A set-based reasoner for the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Domenico Cantone, Marianna Nicolosi-Asmundo, and Daniele Francesco Santamaria

University of Catania, Dept. of Mathematics and Computer Science
email: {cantone,nicolosi,santamaria}@dmi.unict.it

Abstract. We present a KE-tableau-based implementation of a reasoner for a decidable fragment of (stratified) set theory expressing the description logic $\mathcal{DL}\langle\mathsf{4LQS}^{\mathsf{R},\times}\rangle(\mathbf{D})$ ($\mathcal{DL}_{\mathbf{D}}^{4,\times}$, for short). Our application solves the main TBox and ABox reasoning problems for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. In particular, it solves the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge bases represented in set-theoretic terms, and a generalization of the *Conjunctive Query Answering* problem in which conjunctive queries with variables of three sorts are admitted. The reasoner, which extends and optimizes a previous prototype for the consistency checking of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge bases (see [7]), is implemented in C++. It supports $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge bases serialized in the OWL/XML format, and it admits also rules expressed in SWRL (Semantic Web Rule Language).

1 Introduction

A wealth of decidability results has been collected over the years within the research field of *Computable Set Theory* [2, 11, 17]. However, only recently some of these results have been applied in the context of knowledge representation and reasoning for the semantic web. Such efforts have been motivated by the characteristics of the set-theoretic fragments considered, as they provide very expressive unique formalisms that combine the modelling capabilities of a rule language with the constructs of description logics. The decidable multi-sorted quantified set-theoretic fragment $\mathsf{4LQS}^{\mathsf{R}}$ [3] is appropriate in this sense, in consideration of the fact that its decision procedure is efficiently implementable. We recall that the language of $\mathsf{4LQS}^{\mathsf{R}}$ involves variables of four sorts, pair terms, and a restricted form of quantification.

In [5], the theory $\mathsf{4LQS}^{\mathsf{R}}$ has been used to represent the expressive description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ by means of a suitable translation mapping. Moreover, decidability of the most widespread reasoning problems for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, such as the consistency problem and the Conjunctive Query Answering (CQA) problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge bases (KBs) were proved via a reduction to the satisfiability problem for $\mathsf{4LQS}^{\mathsf{R}}$. Since $\mathsf{4LQS}^{\mathsf{R}}$ admits variables of four sorts, the CQA problem was generalized in such a way as to admit queries over three sorts of variables. Such a generalization, called Higher-Order Conjunctive Query Answering (HOCQA) problem can be instantiated to the most widespread reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABox.

The description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ admits Boolean operators on concepts and abstract roles, concept domain and range, and existential and minimum cardinality restriction on the left-hand side of inclusion axioms. It also supports role chains on the left-hand side of inclusion axioms and properties on roles such as transitivity, symmetry, and reflexivity. In [4], its consistency problem has been shown to be NP-complete under not very restrictive constraints. Such a low complexity result depends on the fact that existential quantification cannot appear on the right-hand side of inclusion axioms. Nonetheless, $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ turns out to be more expressive than other low complexity logics such as OWL RL [16] and therefore it is very suitable for representing real-world ontologies. For instance, the restricted version of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ mentioned above allows one to express several OWL ontologies, such as ArcheOntology [16] and OntoCeramic [10], for the classification of archaeological finds, and ArchivioMuseoFabbrica [1], concerning the renovation of the Monastery of San Nicola l'Arena in Catania by the architect Giancarlo De Carlo. Since existential quantification is admitted only on the left-hand side of inclusion axioms, $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is less expressive than logics such as $\mathcal{SROIQ}(\mathbf{D})$ [13] as long 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, as the roles involved in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ are not subject to any ordering relationship, and the notion of simple role is not needed. For example, the role hierarchy presented in [13, page 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

$$Person(?p) \wedge \neg hasHome(?p, ?h) \implies HomelessPerson(?p)$$

that are not supported by the SWRL language.

In [7], we presented a first effort to implement in C++ a KE-tableau-based decision procedure for the consistency problem of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs, by resorting to the algorithm introduced in [5]. The choice of KE-tableau systems [14], instead of traditional semantic tableaux [19], was motivated by the fact that KE-tableau systems introduce an analytic cut rule which permits to construct trees whose branches define mutually exclusive situations, thus avoiding the proliferation of redundant branches, typical of Smullyan's semantic tableaux [19]. Thus, given as input a consistent KB, the procedure yields a KE-tableau whose open branches induce distinct models of the KB. Otherwise, a closed KE-tableau is returned.

In this contribution we improve the reasoner presented in [7] by introducing a system called KE$^{\gamma}$-tableau which admits a generalization of the KE-elimination rule incorporating the γ-rule, namely the expansion rule for handling universally quantified formulae. The reasoner also includes a procedure to compute the HOCQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Finally, through suitable benchmark tests, we show that such a novel reasoner is more efficient than the one introduced in [7].

2 Preliminaries

2.1 The set-theoretic fragment

We summarize the set-theoretic notions underpinning the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and its reasoning tasks. For the sake of conciseness, we avoid to report here the syntax and semantics of the whole $\mathsf{4LQS^R}$ theory (the interested reader can find it in [3] together with the decision procedure for its satisfiability problem). Thus, we focus on the $\mathsf{4LQS^R}$-formulae *de facto* involved in the set-theoretic representation of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, namely propositional combinations of $\mathsf{4LQS^R}$-literals (atomic formulae or their negations) and $\mathsf{4LQS^R}$ purely universal formulae of the types displayed in Table 1. The class of such $\mathsf{4LQS^R}$-formulae is called $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$.

We recall that the fragment $\mathsf{4LQS^R}$ admits four collections, $\mathtt{Var}_i$, of variables of sort i denoted by $X^i, Y^i, Z^i, \ldots$, for $i = 0, 1, 2, 3$ (variables of sort 0 are also denoted by $x, y, z, \ldots$). Besides variables, also *pair terms* of the form $\langle x, y\rangle$, with $x, y \in \mathtt{Var}_0$, are allowed. Since the types of formulae displayed in Table 1 do not contain variables of sort 2, here we limit ourselves only to notions and definitions relative to $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae involving variables of sorts $0, 1$, and 3.

