a  $4\text{LQS}^R$ -formula and let  $\psi$  be a conjunction of  $4\text{LQS}^R$ -quantifier-free atomic formulae of level 0 of the types  $x = y$ ,  $x \in X^1$ ,  $\langle x, y \rangle \in X^3$ , or their negations.

The *HOCQA problem for  $\psi$  w.r.t.  $\phi$*  consists in computing the HO *answer set of  $\psi$  w.r.t.  $\phi$* , namely the collection  $\Sigma'$  of all the substitutions  $\sigma'$  such that  $\mathcal{M} \models \phi \wedge \psi\sigma'$ , for some  $4\text{LQS}^R$ -interpretation  $\mathcal{M}$ .

In view of the decidability of the satisfiability problem for  $4\text{LQS}^R$ -formulae, the HOCQA problem for  $4\text{LQS}^R$ -formulae is decidable as well. Indeed, let  $\phi$  and  $\psi$  be two  $4\text{LQS}^R$ -formulae fulfilling the above requirements. To calculate the HO answer set of  $\psi$  w.r.t.  $\phi$ , for each candidate substitution

$$\sigma' := \{x/z, X^1/Y^1, X^2/Y^2, X^3/Y^3\}$$

one has just to check the  $4\text{LQS}^R$ -formula  $\phi \wedge \psi\sigma'$  for satisfiability. Since the number of possible candidate substitutions is  $|\text{Vars}(\phi)|^{|\text{Vars}(\psi)|}$  and the satisfiability problem for  $4\text{LQS}^R$ -formulae is decidable, the HO answer set of  $\psi$  w.r.t.  $\phi$  can be computed effectively. Summarizing,

**Lemma 1.** *The HOCQA problem for  $4\text{LQS}^R$ -formulae is decidable.*  $\square$

The following theorem states decidability of the HOCQA problem for  $\mathcal{DL}_D^{4,\times}$ .

**Theorem 1.** *Given a  $\mathcal{DL}_D^{4,\times}$ -knowledge base  $\mathcal{KB}$  and a HO  $\mathcal{DL}_D^{4,\times}$ -conjunctive query  $Q$ , the HOCQA problem for  $Q$  w.r.t.  $\mathcal{KB}$  is decidable.*  $\square$

The proof of Theorem 1 is much along the same lines of [7, Theorem 1] and, for space reasons, is omitted. However, the interested reader can find it in the extended version of this paper [8]. Here, we just sketch the main ideas of the proof to ease the understanding of the rest of the paper. As mentioned above, the  $\mathcal{DL}_D^{4,\times}$ -HOCQA problem can be solved by reducing it effectively to the HOCQA problem for  $4\text{LQS}^R$ -formulae and then exploiting Lemma 1. The reduction is carried out by means of a transformation function  $\theta$  that maps the  $\mathcal{DL}_D^{4,\times}$ -knowledge



base  $\mathcal{KB}$  in a  $4\text{LQS}^{\text{R}}$ -formula  $\phi_{\mathcal{KB}}$  in Conjunctive Normal Form (CNF) and the HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive query  $Q$  in the  $4\text{LQS}^{\text{R}}$ -formula  $\psi_Q$ . Specifically,<sup>2</sup>

$$\phi_{\mathcal{KB}} := \bigwedge_{H \in \mathcal{KB}} \theta(H) \wedge \bigwedge_{i=1}^{12} \xi_i, \quad \psi_Q := \theta(Q).$$

Let  $\Sigma$  be the HO answer set of  $Q$  w.r.t.  $\mathcal{KB}$  and  $\Sigma'$  the HO answer set of  $\psi_Q$  w.r.t.  $\phi_{\mathcal{KB}}$ . Then  $\Sigma$  consists of all substitutions  $\sigma$  (involving exactly the variables occurring in  $Q$ ) such that  $\theta(\sigma) \in \Sigma'$ . By Lemma 1,  $\Sigma'$  can be calculated effectively and thus  $\Sigma$  can be calculated effectively as well.

Next, we list the most widespread reasoning services for  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -ABox and then show how to define them as particular cases of the HOCQA task.

1. *Instance checking*: the problem of deciding whether or not an individual  $a$  is an instance of a concept  $C$ .
2. *Instance retrieval*: the problem of retrieving all the individuals that are instances of a given concept.
3. *Role filler retrieval*: the problem of retrieving all the fillers  $x$  such that the pair  $(a, x)$  is an instance of a role  $R$ .
4. *Concept retrieval*: the problem of retrieving all concepts which an individual is an instance of.
5. *Role instance retrieval*: the problem of retrieving all roles which a pair of individuals  $(a, b)$  is an instance of.

