Epistemic First-Order Queries over Description Logic Knowledge Bases*

Diego Calvanese[1], Giuseppe De Giacomo[2], Domenico Lembo[2], Maurizio Lenzerini[2], Riccardo Rosati[2]

[1] Faculty of Computer Science
Free University of Bozen-Bolzano
Piazza Domenicani 3
39100 Bolzano, Italy
calvanese@inf.unibz.it

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

Abstract

Querying Description Logic knowledge bases has received great attention in the last years. The need of coping with incomplete information is the distinguishing feature with respect to querying databases. Due to this feature, we have to deal with two conflicting needs: on the one hand, we would like to query the knowledge base with sophisticated mechanisms provided by full first-order logic as in databases; on the other hand, the presence of incomplete information makes query answering a much more difficult task than in databases. In this paper we advocate the use of an epistemic first-order query language, which is able to incorporate closed-world reasoning on demand, as a means for expressing sophisticated queries over Description Logic knowledge bases. We show that through a subset of this language, called *EQL-Lite*, we are able to formulate full first-order queries over Description Logic knowledge bases, while keeping computational complexity of query answering under control. In particular, we show that *EQL-Lite* queries over *DL-Lite* knowledge bases are first-order reducible (i.e., can be compiled into SQL) and hence can be answered in LogSpace through standard database technologies.

1 Introduction

Querying Description Logic (DL) knowledge bases has received great attention in the last years. Indeed, the definition of suitable query languages, and the design of query answering algorithms is arguably one of the crucial issues in applying DLs to ontology management and to the Semantic Web [9].

Answering queries in DLs must take into account the open-world semantics of such logics, and is therefore much more difficult than in Databases. For example,

*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).

while First Order Logic (FOL) is the basis of any query language (e.g., relational algebra and SQL) for relational databases [1], it is well-known that answering FOL queries posed to DL knowledge bases is undecidable[1]. More precisely, to the best of our knowledge, the most expressive class of queries that go beyond instance checking, and for which decidability of query answering has been proved in DLs, is the class of union of conjunctive queries [5, 14, 15]. This restriction on the query language may constitute a serious limitation to the adoption of DLs technology in information management tasks, such as those required in Semantic Web applications.

The open-world semantics of DLs on one hand is essential for representing incomplete information, but on the other hand may complicate the task of interpreting the answers by the users, or may call for the need of reasoning about the incompleteness of the knowledge base. For example, knowing that there are no parents with only female children, one might become interested in asking for all parents whose known chidren are all female. Note that querying mechanisms such as the one mentioned in the example go beyond FOL.

To summarize, due to the need of coping with incomplete information in DL knowledge bases, two conflicting requirements arise in querying: on one hand, we would like to query the knowledge base with powerful mechanisms that are able to reason about incompleteness, and on the other hand we aim at query languages that are both close in expressive power to FOL, and decidable (and, possibly, computationally tractable).

This paper presents the following contributions:

- We define a new query language for DL knowledge bases, called *EQL* (see Section 2), based on a variant of the well-known first-order modal logic of knowledge/belief [12, 17, 13]. The language incorporates a modal operator $\mathbf{K}$, that is used to formalize the epistemic state of the knowledge base. Informally, the formula $\mathbf{K}\phi$ is read as "ϕ is known to hold (by the knowledge base)". Using this operator, we are able to pose queries that reason about the incompleteness of information represented by the knowledge base. For instance, a user can express queries that are able to incorporate closed-world reasoning on demand.
- We show (see Section 3) that through a subset of this language, called *EQL-Lite*($\mathcal{Q}$), we are able to formulate queries that are interesting both from the expressive power point of view, and from the computational complexity perspective. Queries in *EQL-Lite*($\mathcal{Q}$) have atoms that are expressed using a specific query language $\mathcal{Q}$, and enjoy the property that they can be evaluated essentially with the same data complexity (i.e., measured wrt the size of the ABox only) as $\mathcal{Q}$.
- We investigate the properties of *EQL-Lite*($\mathcal{Q}$) for the cases of $\mathcal{ALCQI}$ (Section 4) and *DL-Lite* (Section 5) knowledge bases, under the assumption that $\mathcal{Q}$ is the language of unions of conjunctive queries. We study the data complexity of query answering for both cases. In particular, we show that answering such queries over *DL-Lite* knowledge bases is LogSpace, and, notably, can be reduced to

[1] Indeed, query answering can be easily reduced to validity checking in FOL.

evaluating FOL queries over the ABox, when considered as a database. It follows that query processing in this setting can be done through standard database technologies.

