The Limits of Querying Ontologies

Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
rosati@dis.uniroma1.it

Abstract. We study query answering in Description Logics (DLs). In particular, we consider conjunctive queries, unions of conjunctive queries, and their extensions with safe negation or inequality, which correspond to well-known classes of relational algebra queries. We provide a set of decidability, undecidability and complexity results for answering queries of the above languages over various classes of Description Logics knowledge bases. In general, such results show that extending standard reasoning tasks in DLs to answering relational queries is unfeasible in many DLs, even in inexpressive ones. In particular: (i) answering even simple conjunctive queries is undecidable in some very expressive DLs in which standard DL reasoning is decidable; (ii) in DLs where answering (unions of) conjunctive queries is decidable, adding the possibility of expressing safe negation or inequality leads in general to undecidability of query answering, even in DLs of very limited expressiveness. We also highlight the negative consequences of these results for the integration of ontologies and rules. We believe that these results have important implications for ontology-based information access, in particular for the design of query languages for ontologies.

1 Introduction

Description Logics (DLs) [5] are currently playing a central role in the research on ontologies and the Semantic Web. Description Logics are a family of knowledge representation formalisms based on first-order logic (in fact, almost all DLs coincide with decidable fragments of function-free first-order logic with equality) and exhibiting well-understood computational properties. DLs are currently the most used formalisms for building ontologies, and have been proposed as standard languages for the specification of ontologies in the Semantic Web [24].

Recently, a lot of research and implementation work has been devoted to the extension of DL knowledge bases towards expressive query languages: one of main motivations for this effort is to provide users of the Semantic Web with more powerful ontology accessing tools than the ones deriving from the standard reasoning services provided by DL knowledge bases [17]. To this aim, relational database query languages have been considered as very promising query languages for DLs, in particular *conjunctive queries* (CQs) and *unions of*

conjunctive queries (UCQs). A lot of the current research in DLs is studying this problem, and many results have recently been obtained, both from the theoretical side (see Section 2) and the implementation side (see e.g., [21, 26]).

These studies are in principle very close to relational databases, not only because of the common query language, but also because, from the semantic viewpoint, query answering in DLs corresponds to a well-known problem in database theory, namely query answering over databases with incomplete information [18, 29], or query answering in databases under Open-World Assumption [31]. Then, of course, there is an important difference between the two settings, which lies in the different "schema language" adopted: DLs and relational schemas indeed correspond to two different subsets of function-free first-order logic. Nevertheless, there are well-known and important correspondences between DLs and (relational) data models (see e.g., [12, 8]): more generally, the relationship between DLs and databases is now quite well-assessed.

In this paper we study query answering over Description Logics knowledge bases. In particular, we do not restrict our attention to (unions of) conjunctive queries, and analyze several subclasses of first-order queries.[1] In particular, we consider CQs, UCQs, and their extensions with safe negation ($\text{CQ}^{\neg s}$s, $\text{UCQ}^{\neg s}$s) and inequality ($\text{CQ}^{\neq}$s, $\text{UCQ}^{\neq}$s), which correspond to well-known classes of relational algebra queries.

We provide a set of decidability, undecidability and complexity results for answering queries of the above languages over various classes of Description Logics knowledge bases. In particular, we mainly consider the following, rather inexpressive, DLs: *RDFS(DL)* [16], $\mathcal{EL}$ [4], *DL-Lite*$_R$ [9], and $\mathcal{AL}$ [5]. Many of the results obtained for such logics extend to more expressive DLs. A summary of the results obtained is reported in Figure 1 (Section 6).

In general, such results show that extending standard reasoning tasks in DLs to answering relational queries is unfeasible in many DLs, even in rather inexpressive ones. In particular:

- answering CQs and UCQs is already an unsolvable problem in decidable fragments of FOL, in particular in $\mathcal{L}^2$, the two-variable fragment of function-free FOL, which is very close to many DLs, and in which all standard DL reasoning tasks are decidable;
- in DLs where CQs and UCQs are decidable, adding safe negation generally leads to undecidability of query answering (even in DLs of very limited expressiveness);
- in the same way, adding inequality (and more generally, comparison operators) generally leads to undecidability of query answering.

We believe that these results have important implications for ontology-based information access, in particular for the design of query languages for ontologies, since they clearly highlight critical combinations of DL constructs and query constructs with respect to the decidability and complexity of query answering.

[1] We recall that, even for empty knowledge bases, the problem of answering arbitrary first-order queries is undecidable, both over finite and over unrestricted models [28].

Finally, we briefly point out that the above results have also important consequences in the design of rule layers for the Semantic Web, which is currently under standardization by the Rule Interchange Format (RIF) working group[2] of the World Wide Web Consortium (W3C). Indeed, almost all the rule formalisms proposed in this setting allow for posing relational queries (e.g., are able to express forms of Datalog queries). The results reported in this paper establish that not only recursion may lead to undecidability of reasoning in DL knowledge bases augmented with rules (which has been shown in [20, 13]), but also the presence of very restricted forms of nonrecursive negation and/or inequality in the rules might easily lead to undecidability of reasoning.

2 Description Logics and query languages

In this section we briefly introduce Description Logics and the query languages analyzed in the paper.

2.1 Description Logics

We now briefly recall Description Logics (DLs). We assume that the reader is familiar with first-order logic (FOL). For a more detailed introduction to DLs, we refer the reader to [5].

We start from an alphabet of concept names, an alphabet of role names and an alphabet of constant names. Concepts correspond to unary predicates in FOL, roles correspond to binary predicates, and constants corresponds to FOL constants.

Starting from concept and role names, *concept expressions* and *role expressions* can be constructed, based on a formal syntax. Different DLs are based on different languages concept and role expressions. Details on the concept and role languages for the DLs considered in this paper are reported below.