The instance checking problem is a specialisation of the HOCQA problem admitting HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries of the form  $Q_{IC} = C(w_1)$ , with  $w_1 \in \mathbf{Ind}$ . The instance retrieval problem is a particular case of the HOCQA problem in which HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries have the form  $Q_{IR} = C(w_1)$ , where  $w_1$  is a variable in  $V_i$ . The HOCQA problem can be instantiated to the role filler retrieval problem by admitting HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries  $Q_{RF} = R(w_1, w_2)$ , with  $w_1 \in \mathbf{Ind}$  and  $w_2$  a variable in  $V_i$ . The concept retrieval problem is a specialization of the HOCQA problem allowing HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries of the form  $Q_{QR} = c(w_1)$ , with  $w_1 \in \mathbf{Ind}$  and  $c$  a variable in  $V_c$ . Finally, the role instance retrieval problem is a particularization of the HOCQA problem, where HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries have the form  $Q_{RI} = r(w_1, w_2)$ , with  $w_1, w_2 \in \mathbf{Ind}$  and  $r$  a variable in  $V_{cr}$ .

Notice that the CQA problem for  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  defined in [7] is an instance of the HOCQA problem admitting HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries of the form  $(L_1 \wedge \dots \wedge L_m)$ , where the conjuncts  $L_i$  are atomic formulae of any of the types  $R(w_1, w_2)$ ,

<sup>2</sup> The map  $\theta$  coincides with the transformation function defined in [7] as far as it concerns the translation of each axiom or assertion  $H$  of  $\mathcal{KB}$  in a  $4\text{LQS}^{\text{R}}$ -formula  $\theta(H)$ . The map  $\theta$  extends the function introduced in [7] for what regards the translation of the HO query  $Q$  and of the substitutions  $\sigma$  of the HO answer set  $\Sigma$ . In particular it maps effectively variables in  $V_c$  in variables of sort 1 (in the language of  $4\text{LQS}^{\text{R}}$ ), and variables in  $V_{ar}$  and in  $V_{cr}$  in variables of sort 3. The constraints  $\xi_1$ – $\xi_{12}$  are added to make sure that each  $4\text{LQS}^{\text{R}}$ -model of  $\phi_{\mathcal{KB}}$  can be transformed into a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -interpretation (cf. [7, Theorem 1]).

$C(w_1)$ , and  $w_1 = w_2$  (or their negation), with  $w_1, w_2 \in (\mathbf{Ind} \cup \mathbf{V}_i)$ . Notice also that problems 1, 2, and 3 are instances of the CQA problem for  $\mathcal{DL}_D^{4,\times}$ , whereas problems 4 and 5 fall outside the definition of CQA. As shown above, they can be treated as specializations of HOCQA.

## 4 An Algorithm for the HOCQA Problem for $\mathcal{DL}_D^{4,\times}$

In this section we introduce an effective set-theoretic procedure to compute the answer set of a HO  $\mathcal{DL}_D^{4,\times}$ -conjunctive query  $Q$  w.r.t. a  $\mathcal{DL}_D^{4,\times}$  knowledge base  $\mathcal{KB}$ . Such procedure, called HOCQA- $\mathcal{DL}_D^{4,\times}$ , takes as input  $\phi_{\mathcal{KB}}$  (i.e., the  $4\text{LQS}^R$ -translation of  $\mathcal{KB}$ ) and  $\psi_Q$  (i.e., the  $4\text{LQS}^R$ -formula representing the HO  $\mathcal{DL}_D^{4,\times}$ -conjunctive query  $Q$ ), and returns a KE-tableau  $\mathcal{T}_{\mathcal{KB}}$ , representing the saturation of  $\mathcal{KB}$ , and the answer set  $\Sigma'$  of  $\psi_Q$  w.r.t.  $\phi_{\mathcal{KB}}$ , namely the collection of all substitutions  $\sigma'$  such that  $\mathcal{M} \models \phi_{\mathcal{KB}} \wedge \psi_Q \sigma'$ , for some  $4\text{LQS}^R$ -interpretation  $\mathcal{M}$ . Specifically, HOCQA- $\mathcal{DL}_D^{4,\times}$  constructs for each open branch of  $\mathcal{T}_{\mathcal{KB}}$  a decision tree whose leaves are labelled with elements of  $\Sigma'$ .

Let us first introduce some definitions and notations useful for the presentation of Procedure HOCQA- $\mathcal{DL}_D^{4,\times}$ .

Assume without loss of generality that universal quantifiers in  $\phi_{\mathcal{KB}}$  occur as inward as possible and that universally quantified variables are pairwise distinct. Let  $S_1, \dots, S_m$  be the conjuncts of  $\phi_{\mathcal{KB}}$  having the form of  $4\text{LQS}^R$ -purely universal formulae. For each  $S_i := (\forall z_1^i) \dots (\forall z_{n_i}^i) \chi_i$ , with  $i = 1, \dots, m$ , we put

$$\text{Exp}(S_i) := \bigwedge_{\{x_{a_1}, \dots, x_{a_{n_i}}\} \subseteq \text{Var}_0(\phi_{\mathcal{KB}})} S_i \{z_1^i/x_{a_1}, \dots, z_{n_i}^i/x_{a_{n_i}}\}.$$

We also define the *expansion*  $\Phi_{\mathcal{KB}}$  of  $\phi_{\mathcal{KB}}$  by putting

$$\Phi_{\mathcal{KB}} := \{F_j : j = 1, \dots, k\} \cup \bigcup_{i=1}^m \text{Exp}(S_i), \quad (1)$$

where  $F_1, \dots, F_k$  are the conjuncts of  $\phi_{\mathcal{KB}}$  having the form of  $4\text{LQS}^R$ -quantifier free atomic formulae.

