EQL-Lite: Effective First-Order Query Processing in Description Logics*

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

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

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

Abstract

Querying Description Logic knowledge bases has received great attention in the last years. In such a problem, 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 (FOL); 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 a nonmonotonic epistemic FOL query language as a means for expressing sophisticated queries over Description Logic knowledge bases. We show that through a controlled use of the epistemic operator, resulting in the language called *EQL-Lite*, we are able to formulate full FOL 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 FOL reducible (i.e., compilable 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, 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 (UCQ) [7; 16]. 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, while being essential for representing incomplete information, 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 children 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 the 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, tractable).

This paper presents the following contributions. We define a 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 [14; 17; 15]. The language incorporates a minimal knowledge operator $\mathbf{K}$, which 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 controlled use of the operator $\mathbf{K}$, resulting in the language called *EQL-*

*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).

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

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 $\mathcal{Q}$, called *embedded query language*, 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 queries expressed in $\mathcal{Q}$.

We investigate the properties of *EQL-Lite*($\mathcal{Q}$) for several interesting cases, characterizing the data complexity of query answering (see Section 4). In particular, we consider the following cases: $\mathcal{SHIQ}$ with simple concept and role expressions as embedded queries, $\mathcal{DLR}$ with embedded unions of conjunctive queries, PTIME-complete DLs such as *Horn-*$\mathcal{SHIQ}$ or $\mathcal{EL}$, with simple concept and role expressions as embedded queries, basic DLs such as $\mathcal{ALC}$ with epistemic concepts as embedded queries, and highly tractable DLs, such as those of the *DL-Lite* family, with embedded unions of conjunctive queries. For the latter case, we show that answering *EQL-Lite*(UCQ) is in LOGSPACE, and, notably, can be reduced to 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.

Finally, we briefly discuss (see Section 5) the use of *EQL-Lite*($\mathcal{Q}$) for introducing the notion of integrity constraints in DL KBs.

2 Epistemic query language

In this paper we consider queries over a Description Logic (DL) knowledge base [4]. We don't focus on any particular DL. We simply assume that, through the DL, we are able to express our knowledge in terms of *atomic concepts*, i.e., unary predicates, and *atomic roles/relations*, i.e., binary/n-ary predicates. General concepts and roles/relations are built through the constructs allowed in the DL, and we assume that such constructs are expressible in FOL. As usual, a DL *knowledge base (KB)* is formed by a set of *assertions*, typically divided into a *TBox*, expressing intensional knowledge, and an *ABox*, expressing extensional knowledge. We assume again that such assertions can be expressed as FOL sentences (i.e., closed FOL formulas). In other words, DL KBs can be seen as FOL theories (of specific forms). Observe that most DLs fulfill such assumptions: the only notable exceptions are those that include some form of second-order constructs such as transitive closure or fixpoints [4].

As usual, when talking about query answering, w.l.o.g., we interpret DL KBs on interpretations sharing the *same infinite countable domain* Δ, and we assume that our language includes an infinitely countable set of disjoint constants corresponding to elements of Δ, also known as *standard names* [15]. This allows us to blur the distinction between such constants (which are syntactic objects) and the elements of Δ that they denote (which are semantical objects).

As a query language, we make use of a variant of the well-known first-order modal logic of knowledge/belief [14; 17; 15; 12], here called *EQL*. The language *EQL* is a first-order modal language with equality and with a single modal operator $\mathbf{K}$, constructed from concepts (i.e., unary predicates) and roles/relations (i.e., binary/n-ary predicates) and the constants introduced above (i.e., the standard names corresponding to Δ). In *EQL*, the modal operator is used to formalize the epistemic state of the DL KB, according to the minimal knowledge semantics (see later). Informally, the formula $\mathbf{K}\phi$ should be read as "ϕ is known to hold (by the KB)".