A *concept inclusion* is an expression of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept expressions. Similarly, a *role inclusion* is an expression of the form $R_1 \sqsubseteq R_2$, where R_1 and R_2 are role expressions.

An *instance assertion* is an expression of the form $A(a)$ or $P(a, b)$, where A is a concept expression, P is a role expression, and a, b are constant names. We do not consider complex concept and role expressions in instance assertions, since we are interested in data complexity of query answering, as explained below.

A *DL knowledge base* is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$, called the *TBox*, is a set of concept and role inclusions, and $\mathcal{A}$, called the *ABox*, is a set of instance assertions.

The DLs mainly considered in this paper are the following (from now on, we use the symbol A to denote a concept name and the symbol P to denote a role name):

[2] http://www.w3.org/2005/rules/

– *DL-Lite*$_{RDFS}$ is the DL whose language for concept and role expressions is defined by the following abstract syntax:

$$C_L ::= A \mid \exists R$$
$$C_R ::= A$$
$$R ::= P \mid P^-$$

and both concept inclusions of the form $C_L \sqsubseteq C_R$ and role inclusions $P_1 \sqsubseteq P_2$ are allowed in the TBox. Such DL corresponds to (a subset of) RDFS [1], the schema language for RDF.[3]

– *DL-Lite*$_R$ is the DL whose language for concept and role expressions is defined by the following abstract syntax:

$$C_L ::= A \mid \exists R$$
$$C_R ::= A \mid \neg C_R \mid \exists R$$
$$R ::= P \mid P^-$$

and both concept inclusions of the form $C_L \sqsubseteq C_R$ and role inclusions $R_1 \sqsubseteq R_2$ are allowed in the TBox.

– $\mathcal{EL}$ is the DL whose language for concept expressions is defined by the following abstract syntax:

$$C ::= A \mid C_1 \sqcap C_2 \mid \exists P.C$$

and only concept inclusions $C_1 \sqsubseteq C_2$ are allowed in the TBox.

– $\mathcal{AL}$ is the DL whose language for concept expressions is defined by the following abstract syntax:

$$C ::= A \mid \top \mid \bot \mid \neg A \mid C_1 \sqcap C_2 \mid \exists P \mid \forall P.C$$

and only concept inclusions $C_1 \sqsubseteq C_2$ are allowed in the TBox.

– $\mathcal{ALC}$ is the DL whose language for concept expressions is defined by the following abstract syntax:

$$C ::= A \mid \neg C \mid C_1 \sqcap C_2 \mid \exists P.C$$

and only concept inclusions $C_1 \sqsubseteq C_2$ are allowed in the TBox.

– $\mathcal{ALCHIQ}$ is the DL whose language for concept and role expressions is defined by the following abstract syntax:

$$C ::= A \mid \neg C \mid C_1 \sqcap C_2 \mid (\geq n\, R\, C)$$
$$R ::= P \mid P^-$$

and both concept inclusions $C_1 \sqsubseteq C_2$ and role inclusions $R_1 \sqsubseteq R_2$ are allowed in the TBox.

[3] *DL-Lite*$_{RDFS}$ is very similar to the description logic *RDFS*(*DL*) defined in [16].

Besides the inclusions defined by the concept and role expressions introduced above, in the following we will also consider role inclusions of the form $\neg P_1 \sqsubseteq P_2$, where P_1, P_2 are role names.

We give the semantics of DLs through the well-known translation of DL knowledge bases into FOL theories with counting quantifiers (see [5]).

$$\begin{aligned}
\rho_{fol}(\langle \mathcal{T}, \mathcal{A} \rangle) &= \rho_{fol}(\mathcal{T}) \cup \rho_{fol}(\mathcal{A}) \\
\rho_{fol}(C_1 \sqsubseteq C_2) &= \forall x.\rho_{fol}(C_1, x) \rightarrow \rho_{fol}(C_2, x) \\
\rho_{fol}(R_1 \sqsubseteq R_2) &= \forall x.\rho_{fol}(R_1, x, y) \rightarrow \rho_{fol}(R_2, x, y) \\
\rho_{fol}(A, x) &= A(x) \\
\rho_{fol}(\neg C, x) &= \neg \rho_{fol}(C, x) \\
\rho_{fol}(C_1 \sqcap C_2, x) &= \rho_{fol}(C_1, x) \wedge \rho_{fol}(C_2, x) \\
\rho_{fol}(\exists R, x) &= \exists y.\rho_{fol}(R, x, y) \\
\rho_{fol}(\exists R.C, x) &= \exists y.\rho_{fol}(R, x, y) \wedge \rho_{fol}(C, y) \\
\rho_{fol}((\geq n\, R\, C), x) &= \exists^{\geq n} y.\rho_{fol}(R, x, y) \wedge \rho_{fol}(C, y) \\
\rho_{fol}(P, x, y) &= P(x, y) \\
\rho_{fol}(P^-, x, y) &= P(y, x) \\
\rho_{fol}(\neg P, x, y) &= \neg P(x, y)
\end{aligned}$$

A *model* of a DL-KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a FOL model of $\rho_{fol}(\mathcal{K})$. Therefore, DLs inherit the classical semantics of FOL, hence, in every interpretation, constants and predicates are interpreted over a non-empty interpretation domain which is either finite or countably infinite. In this paper the only reasoning service we are interested in is query answering, whose semantics is defined in the following subsection.