Literals of level 0	Purely universal quantified formulae of level 1
$x = y,\ x \in X^1,\ \langle x, y\rangle \in X^3$ $\neg(x = y),\ \neg(x \in X^1),\ \neg(\langle x, y\rangle \in X^3)$	$(\forall z_1)\ldots(\forall z_n)\varphi_0$, where $z_1, \ldots, z_n \in \mathtt{Var}_0$ and φ_0 is any propositional combination of literals of level 0.

Table 1. Types of literals and quantified formulae admitted in $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$.

The variables $z_1, \ldots, z_n$ are said to occur *quantified* in $(\forall z_1)\ldots(\forall z_n)\varphi_0$. A variable occurs *free* in a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula φ if it does not occur quantified in any subformula of φ. For $i = 0, 1, 3$, we denote with $\mathtt{Var}_i(\varphi)$ the collections of variables of sort i occurring free in φ.

Given sequences of distinct variables $\boldsymbol{x}$ (in $\mathtt{Var}_0$), $\boldsymbol{X}^1$ (in $\mathtt{Var}_1$), and $\boldsymbol{X}^3$ (in $\mathtt{Var}_3$), of length n, m, and q, respectively, and sequences of (not necessarily distinct) variables $\boldsymbol{y}$ (in $\mathtt{Var}_0$), $\boldsymbol{Y}^1$ (in $\mathtt{Var}_1$), and $\boldsymbol{Y}^3$ (in $\mathtt{Var}_3$), also of length n, m, and q, respectively, the $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-substitution $\sigma \coloneqq \{\boldsymbol{x}/\boldsymbol{y}, \boldsymbol{X}^1/\boldsymbol{Y}^1, \boldsymbol{X}^3/\boldsymbol{Y}^3\}$ is the mapping $\varphi \mapsto \varphi\sigma$ such that, for any given universal quantified $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula φ, $\varphi\sigma$ is the result of replacing in φ the free occurrences of the variables x_i in $\boldsymbol{x}$ (for $i = 1, \ldots, n$) with the corresponding y_i in $\boldsymbol{y}$, of X_j^1 in $\boldsymbol{X}^1$ (for $j = 1, \ldots, m$) with Y_j^1 in $\boldsymbol{Y}^1$, and of X_h^3 in $\boldsymbol{X}^3$ (for $h = 1, \ldots, q$) with Y_h^3 in $\boldsymbol{Y}^3$, respectively. A substitution σ is *free* for φ if the formulae φ and $\varphi\sigma$ have exactly the same occurrences of quantified variables. The *empty substitution*, denoted ϵ, satisfies $\varphi\epsilon = \varphi$, for each $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula φ.

A $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-*interpretation* is a pair $\boldsymbol{\mathcal{M}} = (D, M)$, where D is a nonempty collection of objects (called *domain* or *universe* of $\boldsymbol{\mathcal{M}}$) and M is an assignment over the variables in $\mathtt{Var}_i$, for $i = 0, 1, 3$, such that:

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

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

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

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

Next, let

- $\boldsymbol{\mathcal{M}} = (D, M)$ be a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-interpretation,
- $x_1, \ldots, x_n \in \mathtt{Var}_0$, and
- $u_1, \ldots, u_n \in D$.

By $\boldsymbol{\mathcal{M}}[\boldsymbol{x}/\boldsymbol{u}]$, we denote the interpretation $\boldsymbol{\mathcal{M}}' = (D, M')$ such that $M'x_i = u_i$ (for $i = 1, \ldots, n$), and which otherwise coincides with M on all remaining variables. For a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-interpretation $\boldsymbol{\mathcal{M}} = (D, M)$ and a formula φ, the satisfiability relationship $\boldsymbol{\mathcal{M}} \models \varphi$ is recursively defined over the structure of φ as follows. Literals are evaluated in a standard way, based on the usual interpretation of the predicates '$\in$' and '$=$', and of the propositional negation '$\neg$'. Compound formulae are interpreted according to the standard rules of propositional logic. Finally, purely universal formulae are evaluated as follows:

- $\boldsymbol{\mathcal{M}} \models (\forall z_1) \ldots (\forall z_n)\varphi_0$ iff $\boldsymbol{\mathcal{M}}[\boldsymbol{z}/\boldsymbol{u}] \models \varphi_0$, for all $\boldsymbol{u} \in D^n$.

If $\boldsymbol{\mathcal{M}} \models \varphi$, then $\boldsymbol{\mathcal{M}}$ is said to be a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-model for φ. A $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula is said to be *satisfiable* if it has a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-model. A $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula is *valid* if it is satisfied by all $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-interpretations.

2.2 The logic $\mathcal{DL}\langle\mathsf{4LQS}^{\mathsf{R},\times}\rangle(\mathbf{D})$

It is convenient to recall the main notions and definitions concerning the description logic $\mathcal{DL}\langle\mathsf{4LQS}^{\mathsf{R},\times}\rangle(\mathbf{D})$ (also called $\mathcal{DL}^{4,\times}_{\mathbf{D}}$) [4].

Let $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{C}$, and $\mathbf{Ind}$ 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.

Data types are introduced through the notion of data type maps, defined according to [15] as follows. A *data type map* is a quadruple $\mathbf{D} = (N_D, N_C, N_F, \cdot^{\mathbf{D}})$, 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 (i) a function assigning a data type interpretation $d^{\mathbf{D}}$ to each data type $d \in N_D$, (ii) a facet interpretation $f^{\mathbf{D}} \subseteq d^{\mathbf{D}}$ to each facet $f \in N_F(d)$, and (iii) a data value $e_d^{\mathbf{D}} \in d^{\mathbf{D}}$ to every constant $e_d \in N_C(d)$. Facets determine subsets of data values considered of interest in a specific application domain. We shall assume that the interpretations of the data types in N_D are nonempty pairwise disjoint sets.
(a) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$*-data types*, (b) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$*-concepts*, (c) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$*-abstract roles*, and (d) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$*-concrete role terms* are defined according to the DL standard notation (see [13]) as follows:

(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) $C_1, C_2 \longrightarrow A \mid \top \mid \bot \mid \neg C_1 \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \{a\} \mid \exists R.Self \mid \exists R.\{a\} \mid \exists P.\{e_d\}$,

(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) $P_1, P_2 \longrightarrow T \mid \neg P_1 \mid P_1 \sqcup P_2 \mid P_1 \sqcap P_2 \mid P_{C_1|} \mid P_{|t_1} \mid P_{C_1|t_1}$,

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. Notice that data type terms are intended to represent derived data types.