In the following, we use c to denote a constant, $\vec{c}$ to denote a tuple of constants, x to denote a variable, $\vec{x}$ to denote a tuple of variables, and ϕ, ψ to denote arbitrary formulas, and ψ_c^x to denote a formula where each x is replaced by c.

A *world* is a FOL 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:

$E, w \models c_1 = c_2$	iff	$c_1 = c_2$
$E, w \models P(\vec{c})$	iff	$w \models P(\vec{c})$
$E, w \models \phi_1 \wedge \phi_2$	iff	$E, w \models \phi_1$ and $E, w \models \phi_2$
$E, w \models \neg\phi$	iff	$E, w \not\models \phi$
$E, w \models \exists x.\psi$	iff	$E, w \models \psi_c^x$ for some constant c
$E, w \models \mathbf{K}\psi$	iff	$E, w' \models \psi$ for every $w' \in E$

Formulas without occurrences of $\mathbf{K}$ are said to be *objective*, since they talk about what is true. Observe that to check whether $E, w \models \phi$, 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 KB are indeed objective sentences. Instead, formulas where each occurrence of predicates and of the equality is in the scope of the $\mathbf{K}$ 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 whether $E, w \models \phi$ we do not have to look at w but only at E. We use such formulas to query what the KB knows. In other words, through subjective sentences we do not query information about the world represented by the KB; instead, we query the epistemic state of the KB itself. Obviously there are formulas that 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 represent the *minimal epistemic state* of the DL KB, i.e., the state in which the KB has minimal knowledge. Namely: let Σ be a DL KB (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 for which $E = Mod(\Sigma)$. A sentence ϕ is *EQL-logically implied* by Σ, written $\Sigma \models_{EQL} \phi$, if for every Σ-EQL-interpretation E, w we have $E, w \models \phi$. Observe that for objective formulas such a definition becomes the standard one, namely $w \models \phi$ for all $w \in Mod(\Sigma)$, denoted by $\Sigma \models \phi$.

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

Proposition 1 *For every DL KB* Σ *and every EQL-sentence*

ϕ *we have:*

$$\Sigma \models_{EQL} \mathbf{K}\phi \supset \phi$$
$$\Sigma \models_{EQL} \mathbf{K}\phi \supset \mathbf{K}\mathbf{K}\phi$$
$$\Sigma \models_{EQL} \neg\mathbf{K}\phi \supset \mathbf{K}\neg\mathbf{K}\phi$$

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 KB has "complete knowledge on what is known and not known".

Proposition 2 *For every DL KB* Σ *and every EQL-sentence* ϕ *we have that:*

$$\Sigma \models_{EQL} \mathbf{K}\phi \quad \textit{or} \quad \Sigma \models_{EQL} \neg\mathbf{K}\phi$$

The above proposition tells us that for any sentence ϕ the KB logically implies either that the sentence is known or that the sentence is not known, i.e., we have complete information on what the KB knows. Notably, this is a consequence of the minimal knowledge semantics that we are adopting.

Proposition 3 *For every DL KB* Σ *and every objective EQL-sentence* ϕ *we have:*

$$\Sigma \models \phi \quad \textit{iff} \quad \Sigma \models_{EQL} \mathbf{K}\phi$$
$$\Sigma \not\models \phi \quad \textit{iff} \quad \Sigma \models_{EQL} \neg\mathbf{K}\phi$$

The above proposition relates knowledge to FOL logical implication, and is again a consequence of the minimal knowledge semantics. It allows us to give a very concrete interpretation to knowledge for objective sentences: ϕ is known iff it is logically implied, otherwise it is not known.

We are now ready to define *EQL*-queries: An *EQL-query* is simply an *EQL*-formula, possibly an open one.