We will also mention the following logics: (i) the DL $\mathcal{DLR}$ [11], which extends $\mathcal{ALCHIQ}$ essentially through the use of n-ary relations, and for which decidability results on query answering are known; (ii) $\mathcal{L}^2$, i.e., the two-variable fragment of function-free first-order logic with equality [7]; (iii) $\mathcal{C}^2$, i.e., the extension of the two-variable fragment $\mathcal{L}^2$ through *counting quantifiers* [15]. The above two fragments of FOL are very much related to DLs, since almost all DLs are subsets of $\mathcal{L}^2$ or $\mathcal{C}^2$. Indeed, it can be easily seen that the above mentioned DLs and fragments of FOL satisfy the following partial order with respect to their relative expressive power (see [5] for details):

$$\begin{aligned}
&\textit{DL-Lite}_{RDFS} \subset \textit{DL-Lite}_R \subset \mathcal{ALCHIQ} \subset \mathcal{DLR} \\
&\mathcal{EL} \subset \mathcal{ALC} \subset \mathcal{ALCHIQ} \subset \mathcal{C}^2 \\
&\mathcal{AL} \subset \mathcal{ALC} \subset \mathcal{L}^2 \subset \mathcal{C}^2 \\
&\textit{DL-Lite}_R \subset \mathcal{L}^2
\end{aligned}$$

2.2 Queries

We now introduce the query languages that will be considered in the paper. A *union of conjunctive queries* (UCQ) is an expression of the form

$$\{\boldsymbol{x} \mid \mathit{conj}_1(\boldsymbol{x}, \boldsymbol{c}) \vee \ldots \vee \mathit{conj}_m(\boldsymbol{x}, \boldsymbol{c})\} \tag{1}$$

where each $conj_i(\boldsymbol{x}, \boldsymbol{c})$ is an expression of the form $conj_i(\boldsymbol{x}, \boldsymbol{c}) = \exists \boldsymbol{y}.a_1 \wedge \ldots \wedge a_n$ in which each a_i is an atom whose arguments are terms from the sets of variables $\boldsymbol{x}$, $\boldsymbol{y}$, and from the set of constants $\boldsymbol{c}$ and such that each variable from $\boldsymbol{x}$ and $\boldsymbol{y}$ occurs in at least one atom a_i. The variables $\boldsymbol{x}$ are called the head variables (or distinguished variables) of the query.

A UCQ with safe negation ($\text{UCQ}^{\neg s}$) is an expression of the form (1) in which each a_i is either an atom or a negated atom (a negated atom is an expression of the form $\neg a$ where a is an atom) and such that in each $conj_i(\boldsymbol{x}, \boldsymbol{c})$ each variable from $\boldsymbol{x}$ and $\boldsymbol{y}$ occurs in at least one positive atom.

A UCQ with inequalities ($\text{UCQ}^{\neq}$) is an expression of the form (1) in which each $conj_i(\boldsymbol{x}, \boldsymbol{c})$ is a conjunction $\exists \boldsymbol{y}.a_1 \wedge \ldots \wedge a_n$ where each a_i is either an atom or an expression of the form $z \neq z'$, where z and z' are variables.

A UCQ with universally quantified negation ($\text{UCQ}^{\neg\forall}$) is a UCQ with negated atoms in which the variables that only appear in negated atoms are universally quantified. Formally, a $\text{UCQ}^{\neg\forall}$ is an expression of the form (1) in which each $conj_i(\boldsymbol{x}, \boldsymbol{c})$ is of the form

$$\exists \boldsymbol{y}.\forall \boldsymbol{z}.conj(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{c})$$

where *conj* is a conjunction of literals (atoms and negated atoms) whose arguments are terms from the sets of variables $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ and from the set of constants $\boldsymbol{c}$, in which each variable from $\boldsymbol{x}$ and $\boldsymbol{y}$ occurs in positive atoms, and *each variable in $\boldsymbol{z}$ only occurs in negated atoms*. An example of a $\text{UCQ}^{\neg\forall}$ is the following:

$$\{x \mid (\exists y, z.\forall w.r(x, y) \wedge \neg s(y, z) \wedge \neg t(w, z)) \ \vee \ (\exists y.\forall u.r(x, y) \wedge \neg s(x, u))\}$$

Notice that all the classes of queries above considered correspond to classes of relational algebra queries (hence they are classes of *domain-independent* first-order queries) [3].

We call a UCQ a *conjunctive query* (CQ) when $m = 1$. Analogously, we define the notions of CQ with negation ($\text{CQ}^{\neg}$), safe negation ($\text{CQ}^{\neg s}$), inequalities ($\text{CQ}^{\neq}$), and universally quantified negation ($\text{CQ}^{\neg\forall}$).

A *Boolean* CQ is a CQ without head variables, i.e., an expression of the form $conj_1(\boldsymbol{x}, \boldsymbol{c}) \vee \ldots \vee conj_m(\boldsymbol{x}, \boldsymbol{c})$. Since it is a sentence, i.e., a closed first-order formula, such a query is either true or false in a database. In the same way, we define the Boolean version of the other kinds of queries introduced above. Finally, the *arity* of a query is the number of head variables, while the *size* of a CQ q is the number of atoms in the body of q.

The semantics of queries in DL knowledge bases is immediately obtained by adapting the well-known notion of *certain answers* in indefinite databases (see e.g. [29]). Let q be a query of arity n, let $x_1, \ldots, x_n$ be its head variables, and let $\boldsymbol{c} = c_1, \ldots, c_n$ be a n-tuple of constants. We denote by $q(\boldsymbol{c})$ the Boolean query (i.e., the FOL sentence) obtained from q by replacing each head variable x_i with the constant c_i.

Let q be a query of arity n. A n-tuple $\boldsymbol{c}$ of constants occurring in $\mathcal{K}$ is a *certain answer* to q in $\mathcal{K}$ iff, for each model $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I}$ satisfies the sentence $q(\boldsymbol{c})$ (in this case we write $\mathcal{I} \models q(\boldsymbol{c})$). For a Boolean query q, we say that *true* is a certain answer to q in $\mathcal{K}$ iff, for each model $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I} \models q$.