To prepare for Procedure HOCQA- $\mathcal{DL}_D^{4,\times}$  to be described next, a brief introduction to KE-tableau systems is in order (see [10] for a detailed overview of KE-tableaux). KE-tableaux are a refutation system inspired to Smullyan's semantic tableaux [14]. The main characteristic distinguishing KE-tableaux from the latter is the introduction of an analytic cut rule (PB-rule) that permits to reduce inefficiencies of semantic tableaux. In fact, firstly, the classic tableau system can not represent the use of auxiliary lemmas in proofs; secondly, it can not express the bivalence of classical logic. Thirdly, it is highly inefficient, as witnessed by the fact that it can not polynomially simulate the truth-tables. None of these anomalies occurs if the cut rule is allowed. Procedure HOCQA- $\mathcal{DL}_D^{4,\times}$  constructs a complete KE-tableau  $\mathcal{T}_{\mathcal{KB}}$  for the expansion  $\Phi_{\mathcal{KB}}$  of  $\phi_{\mathcal{KB}}$  (cf. (1)), representing the saturation of the  $\mathcal{DL}_D^{4,\times}$ -knowledge base  $\mathcal{KB}$ .

Let  $\Phi := \{C_1, \dots, C_p\}$  be a collection of disjunctions of 4LQS<sup>R</sup>-quantifier free atomic formulae of level 0 of the types:  $x = y$ ,  $x \in X^1$ ,  $\langle x, y \rangle \in X^3$ .  $\mathcal{T}$  is a *KE-tableau* for  $\Phi$  if there exists a finite sequence  $\mathcal{T}_1, \dots, \mathcal{T}_t$  of trees such that (i)  $\mathcal{T}_1$  is a one-branch tree consisting of the sequence  $C_1, \dots, C_p$ , (ii)  $\mathcal{T}_t = \mathcal{T}$ , and (iii) for each  $i < t$ ,  $\mathcal{T}_{i+1}$  is obtained from  $\mathcal{T}_i$  either by an application of one of the rules in Fig. 1 or by applying a substitution  $\sigma$  to a branch  $\vartheta$  of  $\mathcal{T}_i$  (in particular, the substitution  $\sigma$  is applied to each formula  $X$  of  $\vartheta$ ; the resulting branch will be denoted by  $\vartheta\sigma$ ). The set of formulae  $\mathcal{S}_i^{\bar{\beta}} := \{\bar{\beta}_1, \dots, \bar{\beta}_n\} \setminus \{\bar{\beta}_i\}$  occurring as premise in the E-rule contains the complements of all the components of the formula  $\beta$  with the exception of the component  $\beta_i$ .

$$\begin{array}{c}
 \frac{\beta_1 \vee \dots \vee \beta_n \quad \mathcal{S}_i^{\bar{\beta}}}{\beta_i} \quad \mathbf{E\text{-}Rule} \\
 \text{where } \mathcal{S}_i^{\bar{\beta}} := \{\bar{\beta}_1, \dots, \bar{\beta}_n\} \setminus \{\bar{\beta}_i\}, \\
 \text{for } i = 1, \dots, n
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{}{A \mid \bar{A}} \quad \mathbf{PB\text{-}Rule} \\
 \text{with } A \text{ a literal}
 \end{array}$$

**Fig. 1.** Expansion rules for the KE-tableau.

Let  $\mathcal{T}$  be a KE-tableau. A branch  $\vartheta$  of  $\mathcal{T}$  is *closed* if it contains either both  $A$  and  $\neg A$ , for some formula  $A$ , or a literal of type  $\neg(x = x)$ . Otherwise, the branch is *open*. A KE-tableau is *closed* if all its branches are closed. A formula  $\beta_1 \vee \dots \vee \beta_n$  is *fulfilled* in a branch  $\vartheta$ , if  $\beta_i$  is in  $\vartheta$ , for some  $i = 1, \dots, n$ ; otherwise it is *unfulfilled*. A branch  $\vartheta$  is *fulfilled* if every formula  $\beta_1 \vee \dots \vee \beta_n$  occurring in  $\vartheta$  is fulfilled; otherwise it is *unfulfilled*. A branch  $\vartheta$  is *complete* if either it is closed or it is open, fulfilled, and it does not contain any literal of type  $x = y$ , with  $x, y$  distinct variables. A KE-tableau is *complete* (resp., *fulfilled*) if all its branches are complete (resp., fulfilled or closed). A 4LQS<sup>R</sup>-interpretation  $\mathcal{M}$  *satisfies* a branch  $\vartheta$  of a KE-tableau (or, equivalently,  $\vartheta$  is *satisfied* by  $\mathcal{M}$ ), and we write  $\mathcal{M} \models \vartheta$ , if  $\mathcal{M} \models X$  for every formula  $X$  occurring in  $\vartheta$ .