2 Epistemic query language

We make use of a variant of the well-known first-order modal logic of knowledge/belief [12, 16, 13] (see also [7, 11]), here called *EQL*. The language *EQL* is a first-order modal language with equality and with a single modal operator **K**, constructed from concepts, interpreted as unary predicates, and roles/relations, interpreted as binary/n-ary predicates, and an infinitely countable set of disjoint constants (a.k.a., *standard names* [13]) corresponding to elements of an infinite countable fixed domain Δ. In *EQL*, the modal operator is used to formalize the epistemic state of the knowledge base. Informally, the formula $\mathbf{K}\phi$ should be read as "ϕ is known to hold (by the knowledge base)".

Under this view, a DL knowledge base corresponds to a finite set of FOL sentences (i.e., closed FOL formulas), capturing what is known about the world. We query such information by using (possibly open) *EQL* formulas possibly involving **K**.

In the following, we use the symbol c (possibly with subscript) to denote a constant, the symbol x to denote a variable, and ϕ, ψ to denote arbitrary formulas.

A *world* is a first-order interpretation (over Δ). An *epistemic interpretation* is a pair E, w, where E is a (possibly infinite) set of worlds, and w is a world in E. We inductively define when a sentence (i.e., a closed formula) ϕ *is true in an interpretation* E, w (or, is true in w and E), written $E, w \models \phi$, as follows:[2]

$$
\begin{array}{lll}
E, w \models c_1 = c_2 & \text{iff} & c_1 = c_2 \\
E, w \models P(c_1, \ldots, c_n) & \text{iff} & w \models P(c_1, \ldots, c_n) \\
E, w \models \phi_1 \wedge \phi_2 & \text{iff} & E, w \models \phi_1 \text{ and } E, w \models \phi_2 \\
E, w \models \neg\phi & \text{iff} & E, w \not\models \phi \\
E, w \models \exists x.\psi & \text{iff} & E, w \models \psi^x_c \text{ for some constant } c \\
E, w \models \mathbf{K}\psi & \text{iff} & E, w' \models \psi \text{ for every } w' \in E
\end{array}
$$

Formulas without occurrences of **K** are said to be *objective* since they talk about what is true. Observe that in order to establish if $E, w \models \phi$, where ϕ is an objective formula, we have to look at w but not at E: we only need the FOL interpretation w. All assertions in the DL knowledge base are indeed objective sentences.

Instead, formulas where each occurrence of predicates and of the equality is in the scope of the **K** operator are said to be *subjective*, since they talk about what is known to be true. Observe that for a subjective sentence ϕ, in order to establish if $E, w \models \phi$ we do not have to look at w but only at E. We use such formulas to query what the knowledge base knows. In other words, through subjective sentences we do not query information about the world represented by the knowledge base; instead, we query the epistemic state of the knowledge base itself. Obviously there are formulas that

[2] For a formula ϕ with free variables $x_1, \ldots, x_n$, we use $\phi^{x_1,\ldots,x_n}_{c_1,\ldots,c_n}$ to denote the formula obtained from ϕ by substituting each free occurrence of the variable x_i with the constant c_i, for each $i \in \{1, \ldots, n\}$.

are neither objective nor subjective. For example $\exists x.P(x)$ is an objective sentence, $\mathbf{K}(\exists x.P(x) \wedge \neg\mathbf{K}P(x))$ is a subjective sentence, while $\exists x.P(x) \wedge \neg\mathbf{K}P(x)$ is neither objective nor subjective.

