Querying OWL 2 QL ontologies under the SPARQL Metamodeling Semantics Entailment Regime

Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi

Sapienza Università di Roma
lastname@dis.uniroma1.it

Abstract. OWL 2 QL is the profile of OWL 2 targeted to Ontology-Based Data Access (OBDA) scenarios, where large amount of data are to be accessed, and thus answering conjunctive queries over data is the main task. However, this task is quite restrained wrt the classical KR *Ask-and-Tell* framework based on querying the whole theory, not only facts (data). If we use SPARQL as query language, we get much closer to this ideal. Indeed, SPARQL queries over OWL 2 QL, under the so-called *Direct Semantics Entailment Regime*, may comprise any assertion expressible in the language, i.e., both ABox atoms and TBox atoms, including inequalities expressed by means of DifferentIndividuals. Nevertheless this regime is hampered by the assumption that variables in queries need to be *typed*, meaning that the same variable cannot occur in positions of different types, e.g., both in class and individual position (*punning*). In this paper we dismiss this limiting assumption by resorting to a recent meta modeling semantics and show that query answering in the resulting entailment regime is polynomially compilable into Datalog (and hence PTIME wrt both TBox and ABox).

Introduction

DL-Lite is a family of Description Logics specifically designed for achieving tractability in answering conjunctive queries (CQs) expressed over ontologies. In particular, when the ontology is expressed in one of the logics of the family, i.e., *DL-Lite*$_{\mathtt{R}}$, query answering can be reduced to standard evaluation of first-order queries over a database, and can therefore make use of SQL engines. By virtue of these characteristics, *DL-Lite*$_{\mathtt{R}}$ is at basis of OWL 2 QL, the OWL 2 profile especially designed for Ontology-Based Data Access (OBDA) applications, where the aim is to use an ontology to access a typically big amount of data residing in external data sources.

SPARQL is the de-facto standard language for expressing queries over OWL 2 ontologies. The key form of SPARQL query is the so-called basic graph pattern, that is a conjunction of atoms, where each atom has the form of an axiom expressible in the ontology. So, any axiom that can appear in the ontology can also appear in the conjunctive query pattern corresponding to the SPARQL query. This is coherent with the classical knowledge representation *Ask-and-Tell* framework [10], which is based on "*ask* anything that can be *told* to a knowledge base". Notice that when the Ask-and-Tell framework is specialized to OBDA, it reflects the idea of *querying whole OWL 2 QL theories*, specifying patterns spanning both through the TBox (the intensional knowledge represented in the ontology), and the ABox (the *facts* at the extensional level of

the ontology), with no limitation on the use of variables, and therefore with a distinct metamodeling and metaquerying flavor, see, e.g., [9].

Starting from the seminal work [2], there has been a huge amount of work aiming at designing optimized algorithms for query answering in OBDA, and developing systems implementing such algorithms. So, it is natural to ask whether, after such body of work, the problem of querying OWL 2 QL theories is solved. Surprisingly, the answer to this question is negative, for the following reasons.

First, the queries studied in the great majority of the works on OBDA contain only ABox atoms. In other words, the CQs expressible in the current OBDA systems are able essentially to specify patterns in the data, and retrieve individual objects satisfying such patterns. Obviously, the intensional knowledge represented in the ontology (TBox axioms) is taken into account when answering the query, but in most of the results, the assumption is that TBox atoms do not appear in queries.