A 4LQS<sup>R</sup>-interpretation  $\mathcal{M}$  *satisfies* a KE-tableau  $\mathcal{T}$  (or, equivalently,  $\mathcal{T}$  is *satisfied* by  $\mathcal{M}$ ), and we write  $\mathcal{M} \models \mathcal{T}$ , if  $\mathcal{M}$  satisfies a branch  $\vartheta$  of  $\mathcal{T}$ . A branch  $\vartheta$  of a KE-tableau  $\mathcal{T}$  is *satisfiable* if there exists a 4LQS<sup>R</sup>-interpretation  $\mathcal{M}$  that satisfies  $\vartheta$ . A KE-tableau is *satisfiable* if at least one of its branches is satisfiable.

Let  $\vartheta$  be a branch of a KE-tableau. We denote with  $<_{\vartheta}$  an arbitrary but fixed total order on the variables in  $\text{Var}_0(\vartheta)$ .

Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}$  takes care of literals of type  $x = y$  occurring in the branches of  $\mathcal{T}_{KB}$  by constructing, for each open and fulfilled branch  $\vartheta$  of  $\mathcal{T}_{KB}$ , a substitution  $\sigma_{\vartheta}$  such that  $\vartheta\sigma_{\vartheta}$  does not contain literals of type  $x = y$  with distinct  $x, y$ . Then, for every open and complete branch  $\vartheta' := \vartheta\sigma_{\vartheta}$  of  $\mathcal{T}_{KB}$ , Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}$  constructs a decision tree  $\mathcal{D}_{\vartheta'}$  such that every maximal branch of  $\mathcal{D}_{\vartheta'}$  induces a substitution  $\sigma'$  such that  $\sigma_{\vartheta}\sigma'$  belongs to the answer set of  $\psi_Q$  with respect to  $\phi_{KB}$ .  $\mathcal{D}_{\vartheta'}$  is defined as follows.

Let  $d$  be the number of literals in  $\psi_Q$ . Then  $\mathcal{D}_{\vartheta'}$  is a finite labelled tree of depth  $d + 1$  whose labelling satisfies the following conditions, for  $i = 0, \dots, d$ :

- (i) every node of  $\mathcal{D}_{\vartheta'}$  at level  $i$  is labelled with  $(\sigma'_i, \psi_Q \sigma_{\vartheta} \sigma'_i)$ ; in particular, the root is labelled with  $(\sigma'_0, \psi_Q \sigma_{\vartheta} \sigma'_0)$ , where  $\sigma'_0$  is the empty substitution;
- (ii) if a node at level  $i$  is labelled with  $(\sigma'_i, \psi_Q \sigma_{\vartheta} \sigma'_i)$ , then its  $s$  successors, with  $s \geq 1$ , are labelled with  $(\sigma'_i \varrho_1^{q_{i+1}}, \psi_Q \sigma_{\vartheta} (\sigma'_i \varrho_1^{q_{i+1}})), \dots, (\sigma'_i \varrho_s^{q_{i+1}}, \psi_Q \sigma_{\vartheta} (\sigma'_i \varrho_s^{q_{i+1}}))$ , where  $q_{i+1}$  is the  $(i+1)$ -st conjunct of  $\psi_Q \sigma_{\vartheta} \sigma'_i$  and  $\mathcal{S}_{q_{i+1}} := \{\varrho_1^{q_{i+1}}, \dots, \varrho_s^{q_{i+1}}\}$  is the collection of the substitutions  $\varrho = \{v_1/o_1, \dots, v_n/o_n, c_1/C_1, \dots, c_m/C_m, r_1/R_1, \dots, r_k/R_k, p_1/P_1, \dots, p_h/P_h\}$ , with  $\{v_1, \dots, v_n\} = \text{Var}_0(q_{i+1})$ ,  $\{c_1, \dots, c_m\} = \text{Var}_1(q_{i+1})$ , and  $\{p_1, \dots, p_h, r_1, \dots, r_k\} = \text{Var}_3(q_{i+1})$ , such that  $t = q_{i+1}\varrho$ , for some literal  $t$  on  $\vartheta'$ . If  $s = 0$ , the node labelled with  $(\sigma'_i, \psi_Q \sigma_{\vartheta} \sigma'_i)$  is a leaf node and, if  $i = d$ ,  $\sigma_{\vartheta} \sigma'_i$  is added to  $\Sigma'$ .