Let q be an *EQL*-query with free variables $\vec{x}$, where the arity of $\vec{x}$ is $n \geq 0$, and is called the arity of q. We sometimes use the notation $q[\vec{x}]$ to make the free variables $\vec{x}$ explicit. Also we use the notation $q[\vec{c}]$ to denote $q^{\vec{x}}_{\vec{c}}$ (i.e., the formula obtained from q by substituting each free occurrence of the variable x_i in $\vec{x}$ with the constant c_i in $\vec{c}$, where obviously $\vec{x}$ and $\vec{c}$ must have the same arity). Since we are dealing with all the models of the KB, as usual, query answering should return those tuples of constants that make the query true in every model of the KB: the so-called certain answers. Formally, the *certain answers* to a query $q[\vec{x}]$ over a KB Σ are the set

$$ans(q, \Sigma) = \{\vec{c} \in \Delta \times \cdots \times \Delta \mid \Sigma \models_{EQL} q[\vec{c}]\}$$

Example 4 Consider the DL KB 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 do not have female children. This corresponds to the following FOL query q_1:

$$q_1[x] \ = \ \mathsf{Male}(x) \wedge \neg\exists y.\mathsf{PARENT}(x, y) \wedge \mathsf{Female}(y)$$

It is easy to verify that the set of certain answers to q_1 over Σ is empty. In particular, neither john nor paul are certain answers 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 (john, x) or (paul, x) and the interpretation of Female contains the element x.

Suppose now that we want to know who are the *known* males that are not *known* to be parents of a female. This can be expressed by the following *EQL*-query q_2:

$$q_2[x] \ = \ \mathbf{K}\mathsf{Male}(x) \wedge \neg\mathbf{K}(\exists y.\mathsf{PARENT}(x, y) \wedge \mathsf{Female}(y))$$

It is immediate to verify that the certain answers to q_2 over Σ are john and paul, since they are the only known males that are not in the answer to the query $\exists y.\mathsf{PARENT}(x, y) \wedge \mathsf{Female}(y)$.

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:

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

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

Notice that, in an *EQL*-query, we can apply a form of closed world reasoning: for example, in query q_2 above, the evaluation of $\neg\mathbf{K}(\exists y.\mathsf{PARENT}(x, y) \wedge \mathsf{Female}(y))$ corresponds to the evaluation of $\neg\exists y.\mathsf{PARENT}(x, y) \wedge \mathsf{Female}(y)$ under the closed world assumption.

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

We introduce now the query language *EQL-Lite*($\mathcal{Q}$). Such a language is a particularly well-behaved fragment of *EQL*, and is parameterized with respect to an *embedded query language* $\mathcal{Q}$, which again is a subset of *EQL*. Informally, *EQL-Lite*($\mathcal{Q}$) is the FOL query language with equality whose atoms are epistemic formulas of the form $\mathbf{K}\varrho$ where ϱ is a query of $\mathcal{Q}$. Formally, *an EQL-Lite*($\mathcal{Q}$) query is a possibly open *EQL*-formula built according to the following syntax:

$$\psi \ ::= \ \mathbf{K}\varrho \mid x_1 = x_2 \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \exists x.\psi,$$

where ϱ is a query in the embedded query language $\mathcal{Q}$. We call *epistemic atoms* the formulas $\mathbf{K}\varrho$ occurring in an *EQL-Lite*($\mathcal{Q}$) query.

Observe that in *EQL-Lite*($\mathcal{Q}$) we do not allow the $\mathbf{K}$ operator to occur outside of the epistemic atoms $\mathbf{K}\varrho$. Indeed, allowing for occurrences of the $\mathbf{K}$ outside such atoms does not actually increase the expressive power of *EQL-Lite*($\mathcal{Q}$), as the following proposition shows.

Proposition 5 *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* $\psi \ ::= \ \mathbf{K}\psi$. *Then, for each query* $q \in$ *EQL-Lite*$(\mathcal{Q})^+$, *there exists a query* $q' \in$ *EQL-Lite*$(\mathcal{Q})$ *such that* $E, w \models \forall\vec{x}.q[\vec{x}] \equiv q'[\vec{x}]$, *for every epistemic interpretation* E, w.