Second, the syntax and the semantics of SPARQL conjunctive queries over OWL 2 QL ontologies are defined by means of the so-called *Direct Semantics Entailment Regime (DSER)* [3], which interprets the ontology under the *Direct Semantics (DS)*, i.e., as a first-order theory, and defines the solutions to a query as the set of tuples of IRIs occurring in the ontology that, once substituted to the variables within the query, make the resulting set of axioms *logically implied* by the ontology. The fact that DS interprets the ontology as a first order theory has an important implication: although the syntactic rules for defining an ontology allow *punning*, that is the capability of using the same name in positions of different *type* (i.e., in an object and in a predicate position, or in a class and in a property position), the semantics imposes that occurrences of the same name in different positions are treated as is they were occurrences of different names. A direct consequence of this choice is that the use of variables in the queries is limited by the so-called *typing constraint*, by which no variable can appear both in object position (i.e., as an argument of a class or of a property), and in predicate position (i.e., as class or property). The result is that we cannot join variables denoting classes with those denoting individuals, thus preventing the specification of interesting queries related to metamodeling, e.g., the one asking for all classes that are instances of another class [9].

Third, although OWL 2 QL allows specifying inequalities between IRIs, by means of axioms of the form DifferentIndividuals(e_1 e_2), imposing that e_1 and e_2 denote distinct objects of the domain, the inequality predicate is completely dismissed in the work on query answering over OWL 2 QL ontologies. Notably, it is a folk theorem that extending the approach to cover the inequality predicate is completely trivial, but we argue in this paper that this is not the case in general, and in particular when querying whole theories. Indeed, inequalities between ontology entities can be logically implied by an OWL 2 QL ontology, as shown in the following simple example. Consider the ontology consisting of the axioms:

```
ClassAssertion(:Male :p).
ClassAssertion(:Male :peter).
ClassAssertion(:Female :petra).
SubClassOf(:Female :Person).
SubClassOf(:Male :Person).
DisjointClasses(:Female :Male)
```

and suppose we want to retrieve all classes that contain at least two distinct instances, by means of the SPARQL query:

```
select $x where {ClassAssertion($x $y).ClassAssertion($x $z).
                 DifferentIndividuals($y $z)}
```

It is easy to see that, since :Male and :Female are disjoint, :petra and :peter denote distinct domain objects in every model. Hence, the answer to the query is {:Person, owl:Thing}. This clearly shows that the presence of inequalities within queries requires forms of reasoning taking into account the whole ontology.

The goal of this paper is to present the first approach to querying OWL 2 QL theories, which unleashes the potentiality of the metamodeling characteristics inherent in OWL 2 QL, and of the metaquerying capabilities of SPARQL. To this aim we introduce a new entailment regime, called *Metamodeling Semantics Entailment Regime (MSER)*, which generalizes DSER by (*i*) adopting the Metamodeling Semantics (MS) for OWL 2 QL, introduced in [9], and (*ii*) relaxing the typing constraint of DSER, thus allowing the same variable to occur in positions of different type, e.g., in object and class position.

The main contribution of this paper is to show that both checking the consistency of an OWL 2 QL ontology and answering SPARQL queries over an OWL 2 QL ontology under MSER are polynomially compilable into Datalog (and hence PTIME wrt both TBox and ABox). It is easy to verify that querying OWL 2 QL ontologies with SPARQL under DSER (but *without inequality axioms*) can be polynomially reduced to evaluating a Datalog program. Our result shows that the same is true for the more general problem of querying OWL 2 QL theories under MSER, with inequalities. Moreover, our Datalog reduction provides the first algorithm for querying OWL 2 QL theories with SPARQL basic graph patterns without any restriction.

Related work. Related to our work are results on query answering in *DL-Lite*$_{\mathrm{R}}$ [2, 1, 6], i.e., the logic underpinning OWL 2 QL, and, in particular, results of [5] showing that answering conjunctive queries with inequalities in *DL-Lite*$_{\mathrm{R}}$ is in general undecidable. We point out, however, that such a negative result is due to the fact that existential variables and union are assigned the standard logical meaning, which is not the case in SPARQL. Few recent works [7, 4, 9, 8] have investigated the problem of answering SPARQL queries over OWL 2 QL theories. However, none of such works considers queries possibly containing inequalities. Moreover, all such works interpret ontologies according to DS and thus do not consider queries that violate the typing constraint, with the exception of [9, 8], which consider the problem of querying OWL 2 QL theories comprising metaclasses and metaproperties, as allowed by the OWL 2 standard, and, hence, to overcome the limitations of DS, introduce and use MS[1]. In particular, their study show that the problem of answering untyped queries is in general intractable (in data complexity), even in the absence of inequalities. However, they considered a variant of MSER, that assigns to existential variables and union the classical logical meaning.