We are now ready to define Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathcal{D}}^{4,\times}$ .

```

1: procedure HOCQA- $\mathcal{DL}_{\mathcal{D}}^{4,\times}(\psi_Q, \phi_{\mathcal{KB}})$ ;
2:    $\Sigma' := \emptyset$ ;
3:   - let  $\Phi_{\mathcal{KB}}$  be the expansion of  $\phi_{\mathcal{KB}}$  (cf. (1));
4:    $\mathcal{T}_{\mathcal{KB}} := \Phi_{\mathcal{KB}}$ ;
5:   while  $\mathcal{T}_{\mathcal{KB}}$  is not fulfilled do
6:     - select an unfulfilled open branch  $\vartheta$  of  $\mathcal{T}_{\mathcal{KB}}$  and an unfulfilled formula
        $\beta_1 \vee \dots \vee \beta_n$  in  $\vartheta$ ;
7:     if  $S_j^{\bar{\beta}}$  is in  $\vartheta$ , for some  $j \in \{1, \dots, n\}$  then
8:       - apply the E-Rule to  $\beta_1 \vee \dots \vee \beta_n$  and  $S_j^{\bar{\beta}}$  on  $\vartheta$ ;
9:     else
10:      - let  $B^{\bar{\beta}}$  be the collection of the formulae present in  $\vartheta$  and let
         $h$  be the lowest index such that  $\bar{\beta}_h \notin B^{\bar{\beta}}$ ;
11:      - apply the PB-rule to  $\bar{\beta}_h$  on  $\vartheta$ ;
12:    end if;
13:  end while;
14:
while  $\mathcal{T}_{\mathcal{KB}}$  has open branches containing literals of type  $x = y$ , with distinct  $x$  and  $y$  do
15:   - select such an open branch  $\vartheta$  of  $\mathcal{T}_{\mathcal{KB}}$ ;
16:    $\sigma_{\vartheta} := \epsilon$  (where  $\epsilon$  is the empty substitution);
17:    $\text{Eq}_{\vartheta} := \{\text{literals of type } x = y \text{ occurring in } \vartheta\}$ ;
18:   while  $\text{Eq}_{\vartheta}$  contains  $x = y$ , with distinct  $x, y$  do
19:     - select a literal  $x = y$  in  $\text{Eq}_{\vartheta}$ , with distinct  $x, y$ ;
20:      $z := \min_{<_{\vartheta}}(x, y)$ ;
21:      $\sigma_{\vartheta} := \sigma_{\vartheta} \cdot \{x/z, y/z\}$ ;
22:      $\text{Eq}_{\vartheta} := \text{Eq}_{\vartheta} \sigma_{\vartheta}$ ;
23:   end while;
24:    $\vartheta := \vartheta \sigma_{\vartheta}$ ;
25:   if  $\vartheta$  is open then
26:     - initialize  $\mathcal{S}$  to the empty stack;
27:     - push  $(\epsilon, \psi_Q \sigma_{\vartheta})$  in  $\mathcal{S}$ ;
28:     while  $\mathcal{S}$  is not empty do
29:       - pop  $(\sigma', \psi_Q \sigma_{\vartheta} \sigma')$  from  $\mathcal{S}$ ;
30:       if  $\psi_Q \sigma_{\vartheta} \sigma' \neq \lambda$  then
31:         - let  $q$  be the leftmost conjunct of  $\psi_Q \sigma_{\vartheta} \sigma'$ ;
32:          $\psi_Q \sigma_{\vartheta} \sigma' := \psi_Q \sigma_{\vartheta} \sigma'$  deprived of  $q$ ;
33:          $\text{Lit}_Q^M := \{t \in \vartheta : t = q\rho, \text{ for some substitution } \rho\}$ ;
34:         while  $\text{Lit}_Q^M$  is not empty do
35:           - let  $t \in \text{Lit}_Q^M, t = q\rho$ ;
36:            $\text{Lit}_Q^M := \text{Lit}_Q^M \setminus \{t\}$ ;
37:           - push  $(\sigma' \rho, \psi_Q \sigma_{\vartheta} \sigma' \rho)$  in  $\mathcal{S}$ ;
38:         end while;
39:       else
40:          $\Sigma' := \Sigma' \cup \{\sigma_{\vartheta} \sigma'\}$ ;
41:       end if;
42:     end while;
43:   end if;
44: end while;
45: return  $(\mathcal{T}_{\mathcal{KB}}, \Sigma')$ ;
46: end procedure;

```

$\{\bar{\beta}_1, \dots, \bar{\beta}_n\}$  that is included in  $\vartheta$

For each open branch  $\vartheta$  of  $\mathcal{T}_{\mathcal{KB}}$ , Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}$  computes the corresponding  $\mathcal{D}_\vartheta$  by constructing a stack of its nodes. Initially the stack contains the root node  $(\epsilon, \psi_Q \sigma_\vartheta)$  of  $\mathcal{D}_\vartheta$ , as defined in condition (i). Then, iteratively, the following steps are executed. An element  $(\sigma', \psi_Q \sigma_\vartheta \sigma')$  is popped out of the stack. If the last literal of the query  $\psi_Q$  has not been reached, the successors of the current node are computed according to condition (ii) and inserted in the stack. Otherwise the current node must have the form  $(\sigma', \lambda)$  and the substitution  $\sigma_\vartheta \sigma'$  is inserted in  $\Sigma'$ .