In our setting, among the various epistemic interpretations, we are interested in specific ones that guarantee *minimal knowledge* over a DL knowledge base. Namely: let Σ be a DL knowledge base (TBox and ABox), and let $Mod(\Sigma)$ be the set of all FOL-interpretations that are models of Σ. Then a Σ-*EQL-interpretation* is an epistemic interpretation E, w where $E = Mod(\Sigma)$. We say that a sentence ϕ is Σ-*EQL-satisfiable* if there exists a Σ-*EQL-model* for ϕ, i.e., a Σ-*EQL*-interpretation E, w such that $E, w \models \phi$. Otherwise, we say that ϕ is Σ-*EQL-unsatisfiable*. Observe that for objective formulas this notion of satisfiability becomes the standard notion of FOL-satisfiability (relative to Σ). A sentence ϕ is *EQL-logically implied* by Σ, written $\Sigma \models_{EQL} \phi$, if every Σ-EQL-interpretation is a Σ-EQL-model of ϕ.

It is worth mentioning some characterizing properties of *EQL*.

Proposition 1 *For every DL knowledge base Σ and every EQL sentence ϕ we have:*

$$\begin{array}{l}
\Sigma \models_{EQL} \mathbf{K}\phi \supset \phi \\
\Sigma \models_{EQL} \mathbf{K}\phi \supset \mathbf{K}\mathbf{K}\phi \\
\Sigma \models_{EQL} \neg\mathbf{K}\phi \supset \mathbf{K}\neg\mathbf{K}\phi
\end{array}$$

These are the standard S5 axioms of modal logic. The first one expresses that "what is known is true" (knowledge is accurate), and the latter two express that the knowledge base has "complete knowledge on what is known and not known".

Proposition 2 *For every DL knowledge base Σ and every objective EQL sentence ϕ we have:*

$$\begin{array}{lcl}
\Sigma \models \phi & \textit{iff} & \Sigma \models_{EQL} \mathbf{K}\phi \\
\Sigma \not\models \phi & \textit{iff} & \Sigma \models_{EQL} \neg\mathbf{K}\phi
\end{array}$$

The above proposition relates knowledge to FOL logical implication. It says that if an objective sentence ϕ is logically implied then it is known, and vice-versa, that if ϕ is not logically implied then it is not known. Notably, the latter property is a consequence of the minimal knowledge semantics that we are adopting. Observe also that, as a consequence of this, every objective sentence is either known or not known by a DL knowledge base.

Proposition 3 *For every subjective EQL formula ϕ with free variables $x_1, \ldots, x_n$ there is another subjective EQL formula ϕ', with free variables $x_1, \ldots, x_n$, such that: (i) every occurrence of a subformula of the form $\mathbf{K}\psi$ in ϕ' is such that ψ is a non-subjective formula and $\mathbf{K}\psi$ occurs in ϕ; (ii) for every epistemic interpretation E, w, we have that $E, w \models \forall x_1, \ldots, x_n.\phi \equiv \phi'$.*

The above proposition says that we do not gain expressive power by putting in the scope of the $\mathbf{K}$ operator a formula that is already subjective. In other words, if we start from formulas of the form $\mathbf{K}\psi$, where ψ is not subjective, as the basic building blocks of the language, then applying the full *EQL* constructs actually gives the same expressive power as applying the first-order constructs only. By the way, to get the sentence ϕ' from ϕ we simply need to "push inward" the $\mathbf{K}$ operators through subjective subformulas, stopping when we get to subformulas that are not subjective, and simplifying $\mathbf{KK}\psi$ to $\mathbf{K}\psi$ whenever possible.

Finally we provide the definition of *EQL*-queries.

Definition 4 An *EQL-query* is an *EQL*-formula q with free variables $x_1, \ldots, x_n$, for $n \geq 0$, written $q[x_1, \ldots, x_n]$. ∎

Given a Σ-*EQL*-interpretation E, w, we say that an n-tuple $(c_1, \ldots, c_n)$ of constants in Δ *satisfies an EQL-query* $q[x_1, \ldots, x_n]$ *in* E, w, written $E, w \models q[c_1, \ldots, c_n]$, if $E, w \models q^{x_1,\ldots,x_n}_{c_1,\ldots,c_n}$. A tuple $(c_1, \ldots, c_n)$ of constants in Δ is a *certain answer* to q over Σ, denoted $\Sigma \models_{EQL} q[c_1, \ldots, c_n]$, if $E, w \models q[c_1, \ldots, c_n]$ for every Σ-EQL-interpretation E, w.