[1] In fact, in [9], the authors use the name *Higher Order Semantics (HOS)* to denote what we call here Metamodeling Semantics. We prefer the latter because, even though the proposed semantic structure has a second-order flavor, its expressive power does not exceed first-order.

Preliminaries

OWL 2 QL. Ontology *entities*, such as individuals, classes, object properties, etc., are denoted by *expressions*. *Atomic expressions* correspond to element in the ontology vocabulary. The *vocabulary* $V_{\mathcal{O}}$ of an ontology $\mathcal{O}$ is defined as the tuple (V_N, V_C, V_{OP}, V_{DP}, V_{DT}, $\mathtt{L}_{\mathtt{QL}}$), where (*i*) V_N is the set of IRIs occurring in $\mathcal{O}$ extended with the OWL 2 QL reserved vocabulary, (*ii*) V_C (resp., V_{OP}, V_{DP}), is the subset of V_N consisting of the IRIs that appear in class (resp., object property, data property) positions in $\mathcal{O}$, or are reserved IRIs (*Internationalized Resource Identifiers*) denoting classes (resp., object properties, data properties) (*iii*) V_{DT} is a subset of the datatypes in OWL 2 QL, and (*iv*) $\mathtt{L}_{\mathtt{QL}}$ is the set of literals occurring in some logical axiom of $\mathcal{O}$. *Complex expressions* are built on the basis of $V_{\mathcal{O}}$. We denote by $Exp^{\mathcal{O}}$ the *finite* set of *expressions* that can be built on $V_{\mathcal{O}}$. Thus, for example, if $e \in V_{OP}$, then ObjectInverseOf(e) and ObjectSomeValuesFrom(e_1 e_2) are complex expressions in $Exp^{\mathcal{O}}$. For the sake of simplicity, in the following, we use the Description Logic (DL) syntax for denoting OWL 2 QL expressions. For example, we respectively denote by e^- and $\exists e_1.e_2$ the two above mentioned expressions. Also, we denote by $\top_C, \top_R, \top_D, \bot_C, \bot_R, \bot_D$, respectively, the OWL 2 QL reserved IRIs: owl:Thing, owl:topObjectProperty,owl:topDataProperty,owl:Nothing,owl:bottomObjectProperty,owl:bottomDataProperty.

An OWL 2 QL theory, or OWL 2 QL ontology (or simply *ontology* in the following) is a finite set of OWL 2 QL *(logical) axioms*. As we do for expressions, we write axioms using the DL syntax, which we modify as described next. Inclusions axioms are written as $e_1 \sqsubseteq_* e_2$, where the subscript * is C, R, A, or D, depending on whether the inclusion is an inclusion between classes, object properties, data properties, or datatypes, respectively. Similarly, disjointness axioms are written as $e_1 \sqsubseteq_* \neg e_2$. Also, axioms asserting that an object property e is reflexive (irreflexive), are written ref(e), (resp., irr(e)). We classify logical axioms into (*i*) *positive TBox axioms*, i.e., inclusion and reflexivity axioms, (*ii*) *negative TBox axioms*, i.e., disjointness and irreflexivity axioms, and (*iii*) *ABox axioms*, i.e., axioms of the forms $C(a)$, $R(a, b)$, $A(a, b)$, $a \neq b$. Axioms not in the above list can be expressed by appropriate combinations of the ones listed.