A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*KB* 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.

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

$$\begin{array}{ccccc} R_1 \equiv R_2, & R_1 \sqsubseteq R_2, & R_1 \dots R_n \sqsubseteq R_{n+1}, & \mathsf{Sym}(R_1), & \mathsf{Asym}(R_1), \\ \mathsf{Ref}(R_1), & \mathsf{Irref}(R_1), & \mathsf{Dis}(R_1, R_2), & \mathsf{Tra}(R_1), & \mathsf{Fun}(R_1), \\ R_1 \equiv C_1 \times C_2, & P_1 \equiv P_2, & P_1 \sqsubseteq P_2, & \mathsf{Dis}(P_1, P_2), & \mathsf{Fun}(P_1), \end{array}$$

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 $R_1 \dots R_n \sqsubseteq R$, where $R_1, \dots, R_n, R$ are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract role terms, is called a *role inclusion axiom (RIA)*.

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

- $C_1 \equiv C_2$, $C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \forall R.C_2$, $\exists R.C_1 \sqsubseteq C_2$, $\geq_n R.C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \leq_n R.C_2$,
- $t_1 \equiv t_2$, $t_1 \sqsubseteq t_2$, $C_1 \sqsubseteq \forall P.t_1$, $\exists P.t_1 \sqsubseteq C_1$, $\geq_n P.t_1 \sqsubseteq C_1$, $C_1 \sqsubseteq \leq_n P.t_1$,

where C_1, C_2 are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concept terms, t_1, t_2 data type terms, R a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract role term, and P a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concrete role term. Statements of the form $C \sqsubseteq D$, where C and D are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concept terms, are *general concept inclusion axioms*.

A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*ABox* is a set of *individual assertions* of the forms:

$$a : C_1, \quad (a, b) : R_1, \quad a = b, \quad a \neq b, \quad e_d : t_1, \quad (a, e_d) : P_1,$$

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 via interpretations of the form $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$, where $\Delta^{\mathbf{I}}$ and $\Delta_{\mathbf{D}}$ are nonempty 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 interpretation of concepts and roles, axioms and assertions is defined in [8, Table 2].

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 2]. 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 [8, Table 2]. A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ is consistent if there exists a $\mathbf{D}$-model $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ of $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{R}$.

The HOCQA problem for $\mathcal{DL}_{\mathbf{D}}^{\mathbf{4},\times}$. We recall that the problem of *Higher-Order Conjuctive Query Answering* (HOCQA) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, introduced in [5], is a generalization of the Conjunctive Query Answering problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ defined in [4]. The HOCQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ relies on the notion of *Higher-Order* (HO) $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query, admitting variables of three sorts: individual and data type variables, concept variables, and role variables. The HOCQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ consists in finding the HO answer set of an HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query (see below) with respect to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB.

Specifically, let $\mathsf{V_i} = \{\mathsf{v}_1, \mathsf{v}_2, \ldots\}$, $\mathsf{V_e} = \{\mathsf{e}_1, \mathsf{e}_2, \ldots\}$, $\mathsf{V_d} = \{\mathsf{t}_1, \mathsf{t}_2, \ldots\}$, $\mathsf{V_c} = \{\mathsf{c}_1, \mathsf{c}_2, \ldots\}$, $\mathsf{V_{ar}} = \{\mathsf{r}_1, \mathsf{r}_2, \ldots\}$, and $\mathsf{V_{cr}} = \{\mathsf{p}_1, \mathsf{p}_2, \ldots\}$ be pairwise disjoint denumerably infinite sets of variables disjoint from $\mathbf{Ind}$, $\bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$, $\mathbf{C}$, $\mathbf{R_A}$, and $\mathbf{R_D}$. HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*atomic formulae* are expressions of the following types:

$$R(w_1, w_2), P(w_1, u), C(w_1), t(u), \mathsf{r}(w_1, w_2), \mathsf{p}(w_1, u), \mathsf{c}(w_1), \mathsf{t}(u), w_1 = w_2,$$

where $w_1, w_2 \in \mathsf{V_i} \cup \mathbf{Ind}$, $u \in \mathsf{V_e} \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$, 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, t is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-data type term, $\mathsf{r} \in \mathsf{V_{ar}}$, $\mathsf{p} \in \mathsf{V_{cr}}$, $\mathsf{c} \in \mathsf{V_c}$, $\mathsf{t} \in \mathsf{V_d}$. 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 λ the *empty* HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query.

Let $\mathsf{v}_1, \ldots, \mathsf{v}_n \in \mathsf{V_i}$, $\mathsf{e}_1, \ldots, \mathsf{e}_g \in \mathsf{V_e}$, $\mathsf{t}_1, \ldots, \mathsf{t}_l \in \mathsf{V_d}$, $\mathsf{c}_1, \ldots, \mathsf{c}_m \in \mathsf{V_c}$, $\mathsf{r}_1, \ldots, \mathsf{r}_k \in \mathsf{V_{ar}}$, $\mathsf{p}_1, \ldots, \mathsf{p}_h \in \mathsf{V_{cr}}$, $o_1, \ldots, o_n \in \mathbf{Ind}$, $e_{d_1}, \ldots, e_{d_g} \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$, $C_1, \ldots, C_m \in \mathbf{C}$, $R_1, \ldots, R_k \in \mathbf{R_A}$, and $P_1, \ldots, P_h \in \mathbf{R_D}$. A substitution $\sigma \coloneqq \{\mathsf{v}_1/o_1, \ldots, \mathsf{v}_n/o_n, \mathsf{e}_1/e_{d_1}, \ldots, \mathsf{e}_g/e_{d_g}, \mathsf{t}_1/t_1, \ldots, \mathsf{t}_l/t_l, \mathsf{c}_1/C_1, \ldots, \mathsf{c}_m/C_m,$ $\mathsf{r}_1/R_1, \ldots, \mathsf{r}_k/R_k, \mathsf{p}_1/P_1, \ldots, \mathsf{p}_h/P_h\}$ is a map such that, for every HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-literal L, $L\sigma$ is obtained from L by replacing: (a) the occurrences of v_i in L with o_i, for $i = 1, \ldots, n$; (b) the occurrences of e_b in L with d_b, for $b = 1, \ldots, g$; (c) the occurrences of t_s in L with t_s, for $s = 1, \ldots, l$; (d) the occurrences of c_j in L with C_j, for $j = 1, \ldots, m$; (e) the occurrences of r_ℓ in L with R_ℓ, for $\ell = 1, \ldots, k$; (f) the occurrences of p_t in L with P_t, for $t = 1, \ldots, h$. Substitutions can be extended to HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries in the usual way.