In fact, an *EQL-Lite*$(\mathcal{Q})^+$ query q can be reduced to an equivalent *EQL-Lite*($\mathcal{Q}$) query q' in linear time by simply pushing inward the $\mathbf{K}$ operator, stopping in front of the epistemic atoms, and simplifying $\mathbf{K}\mathbf{K}\psi$ to $\mathbf{K}\psi$ and $\mathbf{K}\neg\mathbf{K}\psi$ to $\neg\mathbf{K}\psi$ whenever possible.

EQL-Lite($\mathcal{Q}$) queries enjoy a very interesting computational property: one can decouple the reasoning needed for answering the epistemic atoms from the reasoning needed for answering the whole query. Formally, let Σ be a DL KB, and $q[\vec{x}]$ be an *EQL-Lite*($\mathcal{Q}$) query over Σ, whose epistemic atoms are $\mathbf{K}\varrho_1, \ldots, \mathbf{K}\varrho_m$. We denote by $q_{FOL}[\vec{x}]$ the FOL query obtained from q by replacing each epistemic atom $\mathbf{K}\varrho_i$ by a new predicate $R_{\mathbf{K}\varrho_i}$ whose arity is the number of free variables in ϱ_i. Also we denote by $\mathcal{I}_{q,\Sigma}$ the FOL interpretation for the predicates $R_{\mathbf{K}\varrho_i}$ defined as follows: (*i*) the interpretation domain is $\Delta^{\mathcal{I}_{q,\Sigma}} = \Delta$; (*ii*) the extension of the predicates $R_{\mathbf{K}\varrho_i}$ is $R^{\mathcal{I}_{q,\Sigma}}_{\mathbf{K}\varrho_i} = ans(\varrho_i, \Sigma)$. Finally, we denote by $eval(q_{FOL}[\vec{x}], \mathcal{I}_{q,\Sigma}) = \{\vec{c} \in \Delta \times \cdots \times \Delta \mid \mathcal{I}_{q,\Sigma} \models q_{FOL}[\vec{c}]\}$ the result of evaluating q_{FOL} over $\mathcal{I}_{q,\Sigma}$.

Theorem 6 *Let Σ be a DL KB, q an EQL-Lite($\mathcal{Q}$) query over Σ, and q_{FOL} and $\mathcal{I}_{q,\Sigma}$ the FOL query and the FOL interpretation defined above. Then $ans(q, \Sigma) = eval(q_{FOL}, \mathcal{I}_{q,\Sigma})$.*

The theorem above tells us that, in order to compute the certain answers of an *EQL-Lite*($\mathcal{Q}$) query q, we can compute the certain answers of queries ϱ_i of the embedded query language $\mathcal{Q}$ occurring in the epistemic atoms of q, and then evaluate the query q as a FOL query, where we consider such certain answers as the extensions of the epistemic atoms.

The theorem above suggests a procedure to compute certain answers in *EQL-Lite*($\mathcal{Q}$). However, for such a procedure to be effective, we need to address two issues: (*i*) the extension of the predicates $R_{\mathbf{K}\varrho_i}$ in the FOL interpretation $\mathcal{I}_{q,\Sigma}$ needs to be finite, otherwise $\mathcal{I}_{q,\Sigma}$ would be infinite and the evaluation of q_{FOL} impossible in practice; (*ii*) since Δ itself is infinite, the evaluation of q_{FOL} must not directly deal with Δ.