OWL 2 Metamodeling Semantics The OWL 2 Metamodeling Semantics (MS) was introduced in [9] and is based on the notion of interpretation structure, which plays the same role as the "interpretation domain" in classical first-order logic. Specifically, an *interpretation structure* is a tuple $\Sigma = \langle \Delta_o, \Delta_v, \cdot^I, \cdot^E, \cdot^R, \cdot^A, \cdot^T \rangle$ where:

- Δ_o, the *object domain*, and Δ_v, the *value domain* are two disjoint nonempty sets, forming the interpretation domain $\Delta = \Delta_o \cup \Delta_v$;
- $\cdot^E : \Delta_o \rightarrow \mathcal{P}(\Delta_o)$ is a partial function;
- $\cdot^R : \Delta_o \rightarrow \mathcal{P}(\Delta_o \times \Delta_o)$ is a partial function;
- $\cdot^A : \Delta_o \rightarrow \mathcal{P}(\Delta_o \times \Delta_v)$ is a partial function;
- $\cdot^T : \Delta_o \rightarrow \mathcal{P}(\Delta_v)$ is a partial function;
- $\cdot^I : \Delta_o \rightarrow \{\mathtt{T}, \mathtt{F}\}$ is a total function s.t. for each $d \in \Delta_o$, if $\cdot^E, \cdot^R, \cdot^A, \cdot^T$ are undefined for d, then $d^I = \mathtt{T}$.

Thus, the interpretation structure is not simply a set, but a mathematical representation of a world made up by elements which have complex inter-relationships, where

such inter-relationships are represented by the various functions constituting Σ. Also, an *interpretation* $\mathcal{I}$ for $\mathcal{O}$ is a pair, $\langle \Sigma, \mathcal{I}_o \rangle$, where Σ is an interpretation structure and $\mathcal{I}_o$ is the *interpretation function* for $\mathcal{I}$, i.e., a function that maps every expression in $Exp^{\mathcal{O}}$ into an object in Δ_o, and every literal in $\mathtt{L_{QL}}$ into a value in Δ_v, according to an appropriate set of conditions. For example, one condition imposes that $\bot_C$ is interpreted as an object associated through $\cdot^E$ to an empty set. As another example, if an entity e is interpreted as an object regarded as a relation R_e, then the expression e^- is interpreted as the inverse of R_e. For the full list of conditions, please refer to [9]. Finally, to define the semantics of logical axioms, MS resorts to the usual notion of satisfaction of an axiom with respect to an interpretation $\mathcal{I}$. Thus, for example, $\mathcal{I} \models (e_1 \sqsubseteq_C e_2)$ if $(e_1^{\mathcal{I}_o})^E$ and $(e_2^{\mathcal{I}_o})^E$ are defined, and $(e_1^{\mathcal{I}_o})^E \subseteq (e_2^{\mathcal{I}_o})^E$ where e_1, e_2 are expressions. As another notable example, $\mathcal{I} \models e_1 \neq e_2$ if $e_1^{\mathcal{I}_o} \neq e_2^{\mathcal{I}_o}$.

SPARQL. We concentrate on conjunctive queries (simply called *queries* in the following), expressed using SPARQL. Let $\mathcal{O}$ be an ontology and $\mathcal{V}$ a set of variables. We start by introducing the notion of query atom. A *query atom over* $\mathcal{O}$ (simply called *atom* in the following) has the same form of an axiom with the difference that its arguments belong to the set of *terms over* $\mathcal{O}$ *and* $\mathcal{V}$. The set of terms, denoted $Exp_{\mathcal{V}}^{\mathcal{O}}$, is defined similarly to $Exp^{\mathcal{O}}$, with the difference that its base set is $V_N^{\mathcal{O}} \cup \mathcal{V}$, rather than simply $V_N^{\mathcal{O}}$. Note that, similarly to axioms, atoms can be classified into TBox atoms and ABox atoms.