Let $Q \coloneqq (L_1 \wedge \ldots \wedge L_m)$ be a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query, and $\mathcal{KB}$ a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB. A substitution σ involving *exactly* the variables occurring in Q is a *solution for Q w.r.t. $\mathcal{KB}$*, if there exists a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}$ such that $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$ and $\mathbf{I} \models_{\mathbf{D}} Q\sigma$. The collection Σ 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 Σ of Q w.r.t. $\mathcal{KB}$.

The HOCQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ can be instantiated to the most significant ABox reasoning problems for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (see [5]).

Representing $\mathcal{DL}_{\mathbf{D}}^{\mathbf{4},\times}$ in set-theoretic terms. $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries can be represented in set-theoretic terms by exploiting a mapping θ defined in [5]. The function θ translates $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ statements in $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae in CNF. Specifically, θ maps injectively individuals a, constants $e_d \in$

$N_C(d)$, variables $w \in \mathsf{V_i}$, and variables $u \in \mathsf{V_e}$ into sort 0 variables x_a, x_{e_d}, x_w, x_u, the constant concepts $\top$ and $\bot$, data type terms t, concept terms C, $\mathsf{c} \in \mathsf{V_c}$, and $\mathsf{t} \in \mathsf{V_d}$ into sort 1 variables $X^1_\top$, $X^1_\bot$, X^1_t, X^1_C, X^1_c, X^1_t respectively, and the universal relation U, abstract role terms R, concrete role terms P, $\mathsf{r} \in \mathsf{V_{ar}}$, and $\mathsf{p} \in \mathsf{V_{cr}}$ into sort 3 variables X^3_U, X^3_R, X^3_P, X^3_r, X^3_p, respectively.[1]

The mapping θ is defined for HO-$\mathcal{DL}^{4,\times}_{\mathbf{D}}$-atomic formulae as follows: $\theta(R(w_1, w_2)) \coloneqq \langle x_{w_1}, x_{w_2}\rangle \in X^3_R$, $\theta(P(w_1, u)) \coloneqq \langle x_{w_1}, x_u\rangle \in X^3_P$, $\theta(C(w_1)) \coloneqq x_{w_1} \in X^1_C$, $\theta(t(u)) \coloneqq x_u \in X^1_t$, $\theta(w_1 = w_2) \coloneqq x_{w_1} = x_{w_2}$, $\theta(\mathsf{t}(u)) \coloneqq x_u \in X^1_\mathsf{t}$, $\theta(\mathsf{c}(w_1)) \coloneqq x_{w_1} \in X^1_\mathsf{c}$, $\theta(\mathsf{r}(w_1, w_2)) \coloneqq \langle x_{w_1}, x_{w_2}\rangle \in X^3_\mathsf{r}$, $\theta(\mathsf{p}(w_1, u)) \coloneqq \langle x_{w_1}, x_u\rangle \in X^3_\mathsf{p}$.

Finally, θ is extended to HO-$\mathcal{DL}^{4,\times}_{\mathbf{D}}$-conjunctive queries and to substitutions in a standard way.

From now on we denote with $\phi_{\mathcal{KB}}$ the $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$ translation of a $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-KB $\mathcal{KB}$ and with ψ_Q the $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula representing the HO-$\mathcal{DL}^{4,\times}_{\mathbf{D}}$-conjunctive query Q. The formula $\phi_{\mathcal{KB}}$ is a conjunction of $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formulae of type $(\forall z_1)\dots(\forall z_n)\varphi_0$, with φ_0 a clause of $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-literals, since (a) each $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-KB $\mathcal{KB}$ is a set of statements H such that $\theta(H)$ is a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula in Table 1; and (b) $\phi_{\mathcal{KB}}$ is constructed by conjoining the $\theta(H)$s, moving universal quantifiers as inward as possible, and renaming quantified variables as to be pairwise distinct. The interested reader is referred to [6] for full details.

Finally, the HOCQA problem for $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formulae can be stated as follows. Let ψ be a conjunction of $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-literals and ϕ a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula. The *HOCQA problem for ψ w.r.t. ϕ* consists in computing the HO *answer set of ψ w.r.t. ϕ*, namely the collection Σ' of all the substitutions σ' such that $\boldsymbol{\mathcal{M}} \models \phi \wedge \psi\sigma'$, for some $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-interpretation $\boldsymbol{\mathcal{M}}$.

3 Overview of the reasoner

We present a general overview of the reasoner and the main notions and definitions concerning the procedures upon which it is based.

The input of the reasoner is an OWL ontology serialized in the OWL/XML syntax and admitting SWRL rules (see Figure 1).

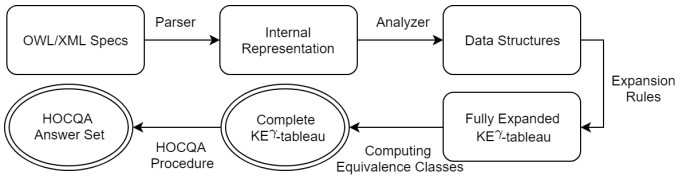

Fig. 1. Execution cycle of the reasoner.

[1] The use of level 3 variables to model abstract and concrete role terms is motivated by the fact that their elements, that is ordered pairs $\langle x, y\rangle$, are encoded in Kuratowski's style as $\{\{x\}, \{x, y\}\}$, namely as collections of sets of objects.