We start by looking at the second issue first. Such an issue has a long tradition in relational databases where indeed one allows only for FOL queries that are "domain independent" [1]. In our context, a FOL 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 $R^{\mathcal{I}_1}_{\mathbf{K}\varrho_i} = R^{\mathcal{I}_2}_{\mathbf{K}\varrho_i}$ for all atomic relations $R_{\mathbf{K}\varrho_i}$, we have that $eval(q, \mathcal{I}_1) = eval(q, \mathcal{I}_2)$. We say that an *EQL-Lite*($\mathcal{Q}$) query q is *domain independent* if its corresponding query q_{FOL} is so. Domain independent FOL queries correspond to relational algebra queries (i.e., SQL queries) and several syntactic sufficient conditions have been devised in order to guarantee domain independence, see e.g., [1]. Such syntactic conditions can be directly translated into syntactic conditions on *EQL-Lite*($\mathcal{Q}$) queries.

As for the other issue, let Σ be a DL KB and ϱ a query of the embedded query language $\mathcal{Q}$. We say that ϱ is Σ-*range-restricted* if $ans(\varrho, \Sigma)$ is a *finite* set of tuples. By extension, an *EQL-Lite*($\mathcal{Q}$) query is Σ-range-restricted if each of its epistemic atoms involves a Σ-range-restricted query. In fact, the following proposition holds.

Proposition 7 *Let Σ be a DL KB and ϱ a Σ-*range-restricted *query in the embedded query language $\mathcal{Q}$. Then $ans(\varrho, \Sigma) \subseteq adom(\Sigma) \times \cdots \times adom(\Sigma)$. where $adom(\Sigma)$ is the set of all constants explicitly appearing in Σ.*

Indeed, if a constant not appearing in Σ occurs in $ans(\varrho, \Sigma)$, then one can substitute such a constant with any other constant not appearing in Σ, still getting a tuple in $ans(\varrho, \Sigma)$. Thus, $ans(\varrho, \Sigma)$ would be infinite.

Obviously it is of interest finding simple syntactic conditions that guarantee range-restrictedness. Here we give just a trivial one, which is however quite practical in several cases. Assume that we can introduce a new concept Adom in the DL KB Σ and assert for each constant c occurring in Σ the ABox assertion $\mathsf{Adom}(c)$. Moreover, let's require that all queries in $\mathcal{Q}$ must be of the form $\mathsf{Adom}(\vec{x}) \wedge \varrho[\vec{x}]$, where $\vec{x} = (x_1, \ldots, x_n)$ and $\mathsf{Adom}(\vec{x}) = \mathsf{Adom}(x_1) \wedge \cdots \wedge \mathsf{Adom}(x_n)$. Then trivially all queries in $\mathcal{Q}$ are Σ-range-restricted.

Now, if we consider *EQL-Lite*($\mathcal{Q}$) queries that are both domain independent and Σ-range-restricted, then we can effectively use the theorem above to compute the certain answers. Moreover we can give a computational complexity characterization of query answering for *EQL-Lite*($\mathcal{Q}$) queries.

Let $\mathcal{Q}$ be an embedded query language, $\mathcal{DL}$ a DL, and $C_{\mathcal{Q},\mathcal{DL}}$ the data complexity (i.e., the complexity measured in the size of the ABox only) of query answering for Σ-range-restricted queries of $\mathcal{Q}$ over KBs Σ expressed in $\mathcal{DL}$.[2] We know that evaluating a domain independent FOL query over a given FOL interpretation is LOGSPACE in data complexity [1], and, by our assumptions, computing whether a tuple of constants is in the relation corresponding to the extension of an epistemic atom, can be done in $C_{\mathcal{Q},\mathcal{DL}}$ in data complexity. Hence, we immediately derive the following result on the data complexity of answering domain independent and 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 8 *Let Σ be a KB expressed in the DL $\mathcal{DL}$, and q a domain independent and Σ-range-restricted EQL-Lite($\mathcal{Q}$) query over Σ. Then, answering q over Σ is in* LOGSPACE$^{C_{\mathcal{Q},\mathcal{DL}}}$ *with respect to data complexity, where $C_{\mathcal{Q},\mathcal{DL}}$ is the data complexity of answering Σ-range-restricted queries of $\mathcal{Q}$ over KBs Σ expressed in $\mathcal{DL}$.*