Given two *EQL*-queries $q[x_1, \ldots, x_n]$ and $q'[x_1, \ldots, x_n]$ we say that $q[x_1, \ldots, x_n]$ is *contained in* (resp., *equivalent to*) $q'[x_1, \ldots, x_n]$ if for every Σ-*EQL*-interpretation E, w and every n-tuple $(c_1, \ldots, c_n)$ of constants in Δ we have that $E, w \models q[c_1, \ldots, c_n]$ implies (resp., if and only if) $E, w \models q[c_1, \ldots, c_n]$.

Example 5 Consider the DL knowledge base Σ constituted by the following TBox $\mathcal{T}$ and ABox $\mathcal{A}$:

$$\begin{array}{rcl} \mathcal{T} & = & \{ \ \mathsf{Male} \sqsubseteq \neg\mathsf{Female} \ \} \\ \mathcal{A} & = & \{ \ \mathsf{Female}(\mathsf{mary}), \mathsf{Female}(\mathsf{ann}), \mathsf{Female}(\mathsf{jane}), \mathsf{Male}(\mathsf{bob}), \\ & & \ \ \mathsf{Male}(\mathsf{john}), \mathsf{Male}(\mathsf{paul}), \mathsf{PARENT}(\mathsf{bob}, \mathsf{mary}), \mathsf{PARENT}(\mathsf{bob}, \mathsf{ann}), \\ & & \ \ \mathsf{PARENT}(\mathsf{john}, \mathsf{paul}), \mathsf{PARENT}(\mathsf{mary}, \mathsf{jane}) \ \} \end{array}$$

Suppose we want to know the set of males that have only female children. This corresponds to the following first-order query q_1:

$$q_1[x] \ = \ \mathsf{Male}(x) \land \forall y.\mathsf{PARENT}(x, y) \rightarrow \mathsf{Female}(y)$$

It is easy to verify that the set of certain answers of q_1 over Σ is empty. In particular, bob is not a certain answer to the above query, since (due to the open-world semantics of DLs) there are models of Σ in which the interpretation of PARENT contains pairs of elements of the form (bob, x) and the interpretation of Male contains the element x.

Suppose now that we want to know who are the persons all of whose *known* children are female. This can be expressed by the following *EQL* query q_2:

$$q_2[x] \ = \ \mathsf{Male}(x) \land \forall y.(\mathbf{K}\mathsf{PARENT}(x, y)) \rightarrow \mathsf{Female}(y)$$

It is immediate to verify that the certain answers over Σ of the query q_2 are bob and paul. In fact, for each Σ-*EQL*-interpretation E, w (we recall that $E = Mod(\Sigma)$), $(\mathsf{bob}, \mathsf{mary})$ and $(\mathsf{bob}, \mathsf{ann})$ are the only pairs (x, y) such that $\Sigma \models_{EQL} \mathbf{K}\mathsf{PARENT}(x, y)$; moreover, paul is a certain answer because he is male and has no known children. Analogously, it can be seen that no other constant is in the set of certain answers of q_2 over Σ. ∎

Example 6 Suppose now that we want to know who are the single children according to what is known, i.e., the known children who have no known sibling. This can be expressed by the following *EQL* query q_3:

$$q_3[x] \;=\; \exists y.(\mathbf{K}\mathsf{PARENT}(y,x)) \land \forall z.(\mathbf{K}\mathsf{PARENT}(y,z)) \rightarrow z = x$$

It is immediate to verify that the certain answers over Σ of the query q_3 are paul and jane. ■

3 *EQL-Lite*($\mathcal{Q}$)

We introduce now the query language *EQL-Lite*($\mathcal{Q}$). Such a language is parameterized with respect to a *basic* query language $\mathcal{Q}$, which is a subset of *EQL*. Informally, *EQL-Lite*($\mathcal{Q}$) is the first-order query language with equality whose atoms are formulas of the form $\mathbf{K}q$ where q is a $\mathcal{Q}$-query, i.e., a query in $\mathcal{Q}$.