A conjunctive query q over an ontology is an expression of the form

$$\mathtt{select}\ x_1 \ldots x_n\ \mathtt{where}\ \{B\} \tag{1}$$

where $n \geq 1$, $x_1, \ldots, x_n$ are variables, called *distinguished variables*, n is the *arity* of the query, and B, called *body of the conjunctive query* q and denoted by *body*(q), is a non-empty conjunction of atoms. In the following we will use the notation $q(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \leftarrow B$, or simply $q(\boldsymbol{x}, \boldsymbol{y})$, to denote a query of the form above, where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y}$ is the tuple of variables that occur in B and do not belong to $\boldsymbol{x}$. Note that in SPARQL jargon, a conjunction of atoms is a *basic graph pattern*, i.e., a conjunction of RDF triples involving variables. However, for the sake of clarity and without loss of generality, instead of using the RDF syntax, we use here the DL syntax.

Queries are interpreted by relying on the notion of SPARQL *entailment regime*, as defined in the SPARQL 1.1. W3C standard specification[2]. Specifically, a SPARQL entailment regime defines (*i*) the syntax and the semantics of axioms constituting the queried ontology, (*ii*) the syntax of conjunctive queries considered legal for the regime, and (*iii*) the semantics of such queries, i.e., what are the answers to a query. The most typical SPARQL Entailment Regime for OWL 2 QL ontologies is the Direct Semantics Entalment Regime (DSER). However, based on its limitations discussed in the introduction, we introduce here a new entailment regime, called *Metamodeling Semantics Entailment Regime (MSER)*, which generalizes DSER as described in the following. As for (*i*), MSER assumes to deal with OWL 2 QL ontologies interpreted according to MS, which generalizes the Direct Semantics adopted by DSER. As for (*ii*), it considers as legal the whole set of queries defined above, while DSER restricts to queries where variables can occur only in positions of the same type. Thus, for example, in

[2] www.w3.org/TR/sparql11-entailment/

DSER, a variable occurring in object position cannot also occur in class position (e.g. in $\sqsubseteq_C$ axioms). As for (*iii*), MSER defines the answers to a query similarly to DSER, except that it uses MS for logical entailment. Specifically, given a tuple of variables $\boldsymbol{z} = (z_1, \ldots, z_n)$, a tuple of IRIs $\boldsymbol{w} = (w_1, \ldots, w_n)$, and a conjunction of atoms B, we denote by $\sigma[\boldsymbol{z} \rightarrow \boldsymbol{w}](B)$ the conjunction of atoms obtained from B by substituting each z_i in $\boldsymbol{z}$ with w_i in $\boldsymbol{w}$, for $i \in \{1, \ldots, n\}$. Now, let $\mathcal{O}$ be an ontology and $q(\boldsymbol{x}, \boldsymbol{y})$ a conjunctive query. An n-tuple of IRIs $\boldsymbol{t}$ is an *answer to* $q(\boldsymbol{x}, \boldsymbol{y})$ *over* $\mathcal{O}$ *under MSER* if there exists an m-tuple of IRIs $\boldsymbol{v}$ such that

$$\mathcal{O} \models \sigma[(\boldsymbol{x}, \boldsymbol{y}) \rightarrow (\boldsymbol{t}, \boldsymbol{v})](body(q)).$$

Also, $Ans(q, \mathcal{O})$ denotes the set of answers to q over $\mathcal{O}$ under MSER. It is worth noting that, in MSER as in DSER, the variables $\boldsymbol{y}$, although projected out from the answer, are required to be *bound* to the same m-tuple of constants $\boldsymbol{v}$, in every model of $\mathcal{O}$. Thus, $\boldsymbol{y}$ are treated as distinguished variables, and not as existential ones in classical logic.

Reducing consistency checking and query answering to Datalog evaluation

In this section, we show that consistency checking and query answering over OWL 2 QL ontologies under MSER can be reduced to the evaluation of a Datalog program. Note that, for the sake of simplicity, from now on we implicitly assume to deal with ontologies that do not contain data properties. But our results can be immediately extended to ontologies containing data properties as well.