Finally, in this paper we focus on *data complexity* of query answering, which is a notion borrowed from relational database theory [30]. First, we recall that there is a recognition problem associated with query answering, which is defined as follows. We have a fixed TBox $\mathcal{T}$ expressed in a DL $\mathcal{DL}$, and a fixed query q: the *recognition problem* associated to $\mathcal{T}$ and q is the decision problem of checking whether, given an ABox $\mathcal{A}$, and a tuple $\boldsymbol{c}$ of constants, we have that $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\boldsymbol{c})$. Notice that neither the TBox nor the query is an input to the recognition problem.

Let $\mathcal{C}$ be a complexity class. When we say that query answering for a certain DL $\mathcal{DL}$ *is in* $\mathcal{C}$ *with respect to data complexity*, we mean that the corresponding recognition problem is in $\mathcal{C}$. Similarly, when we say that query answering for a certain DL $\mathcal{DL}$ is $\mathcal{C}$*-hard with respect to data complexity*, we mean that the corresponding recognition problem is $\mathcal{C}$-hard.

2.3 Previous results on query answering in DLs

So far, only conjunctive queries and union of conjunctive queries have been studied in DLs. In particular, the first results in this field appear in [20], which proves that answering CQs and UCQs is decidable in $\mathcal{ALCNR}$, a DL whose expressiveness lies between $\mathcal{ALC}$ and $\mathcal{ALCHIQ}$. Then, in [11] it has been shown that answering CQs and UCQs is decidable in the very expressive Description Logic $\mathcal{DLR}$. The same paper also establishes undecidability of answering $\text{CQ}^{\neq}$s in $\mathcal{DLR}$, which so far is the only known result for DLs concerning the classes of queries (apart from CQs and UCQs) studied in this paper. Another decidability result appears in [21] and concerns answering conjunctive queries in $\mathcal{ALCIHQ}(\mathbf{D})$, which is the extension of $\mathcal{ALCHIQ}$ with concrete domains.

As for computational characterizations of query answering in DLs, the above mentioned work [20] has shown that the data complexity of answering CQs and UCQs in $\mathcal{ALCNR}$ is coNP-complete. Then, [27] presents the first algorithm for answering conjunctive queries over a description logic with transitive roles. Moreover, [10] provides a set of lower bounds for answering conjunctive queries in many DLs, while in [22] it has been shown that the complexity of answering conjunctive queries in $\mathcal{SHIQ}$ (which is the extension of $\mathcal{ALCHIQ}$ with transitive roles) is coNP-complete, for CQs in which transitive roles do not occur. This result (with the same restriction on roles occurring in queries) has been further extended in in [23] to unions of conjunctive queries, and in [14] to CQs for $\mathcal{SHOQ}$, a DL which extends $\mathcal{ALCHIQ}$ with transitive roles and *nominals*, but does not allow for expressing inverse roles anymore.

3 Results for positive queries

We start our analysis of query answering in DLs by considering, among the queries introduced in the previous section, the classes of positive queries. Thus, we first examine conjunctive queries, and then consider unions of conjunctive queries. In both cases, we identify sets of expressive features of a DL which are sufficient to make query answering undecidable.

Theorem 1. *Let $\mathcal{DL}$ be any DL such that: (i) its concept language allows for binary concept disjointness ($A_1 \sqsubseteq \neg A_2$), concept disjunction ($C_1 \sqcup C_2$), unqualified existential quantification ($\exists R$), and universal quantification ($\forall R.C$); (ii) it allows for concept inclusions and role inclusions of the form $\neg P_1 \sqsubseteq P_2$, where P_1, P_2 are role names. Then, answering UCQs in $\mathcal{DL}$ is undecidable.*

Proof (sketch). The proof is by a reduction from the unbounded tiling problem [6]. Let $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ be an instance of the tiling problem, where $\mathcal{S} = \{t_1, \ldots, t_n\}$ is a finite set of tiles, and $\mathcal{H}$ and $\mathcal{V}$ are binary relations over $\mathcal{S} \times \mathcal{S}$. For each $i \in \{1, \ldots, n\}$, let $\mathcal{T}_h^i = \{t_{h_1^i}, \ldots, t_{h_{k_i}^i}\}$ be the subset of $\mathcal{S}$ such that $\mathcal{T}_h^i = \{x \in \mathcal{S} \mid (t_i, x) \in \mathcal{H}\}$, and let $\mathcal{T}_v^i = \{t_{v_1^i}, \ldots, t_{v_{j_i}^i}\}$ be the subset of $\mathcal{S}$ such that $\mathcal{T}_v^i = \{x \in \mathcal{S} \mid (t_i, x) \in \mathcal{V}\}$.

Now let $\mathcal{T}$ be the following TBox (in which we use a set of concept names $T_1, \ldots, T_n$ in one-to-one correspondence with the elements $t_1, \ldots, t_n$ of $\mathcal{S}$, and the roles H, V and $\overline{V}$):

$$
\begin{array}{l}
\top \sqsubseteq \exists H \\
\top \sqsubseteq \exists V \\
\top \sqsubseteq T_1 \sqcup \ldots \sqcup T_n \\
T_i \sqsubseteq \neg T_j \quad \text{for each } i \neq j,\ i, j \in \{1, \ldots, n\} \\
T_i \sqsubseteq \forall H.T_{h_1^i} \sqcup \ldots \sqcup T_{h_{k_i}^i} \quad \text{for each } i \in \{1, \ldots, n\} \\
T_i \sqsubseteq \forall V.T_{v_1^i} \sqcup \ldots \sqcup T_{v_{j_i}^i} \quad \text{for each } i \in \{1, \ldots, n\} \\
\neg V \sqsubseteq \overline{V}
\end{array}
$$

and let q be the CQ $\exists x_1, x_2, y_1, y_2.H(x_1, x_2) \wedge V(x_1, y_1) \wedge H(y_1, y_2) \wedge \overline{V}(x_2, y_2)$. We prove that there exists a model M for $\mathcal{T}$ such that q is false in M iff the tiling problem instance $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ has a solution. □

Notice that the two-variable fragment $\mathcal{L}^2$ satisfies the conditions of Theorem 1 (in the sense that a DL satisfying the conditions of Theorem 1 can be translated into an equivalent $\mathcal{L}^2$ theory), which implies the following property.