Example 9 Queries q_2 and q_3 in Example 4 are *EQL-Lite*($\mathcal{Q}$) queries, where $\mathcal{Q}$ is the language of conjunctive queries (in fact, for q_3, $\mathcal{Q}$ is the language of atomic queries). It is easy to verify that both such queries are domain independent, and that both are Σ-range-restricted for the KB Σ given in Example 4. In fact q_2 and q_3 are Σ-range-restricted for KBs Σ expressed (in practice) in any standard DL (indeed, the set $ans(\mathsf{PARENT}(x, y), \Sigma)$ may never be infinite). ∎

4 Case studies

We discuss now several notable applications of the above results on *EQL-Lite*($\mathcal{Q}$) for specific combinations of the DL used to express the KB and of the embedded query language $\mathcal{Q}$. Below, we implicitly refer to domain independent *EQL-Lite*($\mathcal{Q}$) queries only.

[2] As usual, when we speak about complexity of query answering, we actually mean the complexity of the associated *recognition problem*: i.e., checking whether a tuple of constants is in the answer to a query [1].

$\mathcal{SHIQ}$ KBs and queries with embedded concept and role expressions. We consider the case in which KBs are specified in the expressive DL $\mathcal{SHIQ}$ [11] (or equivalently, $\mathcal{DLR}$ [7]), and the embedded query language $\mathcal{Q}$ is that of Σ-range restricted $\mathcal{SHIQ}$ concept and role expressions. Note that this case is very significant in practice since it captures in particular the form of queries supported by the Racer and Pellet systems [10; 18]. Indeed, such queries are conjunctive queries over $\mathcal{SHIQ}$ concept and role expressions, in which, however, the existential quantification ranges over the named individuals in the ABox only. Now it turns out that such queries correspond to conjunctions of Σ-range-restricted $\mathcal{SHIQ}$ concept and role expressions prefixed by the **K** operator. Since instance checking in $\mathcal{SHIQ}$ is coNP-complete with respect to data complexity [13], by Theorem 8 we get that in this case answering such queries, as well as every *EQL-Lite*($\mathcal{Q}$) query, is in LOGSPACE^{NP} with respect to data complexity.

$\mathcal{DLR}$ KBs and queries with embedded unions of conjunctive queries. We consider the case in which KBs are again specified in an expressive DL like $\mathcal{DLR}$ or $\mathcal{ALCQI}$, and the embedded query language is that of Σ-range-restricted *unions of conjunctive queries* (UCQs), i.e., we consider queries expressed in *EQL-Lite*(UCQ). Notice that the language of UCQs is currently the most expressive subset of FOL for which query answering over KBs in an expressive DL, such as $\mathcal{DLR}$, is known to be decidable [7], and in fact coNP-complete with respect to data complexity [16][3]. From this characterization, and by applying again Theorem 8, we get that answering Σ-range-restricted *EQL-Lite*(UCQ) queries over $\mathcal{DLR}$ KBs is again in LOGSPACE^{NP} with respect to data complexity.

PTIME-complete DLs for KBs and queries. Next we consider the case in which KBs are specified in a PTIME-complete DL, such as Horn-$\mathcal{SHIQ}$ [13] or $\mathcal{EL}$ [3], and the embedded query language $\mathcal{Q}$ is that of Σ-range-restricted Horn-$\mathcal{SHIQ}$ (resp., $\mathcal{EL}$) concepts and role expressions. In this case, from Theorem 8, we get that answering Σ-range-restricted *EQL-Lite*($\mathcal{Q}$) queries is in PTIME (and, in fact PTIME-complete) with respect to data complexity.