To define *EQL-Lite*($\mathcal{Q}$) formally, we first need to introduce the notion of domain independence for first-order queries, which is the semantical restriction on first-order logic that is needed to get the equivalence to relational algebra [1]. A first-order query q is *domain independent* if for each pair of FOL interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively over domains $\Delta_{\mathcal{I}_1} \subseteq \Delta$ and $\Delta_{\mathcal{I}_2} \subseteq \Delta$, for which $P^{\mathcal{I}_1} = P^{\mathcal{I}_2}$ for all atomic relations P, we have that $q^{\mathcal{I}_1} = q^{\mathcal{I}_2}$ [3].

Given a subset $\mathcal{Q}$ of *EQL*, we call *epistemic atom in* $\mathcal{Q}$ an expression of the form $\mathbf{K}q[x_1,\ldots,x_n]$, where $q[x_1,\ldots,x_n]$ is a $\mathcal{Q}$-query.

Definition 7 An *EQL-Lite($\mathcal{Q}$) query* is a formula ψ that:

- is constructed according to the following syntax:

$$\psi \;::=\; a \mid x_1 = x_2 \mid \psi_1 \land \psi_2 \mid \neg\psi \mid \exists x.\psi,$$

  where a is an epistemic atom in $\mathcal{Q}$, and

- is *domain-independent*, when we consider epistemic atoms as atomic formulas. ■

Observe that in *EQL-Lite*(Q) we do not allow for nesting of the $\mathbf{K}$ operator outside the expressions of the basic query language $\mathcal{Q}$. Indeed, we now show that allowing such nested occurrences of the epistemic operator does not actually increase the expressive power of *EQL-Lite*($\mathcal{Q}$).

Proposition 8 *Let EQL-Lite*($\mathcal{Q}$)$^+$ *be the extension of EQL-Lite($\mathcal{Q}$) obtained by adding to the abstract syntax for EQL-Lite($\mathcal{Q}$) formulas the rule*

$$\psi \;::=\; \mathbf{K}\psi$$

For each query $q \in$ EQL-Lite($\mathcal{Q}$)$^+$, there exists a query $q' \in$ EQL-Lite($\mathcal{Q}$) such that q is equivalent to q'.

[3]For an interpretation $\mathcal{I}$ over domain $\Delta_{\mathcal{I}}$ and a FOL query $q[x_1,\ldots,x_n]$, we use $q^{\mathcal{I}}$ to denote the result of the evaluation of q in $\mathcal{I}$, i.e., the set of tuples $(c_1,\ldots,c_n)$ of constants in $\Delta_{\mathcal{I}}$ such that $\phi^{x_1,\ldots,x_n}_{c_1,\ldots,c_n}$ is true in $\mathcal{I}$.

The above property is an immediate consequence of Proposition 3, since *EQL-Lite*($\mathcal{Q}$) queries are subjective *EQL* formulas, where the $\mathbf{K}$ operator is applied to non subjective (in fact objective) subformulas only, and hence each *EQL-Lite*($\mathcal{Q}$)$^+$ query can be reduced to an equivalent *EQL-Lite*($\mathcal{Q}$) query by pushing inward the $\mathbf{K}$ operator, stopping in front of the epistemic atoms, and simplifying $\mathbf{KK}\psi$ to $\mathbf{K}\psi$ whenever possible.

In spite of its expressive richness, *EQL-Lite*($\mathcal{Q}$) enjoys an interesting complexity characterization of query answering. In the rest of the paper, when we speak about the computational complexity of the query answering problem we actually refer to the computational complexity of the *recognition problem associated with query answering* [1]. Let $\mathcal{Q}$ be a query language and $\mathcal{L}$ a DL language, and let us assume that the query language $\mathcal{Q}$ over $\mathcal{L}$-knowledge bases has data complexity $C_{\mathcal{Q},\mathcal{L}}$, i.e., the complexity of answering queries in $\mathcal{Q}$ over $\mathcal{L}$-knowledge bases measured in the size of the data of the knowledge base is $C_{\mathcal{Q},\mathcal{L}}$.

Let us further consider the following restriction over queries and knowledge bases.