Correctness of Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}$  follows from Theorem 2, which show that  $\phi_{\mathcal{KB}}$  is satisfiable if and only if  $\mathcal{T}_{\mathcal{KB}}$  is a non-closed KE-tableau, and from Theorem 3, which shows that the set  $\Sigma'$  coincides with the HO answer set of  $\psi_Q$  w.r.t.  $\phi_{\mathcal{KB}}$ . Theorems 2 and 3 are stated below. In particular, Theorem 2, requires the following technical lemmas.

**Lemma 2.** *Let  $\vartheta$  be a branch of  $\mathcal{T}_{\mathcal{KB}}$  selected at step 15 of Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}(\psi_Q, \phi_{\mathcal{KB}})$ , let  $\sigma_\vartheta$  be the associated substitution constructed during the execution of the while-loop 18–23, and let  $\mathcal{M} = (D, M)$  be a  $4\text{LQS}^{\text{R}}$ -interpretation satisfying  $\vartheta$ . Then, for every  $x \in \text{Var}_0(\vartheta)$ ,  $Mx = Mx\sigma_\vartheta$  is an invariant of the while-loop 18–23.  $\square$*

**Lemma 3.** *Let  $\mathcal{T}_0, \dots, \mathcal{T}_h$  be a sequence of KE-tableaux such that  $\mathcal{T}_0 = \Phi_{\mathcal{KB}}$ , and where, for  $i = 1, \dots, h-1$ ,  $\mathcal{T}_{i+1}$  is obtained from  $\mathcal{T}_i$  by applying either the rule of step 8, or the rule of step 10, or the substitution of step 24 of Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}(\psi_Q, \phi_{\mathcal{KB}})$ . If  $\mathcal{T}_i$  is satisfied by a  $4\text{LQS}^{\text{R}}$ -interpretation  $\mathcal{M}$ , then  $\mathcal{T}_{i+1}$  is satisfied by  $\mathcal{M}$  as well, for  $i = 1, \dots, h-1$ .  $\square$*

Then we have:

**Theorem 2.** *The formula  $\phi_{\mathcal{KB}}$  is satisfiable if and only if the tableau  $\mathcal{T}_{\mathcal{KB}}$  is not closed.  $\square$*

The proof of Theorem 3 below requires the following technical lemma.

**Lemma 4.** *Let  $\psi_Q := q_1 \wedge \dots \wedge q_d$  be a HO  $4\text{LQS}^{\text{R}}$ -conjunctive query,  $(\mathcal{T}_{\mathcal{KB}}, \Sigma')$  the output of  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}(\psi_Q, \phi_{\mathcal{KB}})$ , and  $\vartheta$  an open and complete branch of  $\mathcal{T}_{\mathcal{KB}}$ . Then, for any substitution  $\sigma$ , we have*

$$\sigma \in \Sigma' \iff \{q_1\sigma, \dots, q_d\sigma\} \subseteq \vartheta. \quad \square$$

**Theorem 3.** *Let  $\Sigma'$  be the set of substitutions returned by Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}(\psi_Q, \phi_{\mathcal{KB}})$ . Then  $\Sigma'$  is the HO answer set of  $\psi_Q$  w.r.t.  $\phi_{\mathcal{KB}}$ .*

Due to space limitations, we do not include here the proofs of Theorems 2 and 3 and of Lemmas 2, 3, and 4, which can be found in the extended version of the paper [8].

Termination of Procedure  $\text{HOCQA-}\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is based on the fact that the while-loops 5–13 and 14–44 terminate. Termination of the while-loop 5–13 can be shown much in the same way as for Procedure 1 in [7].

Concerning the **while-loop** 14–44, its termination can be proved by observing that the number of branches of the KE-tableau resulting from the execution of the previous **while-loop** 5–13 is finite and then showing that the internal **while-loops** 18–23 and 28–42 always terminate. Indeed, initially the set  $\text{Eq}_\vartheta$  contains a finite number of literals of type  $x = y$ , and  $\sigma_\vartheta$  is the empty substitution. It is then enough to show that the number of literals of type  $x = y$  in  $\text{Eq}_\vartheta$ , with distinct  $x$  and  $y$ , strictly decreases during the execution of the internal **while-loop** 18–23. But this follows immediately, since at each of its iterations one puts  $\sigma_\vartheta := \sigma_\vartheta \cdot \{x/z, y/z\}$ , with  $z := \min_{<_\vartheta}(x, y)$ , according to a fixed total order  $<_\vartheta$  over the variables of  $\text{Var}_0(\vartheta)$  and then the application of  $\sigma_\vartheta$  to  $\text{Eq}_\vartheta$  replaces a literal of type  $x = y$  in  $\text{Eq}_\vartheta$ , with distinct  $x$  and  $y$ , with a literal of type  $x = x$ .