If the ontology meets the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ requirements, a parser produces the internal coding of all axioms and assertions of the ontology in set-theoretic terms, as a list of strings. Then the system builds the data-structures required to execute the algorithm. In the two subsequent steps, the reasoner constructs a complete KE$^\gamma$-tableau $\mathcal{T}_{\mathcal{KB}}$ whose open branches represent all possible models for the input KB $\phi_{\mathcal{KB}}$ (see below for the definition of KE$^\gamma$-tableau). The tableau $\mathcal{T}_{\mathcal{KB}}$ is constructed (1) by systematically applying the following two rules: (1a) a generalization of the KE-elimination rule incorporating the γ-rule, and (1b) the principle of bivalence rule (PB-rule) (thus constructing all branches of the KE$^\gamma$-tableau—see Figure 2), and then (2) processing each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ by constructing the equivalence classes of the individuals involved in formulae of type $x = y$ occurring in ϑ and substituting each individual x on ϑ with the representative of the equivalence class of x. Such step returns the complete KE$^\gamma$-tableau. Finally, the reasoner takes as input the internal coding of ψ_Q, i.e. the set-theoretic representation of a query Q, and computes the HO-answer set of ψ_Q with respect to $\phi_{\mathcal{KB}}$. The task of computing the complete KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ is performed by procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ illustrated in [8, Figure 8], whereas the task of computing the HOCQA answer set of a given query w.r.t the KB is performed by procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ shown in [8, Figure 9].

Let $\Phi_{\mathcal{KB}} \coloneqq \{\phi : \phi \text{ is a conjunct of } \phi_{\mathcal{KB}}\}$. The procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ constructs a complete KE$^\gamma$-tableau $\mathcal{T}_{\mathcal{KB}}$ for the set $\Phi_{\mathcal{KB}}$ of the conjuncts of $\phi_{\mathcal{KB}}$ representing the saturation of the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB.

At this point, it is convenient to give the definition of KE$^\gamma$-tableaux. Let $\Phi \coloneqq \{C_1, \ldots, C_p\}$, where each C_i is either a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literal of the types in Table 1 or a $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formula of the form $(\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$, with $\beta_1 \ldots \beta_n$ $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals. $\mathcal{T}$ is a KE$^\gamma$*-tableau* for Φ if there exists a finite sequence $\mathcal{T}_1, \ldots, \mathcal{T}_t$ such that (i) $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $C_1, \ldots, C_p$, (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 (E$^\gamma$-rule or PB-rule) in Figure 2 or by applying a substitution σ to a branch ϑ of $\mathcal{T}_i$ (in particular, the substitution σ is applied to each formula X of ϑ and the resulting branch will be denoted with $\vartheta\sigma$). In the definition of the E$^\gamma$-rule reported in Figure 2, (a) $\tau \coloneqq \{x_1/x_{o_1} \ldots x_m/x_{o_m}\}$ is a substitution such that $x_1, \ldots, x_m$ are the quantified variables in ψ and $x_{o_1}, \ldots, x_{o_m} \in \mathtt{Var}_0(\phi_{\mathcal{KB}})$; and (b) $\mathcal{S}^{\overline{\beta}_i\tau} \coloneqq \{\overline{\beta}_1\tau, \ldots, \overline{\beta}_n\tau\} \setminus \{\overline{\beta}_i\tau\}$ is a set containing the complements of all the disjuncts $\beta_1 \ldots \beta_n$ to which the substitution τ is applied, with the exception of the disjunct β_i.

Initially, the procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ constructs a one-branch KE$^\gamma$-tableau $\mathcal{T}_{\mathcal{KB}}$ for the set $\Phi_{\mathcal{KB}}$ of conjuncts of $\phi_{\mathcal{KB}}$. Then, it expands $\mathcal{T}_{\mathcal{KB}}$ by systematically applying the E$^\gamma$-rule and the PB-rule in Figure 2 to formulae of type $\psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ till they are all fulfilled, giving priority to the E$^\gamma$-rule. Once such rules are no longer applicable, for each open branch ϑ of the resulting KE$^\gamma$-tableau, atomic formulae of type $x = y$ occurring in ϑ are used to compute the equivalence class of x and y. For each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the equivalence class of each variable occurring in ϑ is obtained by computing

the substitution σ_ϑ such that $\vartheta\sigma_\vartheta$ does not contain literals of type $x = y$, for distinct x, y. The resulting pair $(\vartheta, \sigma_\vartheta)$ is added to the set $\mathcal{E}$.

$$\frac{\psi \qquad \mathcal{S}^{\overline{\beta}_i\tau}}{\beta_i\tau}\ \mathbf{E}^\gamma\textbf{-rule}$$

where
$\psi \coloneqq (\forall x_1)\ldots(\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$,
$\tau \coloneqq \{x_1/x_{o_1}\ldots x_m/x_{o_m}\}$,
and $\mathcal{S}^{\overline{\beta}_i\tau} \coloneqq \{\overline{\beta}_1\tau, ..., \overline{\beta}_n\tau\} \setminus \{\overline{\beta}_i\tau\}$,
for $i = 1, ..., n$

$$\frac{}{A \mid \overline{A}}\ \textbf{PB-rule}$$

where A is a literal

Fig. 2. Expansion rules for the KE$^\gamma$-tableau.

The procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes as input a query ψ_Q and the set $\mathcal{E}$ yielded by the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and returns the answer set Σ' of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. For each open and complete branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ builds a decision tree $\mathcal{D}_\vartheta$ where each maximal branch induces a substitution σ' such that $\sigma_\vartheta\sigma'$ belongs to the answer set of ψ_Q w.r.t. to $\phi_{\mathcal{KB}}$.

$\mathcal{D}_\vartheta$ is obtained by constructing a stack of its nodes. Initially the stack contains just the root node $(\epsilon, \psi_Q\sigma_\vartheta)$ of $\mathcal{D}_\vartheta$, with ϵ the empty substitution. At each step, the procedure pops out from the stack an element $(\sigma', \psi_Q\sigma_\vartheta\sigma')$ and iteratively selects a literal q from the query $\psi_Q\sigma_\vartheta\sigma'$ and eliminates it from $\psi_Q\sigma_\vartheta\sigma'$. Then, the set of literals t in ϑ matching q is computed by putting $Lit_q^\vartheta \coloneqq \{t \in \vartheta : t = q\rho$, for some substitution $\rho\}$. The successors of the current node are computed by pushing the node $(\sigma'\rho, \psi_Q\sigma_\vartheta\sigma'\rho)$ in the stack, for each element in Lit_q^ϑ . If the current node has the form of (σ', λ), with λ the empty query, the last literal of ψ_Q has been treated and the substitution $\sigma_\vartheta\sigma'$ is inserted in Σ'. Notice that, in case of a failing query match, the set Lit_q^ϑ is empty and then no successor node is pushed into the stack. Thus, the failing branch of $\mathcal{D}_\vartheta$ is abandoned and another branch is selected by popping one of the nodes of $\mathcal{D}_\vartheta$ from the stack.