Definition 9 Given a knowledge base Σ in $\mathcal{L}$, a query q in $\mathcal{Q}$ is Σ-*range-restricted*, if the certain answers of q over Σ contain only elements of $adom(\Sigma)$, where $adom(\Sigma)$ denotes the set of constants occurring in Σ. By extension, an *EQL-Lite*($\mathcal{Q}$) query is Σ-range-restricted if each of its epistemic atoms involves a Σ-range-restricted query. ■

The class of Σ-range-restricted queries is perfectly natural in this setting: indeed, it can be shown that if the set of certain answers of a query q on a knowledge base Σ contains elements that are not in $adom(\Sigma)$, then the set of certain answers is infinite.

Example 10 The following in an *EQL-Lite*($\mathcal{Q}$) query, where $\mathcal{Q}$ is the language of atomic queries:

$$q_4[x] \;=\; (\mathbf{K}\mathsf{Male}(x)) \wedge \forall y.(\mathbf{K}\mathsf{PARENT}(x,y)) \rightarrow (\mathbf{K}\mathsf{Female}(y))$$

It is easy to verify that for the knowledge base Σ given in Example 5, q_4 is Σ-range-restricted. ■

Observe that, by Proposition 2, we have complete information on each instantiation of the epistemic atoms of an *EQL-Lite*($\mathcal{Q}$) query, i.e., either the instantiated epistemic atom is entailed by Σ or its negation is entailed by Σ. Now, answering a Σ-range-restricted *EQL-Lite*($\mathcal{Q}$) query amounts to evaluating a domain independent first-order query whose variables range over $adom(\Sigma)$ and whose instantiated epistemic atoms $\mathbf{K}q[c_1,\ldots,c_n]$ can be checked by verifying whether $(c_1,\ldots,c_n)$ is a certain answer of q over the knowledge base. We know that evaluating a first-order query over a given database is in LogSpace in data complexity [1], and, by our assumption, computing whether a tuple of elements of $adom(\Sigma)$ is in the relation corresponding to the extension of an epistemic atom, can be done in $C_{\mathcal{Q},\mathcal{L}}$ in data complexity. Hence, we immediately derive the following result on the data complexity of answering Σ-range-restricted *EQL-Lite*($\mathcal{Q}$)-queries, where we denote with $C_1^{C_2}$ the class of languages recognized by a C_1-Turing Machine that uses an oracle in C_2.

Theorem 11 *Let $\mathcal{Q}$ be a query language over $\mathcal{L}$-knowledge bases that is in $C_{\mathcal{Q},\mathcal{L}}$ with respect to data complexity. Let Σ be an $\mathcal{L}$-knowledge base, and q a Σ-range-restricted EQL-Lite($\mathcal{Q}$) query. Then, answering q over Σ is in* $\textsc{LogSpace}^{C_{\mathcal{Q},\mathcal{L}}}$ *with respect to data complexity.*

Among the various choices of the basic query language $\mathcal{Q}$ in *EQL-Lite*($\mathcal{Q}$), a prominent role is played by *unions of conjunctive queries (UCQs)*. In fact, the language of UCQs is currently the most expressive subset of first-order logic for which query answering over DL knowledge bases is known to be decidable [5, 15]. Consequently, in the following we will focus on *EQL-Lite*(UCQ), and will call such a language simply *EQL-Lite*.

4 Answering *EQL-Lite* queries over $\mathcal{ALCQI}$ knowledge bases

As a consequence of the properties shown in the previous section, we now provide a computational characterization of answering Σ-range-restricted *EQL-Lite* queries in $\mathcal{ALCQI}$. It is known that answering unions of conjunctive queries over $\mathcal{ALCQI}$ knowledge bases is coNP-complete with respect to data complexity [15]. Based on this characterization and on Theorem 11, we are able to show the following result.

Theorem 12 *Let Σ be an $\mathcal{ALCQI}$-knowledge base, and q a Σ-range-restricted EQL-Lite query. Then, answering q over Σ is in Θ_2^p with respect to data complexity.*

We recall that $\Theta_2^p = \Delta_2^p[O(\log n)] = P^{NP[O(\log n)]}$ [10, 8], i.e., Θ_2^p is the class of the decision problems that can be solved in polynomial time through a logarithmic number of calls to an NP-oracle. Such a class is considered as "mildly" harder than the class NP, since a problem in Θ_2^p can be solved by solving "few" (i.e., a logarithmic number of) instances of problems in NP. Consequently, answering *EQL-Lite* queries in $\mathcal{ALCQI}$ (and in all the DLs in which answering UCQs is a coNP-complete problem) is "mildly harder" than answering UCQs.