Intuitively, the key ideas of our approach are the following. First, we define a finite number of inference rules which capture reasoning in OWL 2 QL, i.e. which are sound and complete with respect to logical implication in OWL 2 QL. Second, we define a translation function τ from the set of legal OWL 2 QL axioms to the set of instances of a database schema S^{ql}, where S^{ql} comprises a distinct relation for each distinct *form* of axiom, whose arity is the number of atomic expressions that occur in axioms of that form. Then, for example, inclusion axioms of the form $c_1 \sqsubseteq_C \exists r_2{}^-.c_2$, where c_1, c_2 are atomic classes, and r_2 is an atomic object property, are translated into tuples (c_1, r_2, c_2) of the relation isacCI. Third, we use τ to translate each inference rule to a Datalog rule over the predicates of S^{ql} and obtain a set of Datalog rules $\mathcal{P}^{ql}$.

Because of the lack of space, we cannot report on the whole set of inference rules, nor on the whole set of Datalog rules P^{ql}, but we next provide an overview on the main sets of rules it consists of. Thus, we define, in Table 1, the translation function τ from the set of OWL 2 QL axioms and atoms to the set of atoms over the schema $\mathcal{S}^{ql}$, where $x, y, c, c_1, c_2, r_1, r_2$ denote elements of V_N or variables in $\mathcal{V}$ and S^{ql} consists of the predicates that occur in the columns of Table 1 labeled with $\tau(\alpha)$ and of the binary predicates existR and existI. Then, $\mathcal{P}^{ql}$ consists of the sets of rules $\mathcal{P}^{ql}_{\mathcal{T}}$, $\mathcal{P}^{ql}_{\mathcal{T},\mathcal{A}}$, $\mathcal{P}^{ql}_{\neq}$, and $\mathcal{P}^{ql}_{\exists}$. The former are obtained by using τ to translate OWL 2 QL inference rules for deriving logically implied TBox axioms, ground ABox axioms, and inequalities, respectively, while the latter is obtained by translating inference rules that allow deriving first-order assertions that are not expressible in OWL 2 QL but can be logically implied

α	$\tau(\alpha)$	α	$\tau(\alpha)$	α	$\tau(\alpha)$
$c_1 \sqsubseteq_C c_2$	isacCC(c_1, c_2)	$c_1 \sqsubseteq_C \neg c_2$	disjcCC(c_1, c_2)	$r_1 \sqsubseteq_R r_2$	isarRR(r_1, r_2)
$c_1 \sqsubseteq_C \exists r_2.c_2$	isacCR(c_1, r_2, c_2)	$c_1 \sqsubseteq_C \neg\exists r_2$	disjcCR(c_1, r_2)	$r_1 \sqsubseteq_R r_2^-$	isarRI(r_1, r_2)
$c_1 \sqsubseteq_C \exists r_2^-.c_2$	isacCI(c_1, r_2, c_2)	$c_1 \sqsubseteq_C \neg\exists r_2^-$	disjcCI(c_1, r_2)	$r_1 \sqsubseteq_R \neg r_2$	disjrRR(r_1, r_2)
$\exists r_1 \sqsubseteq_C c_2$	isacRC$(r_1\ c_2)$	$\exists r_1 \sqsubseteq_C \neg c_2$	disjcRC(r_1, c_2)	$r_1 \sqsubseteq_R \neg r_2^-$	disjrRI(r_1, r_2)
$\exists r_1 \sqsubseteq_C \exists r_2.c_2$	isacRR(r_1, r_2, c_2)	$\exists r_1 \sqsubseteq_C \neg\exists r_2$	disjcRR(r_1, r_2)	$refl(r)$	refl(r)
$\exists r_1 \sqsubseteq_C \exists r_2^-.c_2$	isacRI(r_1, r_2, c_2)	$\exists r_1 \sqsubseteq_C \neg\exists r_2^-$	disjcRI(r_1, r_2)	$irr(r)$	irref(r)
$\exists r_1^- \sqsubseteq_C c_2$	isacIC$(r_1\ c_2)$	$\exists r_1^- \sqsubseteq_C \neg c_2$	disjcIC(r_1, c_2)	$c(x)$	instc(c, x)
$\exists r_1^- \sqsubseteq_C \exists r_2.c_2$	isacIR(r_1, r_2, c_2)	$\exists r_1^- \sqsubseteq_C \neg\exists r_2$	disjcIR(r_1, r_2)	$r(x, y)$	instr(r, x, y)
$\exists r_1^- \sqsubseteq_C \exists r_2^-.c_2$	isacII(r_1, r_2, c_2)	$\exists r_1^- \sqsubseteq_C \neg\exists r_2^-$	disjcII(r_1, r_2)	$x \neq y$	diff(x, y)