Corollary 1. *Answering CQs in $\mathcal{L}^2$ is undecidable.*

Actually, the above property shows that answering CQs is undecidable already in a very small fragment of $\mathcal{L}^2$.

We point out that, although the syntax of the description logic $\mathcal{DLR}$ satisfies the conditions of the above theorem, such theorem actually does not apply to $\mathcal{DLR}$, due to a different interpretation of negated roles in $\mathcal{DLR}$ with respect to the standard semantics [11].

Then, we analyze unions of conjunctive queries. The next two theorems identify two sets of DL constructs which are sufficient to make query answering undecidable.

Theorem 2. *Let $\mathcal{DL}$ be any DL whose concept language allows for unqualified existential quantification ($\exists P$) and concept disjunction ($C_1 \sqcup C_2$), and which allows for concept inclusions and role inclusions of the form $\neg P_1 \sqsubseteq P_2$, where P_1, P_2 are role names. Then, answering UCQs in $\mathcal{DL}$ is undecidable.*

Proof (sketch). The proof is analogous to the proof of Theorem 1. The only difference is that the concept inclusions defined in the above proof and involving either concept disjointness or universal quantification are encoded by suitable Boolean CQs that are added to the query, thus producing a UCQ. □

The proof of the next theorem is based on a reduction from the word problem for semigroups to answering UCQs in a description logic $\mathcal{DL}$.

Theorem 3. *Let $\mathcal{DL}$ be any DL whose concept language allows for unqualified existential quantification ($\exists R$) and inverse roles ($\exists P^-$), and which allows for concept inclusions and role inclusions of the form $\neg P_1 \sqsubseteq P_2$, where P_1, P_2 are role names. Then, answering UCQs in $\mathcal{DL}$ is undecidable.*

Then, we provide an upper bound for the data complexity of answering UCQs in the DL $\mathcal{EL}$ (we recall that hardness with respect to PTIME has been proved in [9]).

Theorem 4. *Answering UCQs in $\mathcal{EL}$ is in* PTIME *with respect to data complexity.*

Proof (sketch). We prove the thesis by defining a query reformulation algorithm for $\mathcal{EL}$. More precisely, we define an algorithm perfectRefEL that takes as input an $\mathcal{EL}$ TBox $\mathcal{T}$ and a UCQ q, and computes (in a finite amount of time) a positive Datalog query q' which constitutes a *perfect rewriting* [19] of the query q, in the sense that, for each ABox $\mathcal{A}$, the set of certain answers of q in $\langle \mathcal{T}, \mathcal{A} \rangle$ is equal to the answers returned by the standard evaluation of the Datalog query q' in the ABox $\mathcal{A}$ considered as a relational database. Since the evaluation of a positive Datalog query is in PTIME with respect to data complexity, and since the computation of the reformulation q' is independent of the data, it follows that the data complexity of answering UCQs in $\mathcal{EL}$ is in PTIME. □

4 Results for queries with inequality

We now give decidability and complexity results for answering queries with inequality in DL knowledge bases. We first examine CQ$^{\neq}$s, then we turn our attention to UCQ$^{\neq}$s.

We first prove undecidability of answering CQ$^{\neq}$s in $\mathcal{AL}$.

Theorem 5. *Answering CQ$^{\neq}$s in $\mathcal{AL}$ is undecidable.*

Proof (sketch). Again, the proof is by reduction from the tiling problem. Let $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ be an instance of the tiling problem, where $\mathcal{S} = \{t_1, \ldots, t_n\}$ is a finite set of tiles, $\mathcal{H}$ and $\mathcal{V}$ are binary relations over $\mathcal{S} \times \mathcal{S}$. For each $i \in \{1, \ldots, n\}$, let $\mathcal{T}_h^i = \{t_{h_1^i}, \ldots, t_{h_{k_i}^i}\}$ be the subset of $\mathcal{S}$ such that $\mathcal{T}_h^i = \{x \in \mathcal{S} \mid (t_i, x) \notin \mathcal{H}\}$, and let $\mathcal{T}_v^i = \{t_{v_1^i}, \ldots, t_{v_{j_i}^i}\}$ be the subset of $\mathcal{S}$ such that $\mathcal{T}_v^i = \{x \in \mathcal{S} \mid (t_i, x) \notin \mathcal{V}\}$.

Now let $\mathcal{T}$ be the following TBox:

$$\begin{array}{l}
\top \sqsubseteq \exists H \\
\top \sqsubseteq \exists V \\
\neg T_1 \sqcap \ldots \sqcap \neg T_n \sqsubseteq \bot \\
T_i \sqsubseteq \neg T_j \quad \text{for each } i \neq j,\ i,j \in \{1,\ldots,n\} \\
T_i \sqsubseteq \forall H.\neg T_{h_1^i} \sqcap \ldots \sqcap \neg T_{h_{k_i}^i} \quad \text{for each } i \in \{1,\ldots,n\} \\
T_i \sqsubseteq \forall V.\neg T_{v_1^i} \sqcap \ldots \sqcap \neg T_{v_{j_i}^i} \quad \text{for each } i \in \{1,\ldots,n\}
\end{array}$$

and let $q = \exists x_1, x_2, y_1, y_2.H(x_1, x_2) \wedge V(x_1, y_1) \wedge H(y_1, y_2) \wedge V(x_2, y_2') \wedge y_2 \neq y_2'$. We prove that there exists a model M for $\mathcal{T}$ such that q is false in M iff the tiling problem instance $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ has a solution. □

The above theorem improves the undecidability result of containment of CQ$^{\neq}$s presented in [11].