5 Answering *EQL-Lite* queries over *DL-Lite* knowledge bases

In this section we study *EQL-Lite* queries posed over *DL-Lite* knowledge bases. *DL-Lite* [6, 3] is a DL specifically tailored to capture basic ontology languages, while keeping low complexity of reasoning, in particular, polynomial in the size of the instances in the knowledge base. Answering UCQs in *DL-Lite* is in LogSpace with respect to data complexity[4]. Moreover all UCQs in *DL-Lite* are Σ-range-restricted. As a consequence of Theorem 11 we get that moving from UCQs to *EQL-Lite* does not blow up computational complexity of the query answering problem.

[4] It is easy to see that all results for CQs in [6, 3] can be extended to UCQs.

Theorem 13 *Answering EQL-Lite queries over DL-Lite knowledge bases is in* LogSpace *with respect to data complexity.*

We point out that membership in LogSpace for the problem of answering UCQs over *DL-Lite* knowledge bases follows from a notable property of such a DL language, namely, *FOL-reducibility* of query answering [4]. Intuitively, FOL-reducibility means that query answering can be reduced to evaluating queries over the database corresponding to the ABox of a DL knowledge base, which therefore can be maintained in secondary storage. More formally, given an ABox $\mathcal{A}$ involving membership assertions on atomic concept and roles only, we define $\mathcal{I}_\mathcal{A}$ as the interpretation constructed as follows:

- $a^{\mathcal{I}_\mathcal{A}} = a$ for each constant a,
- $A^{\mathcal{I}_\mathcal{A}} = \{a \mid A(a) \in \mathcal{A}\}$ for each atomic concept A, and
- $P^{\mathcal{I}_\mathcal{A}} = \{(a_1, a_2) \mid P(a_1, a_2) \in \mathcal{A}\}$ for each atomic role P.

Then, query answering in a DL $\mathcal{L}$ is *FOL-reducible* if for every query q (of a given language) and every TBox $\mathcal{T}$ expressed in $\mathcal{L}$, there exists a FOL query q_1 such that for every ABox $\mathcal{A}$, we have that $(\mathcal{T}, \mathcal{A}) \models_{EQL} q[c_1, \ldots, c_n]$ if and only if $(c_1, \ldots, c_n)^{\mathcal{I}_\mathcal{A}} \in q_1^{\mathcal{I}_\mathcal{A}}$. In other words, q_1 is evaluated over the ABox $\mathcal{A}$ considered as a database. Observe that FOL-reducibility is a very nice property from a practical point of view. Indeed, in all such cases in which query answering can be reduced to evaluation of a suitable *domain independent* FOL query q_1, then q_1 can be expressed in relational algebra, i.e., in SQL. Therefore, query answering can take advantage of optimization strategies provided by current DBMSs (which are in charge of properly managing ABoxes in secondary storage).

Now, it turns out that FOL-reducibility is also at the basis of the membership in LogSpace of answering *EQL-Lite* queries over *DL-Lite* knowledge bases, as the following theorem shows.

Theorem 14 *Answering EQL-Lite queries in DL-Lite is FOL-reducible. Furthermore, the resulting FOL-queries are domain independent.*

Proof (sketch). We make use of the algorithm for FOL-reducibility of UCQs over *DL-Lite* knowledge bases presented in [3]. More precisely, given a *DL-Lite* TBox $\mathcal{T}$ and an *EQL-Lite* query ψ over $\mathcal{T}$, we execute the algorithm of [3] for each epistemic atom $\mathbf{K}q[x_1, \ldots, x_n]$ of ψ, giving as inputs to each such execution the union of conjunctive queries $q[x_1, \ldots, x_n]$ and the *DL-Lite* TBox $\mathcal{T}$. Then, we substitute to each epistemic atom $\mathbf{K}q[x_1, \ldots, x_n]$ of ψ the union of conjunctive queries $q'[x_1, \ldots, x_n]$ produced by the corresponding execution of the rewriting algorithm, thus obtaining (provided some further syntactic transformations) a FOL query q_1. Now, it is possible to show that q_1 is domain independent, and that for every ABox $\mathcal{A}$, $(\mathcal{T}, \mathcal{A}) \models_{EQL} q[c_1, \ldots, c_n]$ iff $(c_1, \ldots, c_n)^{\mathcal{I}_\mathcal{A}} \in q_1^{\mathcal{I}_\mathcal{A}}$, thus proving the claim. □

As a consequence, to perform query answering of *EQL-Lite* queries in *DL-Lite*, we can rely on traditional relational DBMSs.