Computational complexity results can be found in [9].

3.1 Some implementation details

We first show how the internal coding of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs is represented in terms of $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae and the data-structures used by the reasoner for representing formulae, nodes, and how KE$^\gamma$-tableaux are implemented. Then we describe the most relevant functions that implement the procedures *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and also illustrate an example of reasoning in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

To begin with, $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-variables, quantifiers, Boolean operators, set-theoretic relators, and pairs are mapped into strings as follows. Variables of type X^i_{name} are mapped into strings of the form $Vi\{name\}$. For the sake of uniformity, variables of sort 0 are denoted with $X^0, Y^0, \ldots$, whereas individuals a, concepts C, and roles R of a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB are respectively mapped into the variables X^0_a, X^1_C,

and X^3_R, according to the function θ described in [5]. The symbols $\forall$, $\wedge$, $\vee$, $\neg\wedge$, $\neg\vee$ are mapped into the strings `$FA`, `$AD`, `$OR`, `$DA`, `$RO`, respectively. The relators $\in$, $\notin$, $=$, $\neq$ are mapped into the strings `$IN`, `$NI`, `$EQ`, `$QE`, respectively. A pair $\langle X^0_1, X^0_2\rangle$ is mapped into the string `$OA V01 $CO V02 $AO`, where `$OA` represents the bracket "$\langle$", `$AO` the bracket "$\rangle$", and `$CO` the comma symbol.

Then, data-structures for representing the KB are built. $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-variables are implemented by means of the class `Var` that has four fields. The field `type` of type integer indicates the sort of the variable, the field `name` of type string represents the name of the variable, and the field `var` of type integer represents a free variable if set to 0, and a quantified variable if set to 1. The field `index` stores the position of the variable in the vector `VVL`, delegated to collect free variables. Quantified and free variables are collected in the vectors `VQL` and `VVL` respectively, which provide a subvector for each sort of variable.

The operators admitted in $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$, internally coded as strings, are mapped into three vectors that are fields of the class `Operator`. Specifically, the vector `boolOp` contains the values `$OR`, `$AD`, `$RO`, `$DA`, the vector `setOp` the values `$IN`, `$EQ`, `$NI`, `$QE`, `$OA`, `$AO`, `$CO`, and the vector `qutOp` the value `$FA`.

$\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-literals are stored using the class `Lit` that has two fields. The field `litOp` of type integer represents the operator of the formula and corresponds to the index of one of the first four elements of the vector `setOp`. The field `components` is a vector whose elements point to the variables involved in the literal and stored in `VQL` and `VVL`.

$\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formulae are represented by the class `Formula`, having a binary tree structure, whose nodes contain objects of the class `Lit`. The left and right children contain the left and right subformula, respectively. The class `Formula` contains the following fields: the field `lit` of type pointer to `Lit` represents the literal; the field `operand` represents the propositional operator, and its value is the index of the corresponding element of the vector `boolOp`; the field `psubformula` of type pointer to `Formula` is the pointer to the father node, whereas the fields `lsubformula` and `rsubformula` contain the pointers to the nodes representing the left and the right component of the formula, respectively.

The procedure *Consistency-*$\mathcal{DL}^{4,\times}_{\mathbf{D}}$ is based on the data-structure implemented by the class `Tableau`. This class uses the instances of the class `Node` that represents the nodes of the KE$^\gamma$-tableau. The class `Node` has a tree-shaped structure and four fields: the field `setFormula`, of type vector of `Formula`, that collects the formulae of the current node, and three pointers to instances of the class `Node`. These are the `leftchild`, `rightchild`, and `father` fields, which point to the left child node, right child node, and father node, respectively.

The root node of the class `Tableau` contains the field `root` of type pointer to `Node`. The fields `openbranches` and `closedbranches` collect the set of open branches and of closed branches, respectively. In addition, the class `Tableau` is provided with the field `EqSet` that is a three-dimensional vector of integers storing the equivalence classes induced by atomic formulae of type $X^0 = Y^0$. In particular, `EqSet` stores a vector containing the indices of `VVL` corresponding to the variables belonging to the equivalence classes.

As mentioned above, the reasoner takes as input an OWL ontology compatible with the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ requirements, also admitting SWRL rules, and serialized in the OWL/XML syntax. As first step, the function `readOWLXMLontology` produces the internal coding of all axioms and assertions of the ontology, yielding a list of strings. Then the reasoner builds from the output of `readOWLXMLontology` the objects of type `Formula` that implement the $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae representing the KB, and stores them in the field `root` of an object of type `Tableau`. In this phase, formulae are transformed in CNF and universal quantifiers are moved as inward as possible and renamed in such a way as to be pairwise distinct. The object of type `Tableau` representing the KE$^\gamma$-tableau is the input to the procedure `expandGammaTableau` that expands the KE$^\gamma$-tableau by iteratively selecting and fulfilling purely universal quantified input formulae. Once a purely universal quantified formula has been selected, `expandGammaTableau` builds iteratively the set of substitutions τ to be applied to the selected formula. A substitution τ is a map from the indices of the quantified variables of the formula, selected in order of appearance, to the elements of the vector `VVL`. The implementation of τ applies standard techniques for computing the *variations with repetition* of the set of indices of the elements of `VVL` taken to k by k, where k is the number of quantified variables occurring in the selected formula.

The procedure `expandGammaTableau` fulfills the formula selected by systematically applying the functions `EGrule` with the current τ and `PBrule`, respectively implementing the E$^\gamma$-rule and the PB-rule. More precisely, it works as follows. The disjuncts of the current formula to which τ is applied are stored in a temporary vector and selected iteratively. If a disjunct has its negation on the branch, it is removed from the temporary vector. Once all the elements of the temporary vector have been selected, if the last one does not have its negation on the branch, then `EGrule` is applied to the formula and the last element of the temporary vector is inserted in the branch according to Figure 2. If there is more than one element left in the temporary vector, then the procedure `PBrule` is applied. In case the stack is empty, a contradiction is found and the branch gets closed and inserted in the vector `closedbranches`.