Table 1: Function τ

by $\mathcal{O}$, such as assertions of the form $\exists y \mid r(e, y)$, or $\exists y \mid r(y, e)$. In particular, we use the predicates existR and existI to encode such assertions, and the key point is that the use of such predicates allows us to avoid introducing existential variables within the head of the rule. Thus, the above mentioned assertions are translated into the tuples existR(r, e) and existI(r, e). Note that this, together with the fact that we consider inequalities and we interpret $\mathcal{O}$ under a more general semantics, significantly distinguishes our reduction to Datalog from the one proposed in [4]. In Table 2, we provide examples of Datalog rules belonging to each of the sets mentioned above, together with the inference rule they are obtained from.

	Datalog rule	Inference rule
$\mathcal{P}^{ql}_{\mathcal{T}}$	isacCI(c_1, r_2, c_2) :-isacCC(c_1, c_3), isacCI(c_3, r_2, c_2)	$\mathcal{O} \models c_1 \sqsubseteq c_3,\ \mathcal{O} \models c_3 \sqsubseteq_C \exists r_2.c_2 \ \Rightarrow\ \mathcal{O} \models c_1 \sqsubseteq \exists r_2.c_2$
$\mathcal{P}^{ql}_{\mathcal{T},\mathcal{A}}$	instr(r_1, x, y) :-instr(r_2, y, x), isarRI(r_2, r_1)	$\mathcal{O} \models r_2(y, x),\ \mathcal{O} \models r_2 \sqsubseteq_R r_1^- \ \Rightarrow\ \mathcal{O} \models r_1(x, y)$
$\mathcal{P}^{ql}_{\neq}$	diff(x, y) :-disjrRR(r_1, r_2),instr(r_1, x, z),instr(r_2, y, z)	$\mathcal{O} \models r_1 \sqsubseteq \neg r_2,\ \mathcal{O} \models r_2(x, z),\ \mathcal{O} \models r_1(y, z) \ \Rightarrow\ \mathcal{O} \models x \neq y$
$\mathcal{P}^{ql}_{\exists}$	existR(r, x) :-instr(r, x, y)	$\mathcal{O} \models r(x, y) \ \Rightarrow\ \mathcal{O} \models \exists z r(x, z)$

Table 2: Examples of rules in $\mathcal{P}^{ql}$

Finally, we show that both the problems of checking the consistency of $\mathcal{O}$ under MS and of answering q over $\mathcal{O}$ under MSER can be reduced to the evaluation a Datalog program. Note that, as customary in Datalog, we assume to deal with programs including a "special" intensional predicate, called *answer predicate*, here denoted Ans, and we assume that, given an instance D of a schema $\mathcal{S}$, the answer to a program Π over D, denoted $\Pi(D)$, is the extension of the answer predicate within the instance D' of $\mathcal{S}$ that results from the evaluation of Π over D.