Recently, different versions of *DL-Lite* have been considered, and an entire family of "lite" DLs, namely, the *DL-Lite family*, has been defined [4]. Roughly speaking, DLs of such a family differ one another for the set of constructs allowed in the right-hand side and in the left-hand side of inclusion assertions between concepts and/or roles specified in the TBox (e.g., allowing in certain cases for the presence of existential qualified quantification on the right-hand side, or conjunctions of concepts in the left-hand side), as well as the possibility of specifying functionality assertions on roles, inclusion assertions between roles, and n-any relationships in addition to binary roles. Notably, the DLs of the *DL-Lite* family are the maximal logics allowing for FOL-reducibility of answering unions of conjunctive queries [4]. As for answering *EQL-Lite* queries, we point out that Theorem 14 also holds for all the DLs belonging to the *DL-Lite* family.

6 Conclusions

Motivated by various needs related to querying DL knowledge bases, we have proposed the query language *EQL*, based on a variant of the well-known first-order modal logic of knowledge/belief. Then, we have studied a subset of this language, called *EQL-Lite*($\mathcal{Q}$), arguing that it allows for formulating queries that are interesting both from the expressive power point of view, and from the computational complexity perspective. Finally, we have investigated the properties of *EQL-Lite*($\mathcal{Q}$) for the cases of $\mathcal{ALCQI}$ and *DL-Lite* knowledge bases, under the assumption that $\mathcal{Q}$ is the language of unions of conjuntive queries. In particular, we have shown that answering *EQL-Lite*($\mathcal{Q}$) in the latter setting is LogSpace in data complexity, and, notably, can be done through standard database technologies.

We are working specifically on *EQL-Lite*($\mathcal{Q}$) for *DL-Lite* knowledge bases, for the case where $\mathcal{Q}$ is the query language whose queries are either a UCQ or a comparison atom involving values taken from a set of given domains. We are currently implementing such an extended language with the goal of enhancing the querying capabilities of the QuOnto system [2].

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison Wesley Publ. Co., 1995.

[2] A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QuOnto: QUerying ONTOlogies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 1670–1671, 2005.

[3] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int.*

Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006), 2006.

[5] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[6] D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, and G. Vetere. DL-Lite: Practical reasoning for rich DLs. In *Proc. of the 2004 Description Logic Workshop (DL 2004)*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-104/`, 2004.

[7] B. F. Chellas. *Modal Logic: An introduction.* Cambridge University Press, 1980.

[8] T. Eiter and G. Gottlob. The complexity class Θ_2^p: Recent results and applications in AI and modal logic. In *Proc. of the 11th Int. Symp. on Fundamentals of Computation Theory (FCT'97)*, volume 1279 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer, 1997.

[9] R. Fikes, P. Hayes, and I. Horrocks. OWL-QL: A language for deductive query answering on the semantic web. *J. of Web Semantics*, 2(1), 2005.

[10] G. Gottlob. NP trees and Carnap's modal logic. *J. of the ACM*, 42(2):421–457, 1995.

[11] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen, London (United Kingdom), 1984.

[12] H. J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23:155–212, 1984.

[13] H. J. Levesque and G. Lakemeyer. *The Logic of Knowledge Bases.* The MIT Press, 2001.

[14] M. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006)*, 2006.

[15] M. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of answering unions of conjunctive queries in $\mathcal{SHIQ}$. In *Proc. of the 2006 Description Logic Workshop (DL 2006)*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/`, 2006.

[16] R. Reiter. What should a database know? *J. of Logic Programming*, 14:127–153, 1990.

[17] R. Reiter. Narratives as programs. In *Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 99–108, 2000.