Then, we consider the DL *DL-Lite*$_R$: for this logic, we prove the following hardness result.

Theorem 6. *Answering CQ$^{\neq}$s in DL-Lite*$_R$ *is* coNP*-hard with respect to data complexity.*

Proof (sketch). The proof is by reduction from satisfiability of a 3-CNF propositional formula. The reduction is inspired by an analogous reduction reported in [2] which proves coNP-hardness of answering CQ$^{\neq}$s using views. □

Finally, we show a (quite obvious) property which allows us to immediately define upper bounds for answering CQ$^{\neq}$s in the DLs *DL-Lite*$_{RDFS}$ and $\mathcal{EL}$. In the following, we call *singleton interpretation for* $\mathcal{K}$ an interpretation whose domain Δ is a singleton $\{d\}$, all constants occurring in $\mathcal{K}$ are interpeted as d, the interpretation of every concept name A is Δ, and the interpretation of every role name P is $\Delta \times \Delta$.

Theorem 7. *Let $\mathcal{DL}$ be a DL such that, for each DL-KB $\mathcal{K}$, any singleton interpretation for $\mathcal{K}$ is a model of $\mathcal{K}$. Then, answering CQ$^{\neq}$s in $\mathcal{DL}$ has the same complexity as answering CQs.*

It is immediate to see that both *DL-Lite*$_{RDFS}$ and $\mathcal{EL}$ satisfy the condition of the above theorem.[4] This allows us to extend the computational results of answering CQs to the case of CQ$^{\neq}$s for both the above DLs.

For UCQ$^{\neq}$s, we start by considering DLs allowing for inverse roles and unqualified existential quantification in concept expressions.

The proof of the next theorem is based on a reduction from the word problem for semigroups.

[4] Notice, however, that this property does not hold anymore if the Unique Name Assumption (UNA) [5] is adopted in such description logics (i.e., different constant names must be interpreted as different elements of the domain). Anyway, all the other results of this paper also hold in the case when the DL adopts the UNA.

Theorem 8. *Let $\mathcal{DL}$ be any DL whose concept language allows for unqualified existential quantification ($\exists R$) and inverse roles ($\exists P^-$), and which allows for concept and role inclusions in the TBox. Then, answering UCQ$^{\neq}$s in $\mathcal{DL}$ is undecidable.*

Notice that the above theorem holds for the description logic *DL-Lite*$_R$.

Then, we turn our attention to the description logic $\mathcal{EL}$, and prove a result analogous to the previous theorem (whose proof is obtained by slightly modifying the reduction of the previous proof).

Theorem 9. *Answering UCQ$^{\neq}$s in $\mathcal{EL}$ is undecidable.*

Finally, in a similar way we prove the same undecidability result for the description logic $\mathcal{AL}$.

Theorem 10. *Answering UCQ$^{\neq}$s in $\mathcal{AL}$ is undecidable.*

Actually, the above theorem implies undecidability of answering UCQ$^{\neq}$s already in $\mathcal{FL}^-$, which is obtained from $\mathcal{AL}$ disallowing negation on atomic concepts [5].

Finally, we turn our attention to answering UCQ$^{\neq}$s in *DL-Lite*$_{RDFS}$, and are able to easily prove the following upper bound.

Theorem 11. *Answering UCQ$^{\neq}$s in DL-Lite$_{RDFS}$ is in* LOGSPACE *with respect to data complexity.*

5 Results for queries with negation

In this section, among the queries introduced in Section 2, we consider the classes containing forms of negation. So we first consider CQ$^{\neg s}$s, then UCQ$^{\neg s}$s, and finally UCQ$^{\neg\forall}$s.

We start by proving that answering CQ$^{\neg s}$s is undecidable in the description logic $\mathcal{AL}$ (the proof of next theorem is again by reduction from the tiling problem, in a way similar to the proof of Theorem 5).

Theorem 12. *Answering CQ$^{\neg s}$s in $\mathcal{AL}$ is undecidable.*

Then, we show a hardness result for answering CQ$^{\neg s}$s in *DL-Lite*$_R$.

Theorem 13. *Answering CQ$^{\neg s}$s in DL-Lite$_R$ is* coNP*-hard with respect to data complexity.*

Proof (sketch). We prove the thesis by a reduction from graph 3-colorability. Let $G = (V, E)$ be a directed graph. We define the *DL-Lite*$_R$-KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is the following TBox (independent of the graph instance):

$$
\begin{array}{lll}
\mathit{Red} \sqsubseteq \neg \mathit{Green} & \exists \mathit{EdgeR} \sqsubseteq \mathit{Red} & \exists \mathit{EdgeR}^- \sqsubseteq \neg \mathit{Red} \\
\mathit{Red} \sqsubseteq \neg \mathit{Blue} & \exists \mathit{EdgeG} \sqsubseteq \mathit{Green} & \exists \mathit{EdgeG}^- \sqsubseteq \neg \mathit{Green} \\
\mathit{Green} \sqsubseteq \neg \mathit{Blue} & \exists \mathit{EdgeB} \sqsubseteq \mathit{Blue} & \exists \mathit{EdgeB}^- \sqsubseteq \neg \mathit{Blue}
\end{array}
$$

and $\mathcal{A}$ is the following ABox: $\mathcal{A} = \{Edge(v_1, v_2) \mid (v_1, v_2) \in E\}$. Finally, let q be the CQ$^{\neg s}$ $\exists x, y.Edge(x, y) \wedge \neg EdgeR(x, y) \wedge \neg EdgeG(x, y) \wedge \neg EdgeB(x, y)$. We prove that G is 3-colorable iff *true* is not a certain answer to q in $\mathcal{K}$. □

Notice that the above theorem actually proves coNP-hardness of answering CQ$^{\neg s}$s already for DLs much less expressive than *DL-Lite*$_R$, i.e., for the DL obtained from *DL-Lite*$_R$ by eliminating both role inclusions and existential quantification on the right-hand side of concept inclusions.