Epistemic embedded query languages. In all the cases above the embedded query language consists of objective formulas. However, this does not need to be the case in general. Let us, for example, consider KBs consisting of ABoxes expressed in the basic DL $\mathcal{ALC}$ and embedded queries consisting of concepts and roles expressed in $\mathcal{ALCK}$ [8], i.e., $\mathcal{ALC}$ extended with the **K** operator. Then, since instance checking in $\mathcal{ALCK}$ concept and roles in $\mathcal{ALC}$ ABoxes can be done in PSPACE [8], by Theorem 8 we get that, in this case, answering *EQL-Lite*($\mathcal{Q}$) queries is in PSPACE as well.

[3]We have considered here $\mathcal{DLR}$ and $\mathcal{ALCQI}$ rather than $\mathcal{SHIQ}$, since the data complexity for answering UCQs containing transitive roles is still open.

***DL-Lite* KBs and queries with embedded unions of conjunctive queries.** Finally, we study the case in which KBs are specified using DLs of the *DL-Lite* family [5; 6] and the embedded query language $\mathcal{Q}$ for *EQL-Lite*($\mathcal{Q}$) queries is again that of UCQs.

The *DL-Lite* family [5; 6] is a family of DLs specifically tailored to deal with large amounts of data (i.e., ABoxes). While the expressive power of the DLs in the *DL-Lite* family is carefully controlled to admit tractable query answering, such DLs are expressive enough to capture the main notions (though not all, obviously) of both ontologies, and of conceptual modeling formalisms used in databases and software engineering (i.e., ER and UML class diagrams). Below, for simplicity, we denote by *DL-Lite* any DL that is a member of the *DL-Lite* family.

Answering UCQs in *DL-Lite* is in LOGSPACE with respect to data complexity[4]. Moreover, as a result of the tightly controlled expressive capabilities, we get the following.

Proposition 10 *Let Σ be a DL-Lite KB, and let ϱ be a UCQ over Σ. Then, ϱ is Σ-range-restricted.*

As a consequence of Theorem 8 and of membership in LOGSPACE of the problem of answering UCQs over *DL-Lite* KBs, we get that moving from UCQs to *EQL-Lite*(UCQ) does not change the data complexity of the query answering problem.

Theorem 11 *Answering domain independent EQL-Lite(UCQ) queries in DL-Lite is in* LOGSPACE *with respect to data complexity.*

In fact we can refine such a result by resorting to the notion of *FOL-reducibility* [6]. Intuitively, FOL-reducibility means that query answering can be reduced to evaluating FOL queries over a finite FOL interpretation corresponding to the ABox (which we assume contains assertions involving only atomic concepts and roles/relations) of a DL KB. All members of the *DL-Lite* family enjoy FOL-reducibility of UCQs [6]. We now show that FOL-reducibility holds also for domain independent *EQL-Lite*(UCQ) queries. Given an ABox $\mathcal{A}$ involving membership assertions on atomic concepts and roles only, we define the interpretation $\mathcal{I}_\mathcal{A}$ 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, answering queries in a query language $\mathcal{L}$ (contained in *EQL*) over a KB expressed in a DL $\mathcal{DL}$ is *FOL-reducible* if for every query $q \in \mathcal{L}$ and every TBox $\mathcal{T}$ expressed in $\mathcal{DL}$, there exists a FOL query $rdc(q)$ such that for every ABox $\mathcal{A}$, we have that $ans(q, (\mathcal{T}, \mathcal{A})) = eval(rdc(q), \mathcal{I}_\mathcal{A})$. Observe that FOL-reducibility is a very meaningful property from a practical point of view. Indeed, in all such cases in which query answering can be reduced to evaluation of a domain independent FOL query, then such a query can be expressed in relational algebra, i.e., in SQL. Therefore, query

[4]It is easy to see that all results for CQs in [5; 6] can be immediately extended to UCQs.

answering can take full advantage of optimization strategies provided by current DBMSs (which can be put in charge of managing ABoxes in secondary storage). As shown in [5; 6], this property holds for answering of UCQs over *DL-Lite* KBs. Now, it turns out that the possibility of processing queries by relying on DBMSs technology still holds for *EQL-Lite*(UCQ) queries over *DL-Lite* KBs.