Thus, let $D^{\mathcal{O}}$ be the set of facts computed by applying τ to each axiom in $\mathcal{O}$. Obviously, by construction, $D^{\mathcal{O}}$ is an instance of S^{ql} over elements of V_N. Also, let $\mathcal{P}^{inc}$ be the following fixed set of rules:

```
Ans():-instc(x,z),instc(y,z),disjc(x,y),
Ans():-instr(x,z,v),instc(y,z,v),disjr(x,y),
Ans():-instr(x,z,z),irref(x),
Ans():-⊥C(x),
Ans():-⊥R(x)
```

Intuitively, the rules of $\mathcal{P}^{inc}$ check for the existence of specific *violation patterns* in $D^{\mathcal{O}}$. Then, we have the following.

Theorem 1. *An ontology $\mathcal{O}$ is consistent under MS iff* $(\mathcal{P}^{ql} \cup \mathcal{P}^{inc})(D^{\mathcal{O}}) = \emptyset$.

Now, let q be a query over $\mathcal{O}$ and let r^q be the rule $\texttt{Ans}(\boldsymbol{x}) :- \tau(body(q))$, where, with a little abuse of notation, we denote by $\tau(B)$ the conjunction of the assertions obtained by applying τ to each atom in B. It is easy to see that, by construction, $\tau(B)$ is expressed over the alphabet of S^{ql} and involve only elements of V_N and $\mathcal{V}$. Also, let $\Pi^q = \mathcal{P}^{ql} \cup \mathcal{P}^q$.

Theorem 2. *Let $\mathcal{O}$ be a consistent ontology and q a conjunctive query. Then,*

$$Ans(q, \mathcal{O}) = \Pi^{\mathcal{O},q}(D^{\mathcal{O}}).$$

In a nutshell, the correctness of the two theorems relies on (*i*) the soundness and completeness of the set of OWL 2 QL inference rules, (*ii*) the semantics of MSER which treats existential variables as if they were distinguished, and, finally, (*ii*) the property that for every tuple t of elements of V_N, that is an instance of a predicate in $S^{ql} \setminus \{\texttt{existR}, \texttt{existI}\}$, t belongs to the minimum model of the Datalog program if and only if either $\mathcal{P}^{inc} \neq \emptyset$ and $\mathcal{O}$ is unsatisfiable, or t can be translated, via the inverse funtion of τ, into an axiom that belongs to every model of the ontology.

Clearly, the above theorems provide algorithms that are PTIME in the size of the ontology, and can be readily used by exploiting any off-the-shelf Datalog engine.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The dl-lite family and relations. *CoRR*, abs/1401.3487, 2014.
2. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
3. B. Glimm. Using SPARQL with RDFS and OWL entailment. In *RW-11*, pages 137–201. 2011.
4. G. Gottlob and A. Pieris. Beyond SPARQL under OWL 2 QL entailment regime: Rules to the rescue. In *Proc. of IJCAI 2015*, pages 2999–3007, 2015.
5. V. Gutiérrez-Basulto, Y. A. Ibáñez-García, R. Kontchakov, and E. V. Kostylev. Queries with negation and inequalities over lightweight ontologies. *J. of Web Semantics*, 35:184–202, 2015.
6. S. Kikot, R. Kontchakov, and M. Zakharyaschev. Conjunctive query answering with OWL 2 QL. In *Proc. of KR 2012*, 2012.
7. R. Kontchakov, M. Rezk, M. Rodriguez-Muro, G. Xiao, and M. Zakharyaschev. Answering SPARQL queries over databases under OWL 2 QL entailment regime. In *Proc. of ISWC 2014*, pages 552–567, 2014.
8. M. Lenzerini, L. Lepore, and A. Poggi. Answering metaqueries over hi(owl 2 ql) ontologies. In *Proc. of IJCAI 2016*, 2016.
9. M. Lenzerini, L. Lepore, and A. Poggi. A higher-order semantics for metaquerying in owl 2 ql. In *Proc. of KR 2016*, 2016.
10. H. J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23:155–212, 1984.