If the procedure `expandGammaTableau` terminates with some elements in `openbranches`, then the reasoner builds the set of equivalence classes of the variables involved in formulae of type $X^0 = Y^0$, for each element of `openbranches` by means of the procedure `buildsEqSet`. The latter procedure updates the field `EqSet` of the object of type `Tableau` with the new information concerning the set of equivalence classes. After the execution of `buildsEqSet`, if `openbranches` contains some elements, a consistent KB is returned.

Procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is implemented by the function `performQuery` that takes as input the object of type `Tableau` returned by `buildsEqSet` and a string representing the internal coding of the input query ψ_Q, and returns an object of type `QueryManager` storing, among other information, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. The function `performQuery` uses an object of type `QueryManager` that stores the input query ψ_Q as a string, an object of type `Formula` representing ψ_Q, and the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$, for each element of `openbranches`. The

answer set is implemented by endowing the object of type `QueryManager` with the pair of vectors `VarMatch`. The first vector of `VarMatch` contains an integer for each element in `openbranches`: this is set to 1 if the corresponding branch has solution, 0 otherwise. The second (three-dimensional) vector contains for each element in `openbranches` a vector of solutions, each one constituted by a vector of pairs of pointers to `Var`. The first `Var` of such pair is a variable belonging to the query, whereas the second `Var` is the matched individual.

For each element in `openbranches`, the function `performQuery` implements a decision tree by means of a stack that keeps track of the partial solutions of the query, as nodes of the decision tree. Such a stack, called `matchSet`, is constituted by a vector of pairs of objects of type `Var` such that the first one represents the query variable and the second one the matched element. Initially, `matchSet` is empty. At first step, the procedure selects the first conjunct of the query and, for each match found, it pushes in `matchSet` a vector of pairs representing the match. The procedure selects iteratively the conjuncts of the query and then applies to the selected conjunct the substitution that is currently at the top of `matchSet`. If the literal obtained by the application of such partial solution has one or more matches in the branch, the resulting substitutions are pushed in `matchSet`. Once all the literals of the query have been processed, if `matchSet` is not empty, it contains the leaves of the maximal branches of the decision tree, which are all added to `VarMatch`.

3.2 Example of reasoning in $\mathcal{DL}_{\mathbf{D}}^{\mathbf{4,\times}}$

Let us consider the ontology displayed in Figure 3.

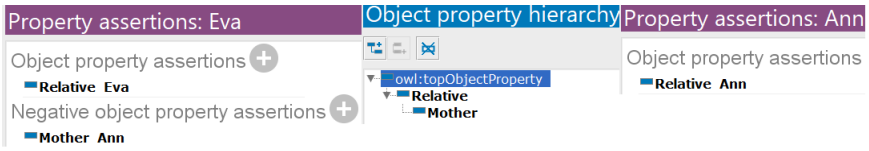

Fig. 3. OWL ontology of the example.

The KB in terms of $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$ is the following formula:

$$\begin{aligned}\phi_{\mathcal{KB}} \coloneqq\ & \neg(\langle x_{Eva}, x_{Ann}\rangle \in X^3_{Mother}) \wedge \\ & \langle x_{Ann}, x_{Ann}\rangle \in X^3_{Relative} \wedge \langle x_{Eva}, x_{Eva}\rangle \in X^3_{Relative} \wedge \\ & (\forall z_1)(\forall z_2)(\neg(\langle z_1, z_2\rangle \in X^3_{Mother}) \vee \langle z_1, z_2\rangle \in X^3_{Relative})\end{aligned}$$

Let $\psi_Q = \langle z, x_{Eva}\rangle \in X^3_{Mother}$ be a query represented in set-theoretic terms. A complete KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ and the decision trees constructed for the evaluation of ψ_Q on each open branch of the KE$^\gamma$-tableau are shown in Figure

4. Notice that, the decision tree constructed on the leftmost open branch of the KE$^\gamma$-tableau provides no solution.

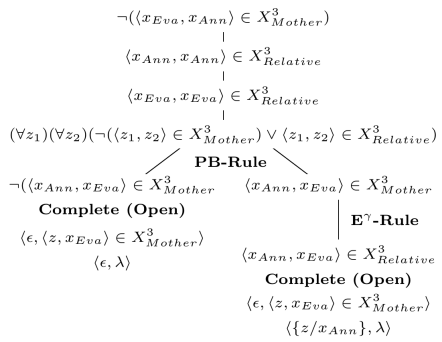

Fig. 4. KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ and decision trees for the evaluation of ψ_Q.

The internal representation of the OWL ontology is shown in Figure 5. The KE$^\gamma$-tableau computed by the reasoner and the evaluation of the query ψ_Q are reported in Figure 6. Finally, Figure 7 shows a performance comparison between our implementation of the KE-tableau presented in [7] and the KE$^\gamma$-tableau system for $\mathcal{DL}^{4,\times}_{\mathbf{D}}$ presented in this paper. The metric used in the benchmarking is the number of models of the input KB computed by the reasoners and the time required to compute such models. As shown in Figure 7, the KE$^\gamma$-tableau has a better performance than the KE-tableau up to about 400%, even if in some cases the performances of the two systems are comparable, as shown in the plot. We conclude that the KE$^\gamma$-tableau system is always convenient, also because the expansions of quantified formulae are not stored in memory.

```
($OA V0{Ann} $CO V0{Ann}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Eva}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Ann}$AO $NI V3{Mother})
($FA V0{z})($FA V0{z1})( ($OA V0{z} $CO V0{z1}$AO $NI V3{Mother})
  $OR ($OA V0{z} $CO V0{z1}$AO $IN V3{Relative}) )
```

Fig. 5. Internal representation of the ontology.

```
Branch: 0
($OA V0{Ann} $CO V0{Eva}$AO $NI V3{Mother})
----
($OA V0{Ann} $CO V0{Ann}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Eva}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Ann}$AO $NI V3{Mother})
($FA V0{z})($FA V0{z1})( ($OA V0{z} $CO V0{z1}$AO $NI V3{Mother})
  $OR ($OA V0{z} $CO V0{z1}$AO $IN V3{Relative}) )
----
Branch: 1
($OA V0{Ann} $CO V0{Eva}$AO $IN V3{Mother})
($OA V0{Ann} $CO V0{Eva}$AO $IN V3{Relative})
----
($OA V0{Ann} $CO V0{Ann}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Eva}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Ann}$AO $NI V3{Mother})
($FA V0{z})($FA V0{z1})( ($OA V0{z} $CO V0{z1}$AO $NI V3{Mother})
  $OR ($OA V0{z} $CO V0{z1}$AO $IN V3{Relative}) )
----
```

```
Tableau branch number: 1
Solution number: 0
V0{z},V0{Ann};
Printing Y/N results ...
Branch number: 0 Answer:0
Branch number: 1 Answer:1
```

Fig. 6. The KE$^\gamma$-tableau and the evaluation of ψ_Q computed by the reasoner.