The **while-loop** 28–42 terminates when the stack  $\mathcal{S}$  of the nodes of the decision tree gets empty. Since the query  $\psi_Q$  contains a finite number of conjuncts and the number of literals on each open and complete branch of  $\mathcal{T}_{\mathcal{KB}}$  is finite, the number of possible matches (namely the size of the set  $\text{Lit}_Q^M$ ) computed at step 33 is finite as well. Thus, in particular, the internal **while-loop** 34–38 terminates at each execution. Once the procedure has processed the last conjunct of the query, the set  $\text{Lit}_Q^M$  of possible matches is empty and thus no element gets pushed in the stack  $\mathcal{S}$  anymore. Since the first instruction of the **while-loop** 28–42 removes an element from  $\mathcal{S}$ , the stack gets empty after a finite number of “pops”. Hence Procedure  $\text{HOCQA-}\mathcal{DL}_D^{4,\times}$  terminates, as we wished to prove.

We provide now some complexity results. Let  $r$  be the maximum number of universal quantifiers in each  $S_i$  ( $i = 1, \dots, m$ ), and put  $k := |\text{Var}_0(\phi_{\mathcal{KB}})|$ . Then, each  $S_i$  generates at most  $k^r$  expansions. Since the knowledge base contains  $m$  such formulae, the number of disjunctions in the initial branch of the KE-tableau is bounded by  $m \cdot k^r$ . Next, let  $\ell$  be the maximum number of literals in each  $S_i$ . Then, the height of the KE-tableau (which corresponds to the maximum size of the models of  $\Phi_{\mathcal{KB}}$  constructed as illustrated above) is  $\mathcal{O}(\ell m k^r)$  and the number of leaves of the tableau (namely, the number of such models of  $\Phi_{\mathcal{KB}}$ ) is  $\mathcal{O}(2^{\ell m k^r})$ . Notice that the construction of  $\text{Eq}_\vartheta$  and of  $\sigma_\vartheta$  in the lines 16–23 of Procedure  $\text{HOCQA-}\mathcal{DL}_D^{4,\times}$  takes  $\mathcal{O}(\ell m k^r)$  time, for each branch  $\vartheta$ .

Let  $\eta(\mathcal{T}_{\mathcal{KB}})$  and  $\lambda(\mathcal{T}_{\mathcal{KB}})$  be, respectively, the height of  $\mathcal{T}_{\mathcal{KB}}$  and the number of leaves of  $\mathcal{T}_{\mathcal{KB}}$  computed by Procedure  $\text{HOCQA-}\mathcal{DL}_D^{4,\times}$ . Plainly,  $\eta(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(\ell m k^r)$  and  $\lambda(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(2^{\ell m k^r})$ , as computed above. It is easy to verify that  $s = \mathcal{O}(\ell k^r)$  is the maximum branching of  $\mathcal{D}_\vartheta$ . Since the height of  $\mathcal{D}_\vartheta$  is  $h$ , where  $h$  is the number of literals in  $\psi_Q$ , and the successors of a node are computed in  $\mathcal{O}(\ell k^r)$  time, the number of leaves in  $\mathcal{D}_\vartheta$  is  $\mathcal{O}(s^h) = \mathcal{O}((\ell k^r)^h)$  and they are computed in  $\mathcal{O}(s^h \cdot \ell k^r \cdot h) = \mathcal{O}(h \cdot (\ell k^r)^{h+1})$  time. Finally, since we have  $\lambda(\mathcal{T}_{\mathcal{KB}})$  of such decision trees, the answer set of  $\psi_Q$  w.r.t.  $\phi_{\mathcal{KB}}$  is computed in time  $\mathcal{O}(h \cdot (\ell k^r)^{h+1} \cdot \lambda(\mathcal{T}_{\mathcal{KB}})) = \mathcal{O}(h \cdot (\ell k^r)^{h+1} \cdot 2^{\ell m k^r})$ .

Since the size of  $\phi_{\mathcal{KB}}$  and of  $\psi_Q$  are related to those of  $\mathcal{KB}$  and of  $Q$ , respectively (see the proof of Theorem 1 in [8] for details on the reduction), the construction of the HO answer set of  $Q$  with respect to  $\mathcal{KB}$  can be done in double-exponential time. In case  $\mathcal{KB}$  contains neither role chain axioms nor qualified cardinality restrictions, the complexity of our *HOCQA* problem is in

EXPTIME, since the maximum number of universal quantifiers in  $\phi_{\mathcal{KB}}$ , namely  $r$ , is a constant (in particular  $r = 3$ ). The latter complexity result is a clue to the fact that the HOCQA problem is intrinsically more difficult than the consistency problem (proved to be NP-complete in [3]). In fact, the consistency problem simply requires to guess a model of the knowledge base, whereas the HOCQA problem forces the construction of all the models of the knowledge base and the computation of a decision tree for each of them.

We remark that such result compares favourably with the complexity of the usual CQA problem for a wide collection of description logics such as the Horn fragment of *SHOIQ* and of *SROIQ*, which are EXPTIME- and 2EXPTIME-complete respectively (see [13] for details).