Theorem 12 *Answering EQL-Lite(UCQ) queries in DL-Lite is FOL-reducible. Moreover, if the EQL-Lite(UCQ) query q is domain independent, then $rdc(q)$ is so as well.*

As a consequence, to answer *EQL-Lite*(UCQ) queries in *DL-Lite*, we can rely on traditional relational DBMSs.

5 Constraints

In this section we discuss an application of *EQL-Lite*($\mathcal{Q}$) to modeling integrity constraints in DL KBs. Indeed, we argue that, in the spirit of [17], a Boolean (i.e., of arity 0) *EQL-Lite*($\mathcal{Q}$) query can be seen as an integrity constraint.

In particular, a DL KB $\Sigma = (\mathcal{T}, \mathcal{A})$ (where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox) *satisfies* an integrity constraint γ expressed as a Boolean *EQL-Lite*($\mathcal{Q}$) query if $ans(\gamma, \Sigma)$ = true, i.e., $\Sigma \models_{EQL} \gamma$. Also, Σ satisfies a set $\mathcal{C}$ of integrity constraints if for all $\gamma \in \mathcal{C}$, $ans(\gamma, \Sigma)$ = true.

With this notion in place, we can revise the notion of DL KB: a *DL KB with constraints* is a triple $(\mathcal{T}, \mathcal{A}, \mathcal{C})$, where $\mathcal{T}$ is a TBox, $\mathcal{A}$ is an ABox, and $\mathcal{C}$ is a set of integrity constraints expressed in *EQL-Lite*($\mathcal{Q}$). The semantics of such a DL KB $\Sigma = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ is defined simply by specifying the set of models of Σ as follows:

$$Mod(\Sigma) = \begin{cases} Mod((\mathcal{T}, \mathcal{A})), & \text{if } (\mathcal{T}, \mathcal{A}) \text{ satisfies } \mathcal{C} \\ \emptyset, & \text{otherwise} \end{cases}$$

We say that a DL KB with constraints Σ is satisfiable if $Mod(\Sigma)$ is non-empty. Notably, the results presented in the previous sections provide several cases where checking satisfiability of DL KBs with constraints can be effectively done, and the data complexity of such a check is fully characterized.

Also, the notions of Σ-interpretation, logical implication, and certain answers of *EQL* queries, easily extend to DL KBs with constraints, as well as all complexity results for query answering in the new setting.

6 Conclusions

Motivated by various needs related to querying DL KBs, 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 various interesting cases.

We are currently working on *EQL-Lite*($\mathcal{Q}$) for *DL-Lite* KBs, for the case where $\mathcal{Q}$ is the query language whose queries are either UCQs or comparison atoms involving values taken from a set of given domains, and we are currently implementing such an extended language within an existing DL reasoning 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 AAAI 2005*, pages 1670–1671, 2005.

[3] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI 2005*, pages 364–369, 2005.

[4] 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.

[5] 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.

[6] 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*, pages 260–270, 2006.

[7] 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.

[8] F. M. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. An epistemic operator for description logics. *Artificial Intelligence*, 100(1–2):225–274, 1998.

[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] V. Haarslev, R. Möller, and M. Wessel. A high performance semantic web query answering engine. In *Proc. of DL 2005*, 2005.

[11] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *J. of Log. and Comp.*, 9(3):385–410, 1999.

[12] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic*. Methuen, London, 1984.

[13] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of IJCAI 2005*, pages 466–471, 2005.

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

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

[16] M. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of answering unions of conjunctive queries in $\mathcal{SHIQ}$. In *Proc. of DL 2006*, 2006.

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

[18] E. Sirin and B. Parsia. Pellet system description. In *Proc. of DL 2006*, 2006.