Finally, we turn our attention to the description logics *DL-Lite*$_{RDFS}$ and $\mathcal{EL}$, and prove a property analogous to Theorem 7. We call *saturated interpretation for* $\mathcal{K}$ an interpretation whose domain Δ is in one-to-one correspondence with the constants occurring in $\mathcal{K}$, all constants are interpreted according to such correspondence, the interpretation of every concept name A is Δ, and the interpretation of every role name P is $\Delta \times \Delta$.

Theorem 14. *Let $\mathcal{DL}$ be a DL such that, for each DL-KB $\mathcal{K}$, any saturated interpretation for $\mathcal{K}$ is a model of $\mathcal{K}$. Then, answering CQ$^{\neg s}$s in $\mathcal{DL}$ has the same complexity as answering CQs.*

It is immediate to see that both *DL-Lite*$_{RDFS}$ and $\mathcal{EL}$ satisfy the condition of the above theorem. This allows us to extend the computational results of answering CQs to the case of CQ$^{\neg s}$s for both the above DLs.

Then, we analyze UCQ$^{\neg s}$s. First, we prove a very strong undecidability result.

Theorem 15. *Let $\mathcal{DL}$ be any DL allowing for unqualified existential quantification ($\exists P$) in concept expressions. Answering UCQ$^{\neg s}$s in $\mathcal{DL}$ is undecidable.*

Proof (sketch). Given a tiling problem instance $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ as in the proof of Theorem 1, we define the following TBox $\mathcal{T}$: $\{\top \sqsubseteq Point, \top \sqsubseteq \exists H, \top \sqsubseteq \exists V\}$. Then, let q be the UCQ$^{\neg s}$ containing the following conjunctions:

$$\exists x.Point(x) \wedge \neg T_1(x) \wedge \ldots \wedge \neg T_n(x)$$
$$\exists x.T_i(x) \wedge T_j(x) \quad \text{for each } i \neq j,\ i, j \in \{1, \ldots, n\}$$
$$\exists x_1, x_2, y_1, y_2.H(x_1, x_2) \wedge V(x_1, y_1) \wedge H(y_1, y_2) \wedge \neg V(x_2, y_2)$$
$$\exists x, y.T_i(x) \wedge H(x, y) \wedge \neg T_{h_1^i}(y) \wedge \ldots \wedge \neg T_{h_{k_i}^i}(y) \quad \text{for each } i \in \{1, \ldots, n\}$$
$$\exists x, y.T_i(x) \wedge V(x, y) \wedge \neg T_{v_1^i}(y) \wedge \ldots \wedge \neg T_{v_{j_i}^i}(y) \quad \text{for each } i \in \{1, \ldots, n\}$$

We prove that there exists a model M for $\mathcal{T}$ such that q is false in M iff the tiling problem instance $(\mathcal{S}, \mathcal{H}, \mathcal{V})$ has a solution. □

The above theorem implies that answering UCQ$^{\neg s}$s is undecidable in all the DLs analyzed in this paper, with the exception of *DL-Lite*$_{RDFS}$, in which the concept inclusions defined in the above proof cannot be expressed. So we turn our attention to answering UCQ$^{\neg s}$s in *DL-Lite*$_{RDFS}$, and prove the following computational characterization.

Theorem 16. *Answering UCQ$^{\neg s}$s in DL-Lite*$_{RDFS}$ *is* coNP*-complete with respect to data complexity.*

	CQ	UCQ	$CQ^{\neq}$	$UCQ^{\neq}$	$CQ^{\neg s}$	$UCQ^{\neg s}$	$UCQ^{\neg\forall}$
DL-Lite$_{RDFS}$	$\leq_{\text{LOGSPACE}}$ [10]	$\leq_{\text{LOGSPACE}}$ [10]	$\leq_{\text{LOGSPACE}}$ [10]+Thm. 7	$\leq_{\text{LOGSPACE}}$ Thm. 11	$\leq_{\text{LOGSPACE}}$ [10]+Thm. 14	$=_{\text{coNP}}$ Thm. 16	UNDEC. Thm. 17
DL-Lite$_R$	$\leq_{\text{LOGSPACE}}$ [10]	$\leq_{\text{LOGSPACE}}$ [10]	$\geq_{\text{coNP}}$ Thm. 6	UNDEC. Thm. 8	$\geq_{\text{coNP}}$ Thm. 13	UNDEC. Thm. 15	UNDEC. Thm. 17
$\mathcal{EL}$	$=_{\text{PTIME}}$ $\geq$: [10] $\leq$: Thm. 4	$=_{\text{PTIME}}$ $\geq$: [10] $\leq$: Thm. 4	$=_{\text{PTIME}}$ $\geq$: [10] $\leq$: Thm.7+4	UNDEC. Thm. 9	$=_{\text{PTIME}}$ $\geq$: [10] $\leq$: Thm.14+4	UNDEC. Thm. 15	UNDEC. Thm. 17
$\mathcal{AL}$, $\mathcal{ALC}$, $\mathcal{ALCHIQ}$	$=_{\text{coNP}}$ $\geq$: [10] $\leq$: [22]	$=_{\text{coNP}}$ $\geq$: [10] $\leq$: [23]	UNDEC. Thm. 5	UNDEC. Thm. 10	UNDEC. Thm. 12	UNDEC. Thm. 15	UNDEC. Thm. 17
$\mathcal{DLR}$	$\geq$ coNP[10] DECID. [11]	$\geq$ coNP[10] DECID. [11]	UNDEC. [11]	UNDEC. [11]	UNDEC. Thm. 12	UNDEC. Thm. 15	UNDEC. Thm. 17
$\mathcal{L}^2$	UNDEC. Thm. 1	UNDEC. Thm. 1	UNDEC. Thm. 1	UNDEC. Thm. 1	UNDEC. Thm. 1	UNDEC. Thm. 1	UNDEC. Thm. 1

Fig. 1. Summary of results.