## 5 Conclusions and Future Work

We have considered an extension of the CQA problem for the description logic  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  to more general queries on roles and concepts. The resulting problem, called HOCQA, can be instantiated to the most widespread ABox reasoning services such as instance retrieval, role filler retrieval, and instance checking. We have proved the decidability of the HOCQA problem by reducing it to the satisfiability problem for the set-theoretic fragment  $4\text{LQS}^{\mathbf{R}}$ .

We have introduced an algorithm to compute the HO answer set of a  $4\text{LQS}^{\mathbf{R}}$ -formula  $\psi_Q$  representing a HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive query  $Q$  w.r.t. a  $4\text{LQS}^{\mathbf{R}}$ -formula  $\phi_{\mathcal{KB}}$  representing a  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  knowledge base. Our procedure, called HOCQA- $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ , is based on the KE-tableau system and on decision trees. It takes as input  $\psi_Q$  and  $\phi_{\mathcal{KB}}$ , and yields a KE-tableau  $\mathcal{T}_{\mathcal{KB}}$  representing the saturation of  $\phi_{\mathcal{KB}}$  and the requested HO answer set  $\Sigma'$ . Procedure HOCQA- $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  is proved correct and complete, and some complexity results are provided. Such procedure extends the one introduced in [7] as it handles HO  $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ -conjunctive queries.

We are currently implementing Procedure HOCQA- $\mathcal{DL}_{\mathbf{D}}^{4,\times}$  and plan to increase its efficiency by providing a parallel model and enhancing the expansion rules. We also intend to allow data types reasoning.

Further, we plan to extend the fragment presented in [4] with a restricted form of composition operator, since this would allow one to represent various logics in set-theoretic terms. The KE-tableau based procedure will be adapted to the new set-theoretic fragments exploiting the techniques introduced in [5, 6] in the area of relational dual tableaux.

## References

1. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: PODS, pp. 149–158 (1998)
2. Calvanese, D., De Giacomo, G., Lenzerini, M.: Conjunctive query containment and answering under description logic constraints. *ACM Trans. Comput. Log.* **9**(3), 22:1–31 (2008)
3. Cantone, D., Longo, C., Nicolosi-Asmundo, M., Santamaria, D.F.: Web ontology representation and reasoning via fragments of set theory. In: Cate, B., Mileo, A. (eds.) RR 2015. LNCS, vol. 9209, pp. 61–76. Springer, Cham (2015). doi:[10.1007/978-3-319-22002-4\\_6](https://doi.org/10.1007/978-3-319-22002-4_6)
4. Cantone, D., Nicolosi-Asmundo, M.: On the satisfiability problem for a 4-level quantified syllogistic and some applications to modal logic. *Fundamenta Informaticae* **124**(4), 427–448 (2013)
5. Cantone, D., Nicolosi-Asmundo, M., Orłowska, E.: Dual tableau-based decision procedures for some relational logics. In: Proceedings of the 25th Italian Conference on Computational Logic, CEUR-WS, vol. 598, Rende, Italy, 7–9 July 2010 (2010)
6. Cantone, D., Nicolosi-Asmundo, M., Orłowska, E.: Dual tableau-based decision procedures for relational logics with restricted composition operator. *J. Appl. Non-Classical Logics* **21**(2), 177–200 (2011)
7. Cantone, D., Nicolosi-Asmundo, M., Santamaria, D.F.: Conjunctive query answering via a fragment of set theory. In: Proceedings of ICTCS 2016, CEUR-WS, Lecce, 7–9 September, vol. 1720, pp. 23–35 (2016)
8. Cantone, D., Nicolosi-Asmundo, M., Santamaria, D.F.: A set-theoretic approach to ABox reasoning services. CoRR, 1702.03096 (2017). Extended version
9. Cantone, D., Nicolosi-Asmundo, M., Santamaria, D.F., Trapani, F.: Ontoceramic: an OWL ontology for ceramics classification. In Proceedings of CILC 2015, CEUR-WS, vol. 1459, pp. 122–127, Genova, 1–3 July 2015
10. D’Agostino, M.: Tableau methods for classical propositional logic. In: D’Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds.) *Handbook of Tableau Methods*, pp. 45–123. Springer (1999)
11. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible *SR<sub>OIQ</sub>*. In: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, pp. 57–67. AAAI Press (2006)
12. Motik, B., Horrocks, I.: OWL datatypes: design and implementation. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 307–322. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88564-1\\_20](https://doi.org/10.1007/978-3-540-88564-1_20)
13. Ortiz, M., Sebastian, R., Šimkus, M.: Query answering in the Horn fragments of the description logics *SH<sub>OIQ</sub>* and *SR<sub>OIQ</sub>*. In: Proceedings of the 22th International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 2, pp. 1039–1044. AAAI Press (2011)
14. Smullyan, R.M.: *First-Order Logic*. Springer, Heidelberg (1968)