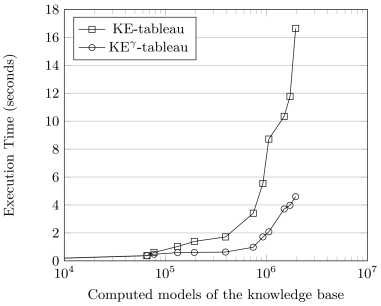

Fig. 7. Comparison between KE-tableau and KE$^\gamma$-tableau systems.

4 Conclusions

We presented a C++ implementation of a KE$^\gamma$-tableau system for the most widespread reasoning tasks of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, such as consistency checking of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and a generalization of the CQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, , called HOCQA problem, admitting conjunctive queries with variables of three sorts. These problems have been addressed by translating $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and higher-order $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries in terms of formulae of the set-theoretic fragment $\mathsf{4LQS}^{\mathsf{R}}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$. The reasoner is an improvement of the KE-tableau system introduced in [7] to check consistency of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs, as it admits a generalization of the KE-elimination rule incorporating the γ-rule. The reasoner takes as input OWL ontologies compatible with the specifications of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ serialized in the OWL/XML format and admitting SWRL rules.

Finally, we showed that the reasoner presented in this paper is more efficient than the one introduced in [7], by means of suitable benchmark test sets.

We plan to extend the set-theoretic fragment underpinning the reasoner to include also a restricted version of the operator of relational composition. This will allow ones to reason with description logics that admit full existential and universal quantification. In addition, we intend to improve our reasoner so as to deal with the reasoning problem of ontology classification. Then, we shall compare the resulting reasoner with existing well-known reasoners such as HermiT [12] and Pellet [18], providing some benchmarking. We also plan to allow data type reasoning by either integrating existing solvers for the Satisfiability Modulo Theories (SMT) problem or by designing ad-hoc new solvers. Finally, as each branch of the KE$^\gamma$-tableau can be computed by a single processing unit, we plan to implement a parallel version of the software by using the Nvidia CUDA library.

References

1. C. Cantale, D. Cantone, M. Nicolosi-Asmundo, and D.F. Santamaria. Distant Reading Through Ontologies: The Case Study of Catania's Benedictines Monastery. *JIS.it, 8,3 (September 2017).*

2. D. Cantone, A. Ferro, and E. G. Omodeo. *Computable set theory.* Number 6 in International Series of Monographs on Computer Science, Oxford Science Publications. Clarendon Press, Oxford, UK, 1989.
3. D. Cantone and M. Nicolosi-Asmundo. On the satisfiability problem for a 4-level quantified syllogistic and some applications to modal logic. *Fundamenta Informaticae*, 124(4):427–448, 2013.
4. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. Conjunctive query answering via a fragment of set theory. In *Proc. of ICTCS 2016, Lecce, September 7-9, CEUR-WS Vol. 1720, pp. 23–35.*
5. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. A set-theoretic approach to ABox reasoning services. In *Costantini S., Franconi E., Van Woensel W., Kontchakov R., Sadri F., Roman D. Rules and Reasoning. RuleML+RR 2017.*, Lecture Notes in Computer Science, vol 10364. Springer, 2017.
6. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. A set-theoretic approach to ABox reasoning services. *CoRR*, 1702.03096, 2017. Extended version.
7. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. A C++ reasoner for the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. In *Proc.of CILC 2017, 26-29 September 2017, Naples, Italy. CEUR WS, ISSN 1613-0073, Vol. 1949, pp. 276-280.*, 2017.
8. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. A set-based reasoner for the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. *CoRR*, 1805.08606, 2018. Extended version.
9. D. Cantone, M. Nicolosi-Asmundo, and D. F. Santamaria. An optimized KE-tableau-based system for reasoning in the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. *CoRR*, 1804.11222, 2018. Extended version.
10. D. Cantone, M. Nicolosi-Asmundo, D. F. Santamaria, and F. Trapani. Onto-ceramic: an OWL ontology for ceramics classification. In *Proc. of CILC 2015, CEUR-WS, vol. 1459, pp. 122–127*, Genova, July 1-3, 2015.
11. D. Cantone, E. Omodeo, and A. Policriti. *Set theory for computing: from decision procedures to declarative programming with sets.* Monographs in Computer Science. Springer-Verlag, New York, NY, USA, 2001.
12. B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang. HermiT: An OWL 2 Reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.
13. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proc. 10th Int. Conf. on Princ. of Knowledge Representation and Reasoning, (Doherty, P. and Mylopoulos, J. and Welty, C. A., eds.)*, pages 57–67. AAAI Press, 2006.
14. M. Mondadori M. D'Agostino. The taming of the cut. Classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994.
15. B. Motik and I. Horrocks. OWL datatypes: Design and implementation. In *Proc. of the 7th Int. Semantic Web Conference (ISWC 2008)*, volume 5318 of *LNCS*, pages 307–322. Springer, October 26–30 2008.
16. D. F. Santamaria. *A Set-Theoretical Representation for OWL 2 Profiles.* LAP Lambert Academic Publishing, ISBN 978-3-659-68797-6, 2015.
17. Jacob T. Schwartz, Domenico Cantone, and Eugenio G. Omodeo. *Computational Logic and Set Theory: Applying Formalized Logic to Analysis.* Texts in Computer Science. Springer-Verlag New York, Inc., 2011.
18. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2):51–53, 2007.
19. R. M. Smullyan. *First-order Logic.* Dover books on advanced Math. Dover, 1995.