Finally, we turn our attention to unions of conjunctive queries with universally quantified negation, and show that answering queries of this class is undecidable in every DL.

The proof of the next theorem is based on a reduction from the word problem for semigroups.

Theorem 17. *Answering $UCQ^{\neg\forall}$s is undecidable in every DL.*

This result identifies a very restricted fragment of FOL queries for which query answering is undecidable, independently of the form of the knowledge base/FOL theory to which they are posed.

6 Summary of results and conclusions

The table displayed in Figure 1 summarizes the results presented in this paper (as well as the already known results for the DLs considered in this paper). In the table, each column corresponds to a different query language, while each row corresponds to a different DL. Each cell reports the data complexity of query answering in the corresponding combination of DL and query language. If the problem is decidable, then hardness ($\geq$) and/or membership ($\leq$) and/or completeness ($=$) results are reported (with reference to the Theorem or the publication which proves the result).

Besides the considerations reported in the introduction about these results, a further interesting aspect is the existence of cases in which adding the possibility of expressing unions changes the complexity of query answering. E.g., in the case of $\mathcal{EL}$, adding the possibility of expressing unions (i.e., going from CQs to UCQs) in the presence of safe negation or inequality makes query answering undecidable, while it is decidable in the absence of unions in queries.

These results are of course only a small step towards a thorough analysis of expressive query languages in DLs. Among the DLs and the query languages

studied in this paper, two interesting open problems concern the full computational characterization of answering $\text{CQ}^{\neg s}$s and $\text{CQ}^{\neq}$s in *DL-Lite*$_R$. Actually, even decidability of query answering in these cases is still unknown.

Finally, we remark that the present research is related to the work reported in [25], which presents a similar analysis for the same query classes in relational databases with incomplete information (instead of DL knowledge bases). However, we point out that none of the results reported in the present paper can be (either directly or indirectly) derived from the proofs of the results in [25], due to the deep differences between the database schema language considered there and the DLs examined in this paper.

Acknowledgments The author wishes to warmly thank Giuseppe De Giacomo and Maurizio Lenzerini for their precious comments. This research has been partially supported by FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT).

References

1. `http://www.w3.org/TR/rdf-schema/`.
2. S. Abiteboul and O. Duschka. Complexity of answering queries using materialized views. unpublished manuscript, available at `ftp://ftp.inria.fr/INRIA/Projects/gemo/gemo/GemoReport-383.pdf`, 1999.
3. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.
4. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI 2005*, pages 364–369, 2005.
5. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
6. R. Berger. The undecidability of the dominoe problem. *Mem. Amer. Math. Soc.*, 66:1–72, 1966.
7. A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.
8. A. Borgida, M. Lenzerini, and R. Rosati. Description logics for data bases. In Baader et al. [5], chapter 16, pages 462–484.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, pages 602–607, 2005.
10. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, 2006.
11. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, pages 149–158, 1998.
12. D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *J. of Artificial Intelligence Research*, 11:199–240, 1999.

13. D. Calvanese and R. Rosati. Answering recursive queries under keys and foreign keys is undecidable. In *Proc. of KRDB 2003*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-79/`, 2003.
14. B. Glimm, I. Horrocks, and U. Sattler. Conjunctive query answering for description logics with transitive roles. In *Proc. of DL 2006*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-189`, 2006.
15. E. Grädel, P. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
16. B. C. Grau. A possible simplification of the semantic web architecture. In *Proc. of the 13th Int. World Wide Web Conf. (WWW 2004)*, pages 704–713, 2004.
17. I. Horrocks and S. Tessaris. Querying the Semantic Web: a formal approach. In *Proc. of ISWC 2002*, volume 2342 of *LNCS*, pages 177–191. Springer, 2002.
18. T. Imielinski and W. L. Jr. Incomplete information in relational databases. *J. of the ACM*, 31(4):761–791, 1984.
19. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.
20. A. Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1–2):165–209, 1998.
21. B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, University of Karlsruhe, 2005.
22. M. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of AAAI 2006*, 2006.
23. M. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of answering unions of conjunctive queries in $\mathcal{SHIQ}$. In *Proc. of DL 2006*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-189`, 2006.
24. P. F. Patel-Schneider, P. J. Hayes, I. Horrocks, and F. van Harmelen. OWL web ontology language; semantics and abstract syntax. W3C candidate recommendation, http://www.w3.org/tr/owl-semantics/, november 2002.
25. R. Rosati. On the decidability and finite controllability of query processing in databases with incomplete information. In *Proc. of PODS 2006*, pages 356–365, 2006.
26. E. Sirin and B. Parsia. Optimizations for answering conjunctive abox queries: First results. In *Proc. of DL 2006*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-189`, 2006.
27. S. Tessaris. *Questions and Answers: Reasoning and Querying in Description Logic*. PhD thesis, University of Manchester, Department of Computer Science, Apr. 2001.
28. B. Trahktenbrot. Impossibility of an algorithm for the decision problem in finite classes. *Transactions of the American Mathematical Society*, 3:1–5, 1963.
29. R. van der Meyden. The complexity of querying indefinite data about linearly ordered domains. *J. of Computer and System Sciences*, 54(1):113–135, 1997.
30. M. Y. Vardi. The complexity of relational query languages. In *Proc. of STOC'82*, pages 137–146, 1982.
31. M. Y. Vardi. On the integrity of databases with incomplete information. In *Proc. of PODS'82*, pages 252–266, 1982.