Query Answering in Expressive Variants of DL-Lite*

A. Artale[1], D. Calvanese[1], R. Kontchakov[2] and M. Zakharyaschev[2]

[1] Faculty of Computer Science
Free University of Bozen-Bolzano
I-39100 Bolzano, Italy
lastname@inf.unibz.it

[2] School of Comp. Science and Inf. Sys.
Birkbeck College
London WC1E 7HX, UK
{roman,michael}@dcs.bbk.ac.uk

Abstract. The use of ontologies in various application domains, such as Data Integration, the Semantic Web, or ontology-based data management, where ontologies provide the access to large amounts of data, is posing challenging requirements w.r.t. the trade-off between the expressive power of a Description Logic and the efficiency of reasoning. The logics of the *DL-Lite* family were specifically designed to meet such requirements and optimized w.r.t. the data complexity of answering complex types of queries. In this paper, we propose *DL-Lite*$_{bool}$, an extension of *DL-Lite* with full Booleans and number restrictions, and study the complexity of reasoning in *DL-Lite*$_{bool}$ and its significant sub-logics. We obtain our results, together with useful insights into the properties of the studied logics, by a novel reduction to the one-variable fragment of first-order logic. We study the computational complexity of satisfiability and subsumption, and the data complexity of answering positive existential queries (which extend unions of conjunctive queries). Notably, we extend the LogSpace upper bound for the data complexity of answering unions of conjunctive queries in *DL-Lite* to positive queries and to the possibility of expressing also number restrictions, hence local functionality.

1 Introduction

Description Logics (DLs) provide the formal foundation for ontologies, and the tasks related to the use of ontologies in various application domains are posing new and challenging requirements w.r.t. the trade-off between the expressive power of a DL and the efficiency of reasoning over knowledge bases (KBs) expressed in the DL. On the one hand, it is expected that the DL provides the ability to express TBoxes without limitations. On the other hand, tractable reasoning is essential in a context where ontologies become large and/or are used to access large amounts of data. This is a scenario emerging, e.g., in Data Integration [1], the Semantic Web [2], P2P data management [3, 4], ontology-based data

* This paper is an abridged version of a paper published in the Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007). **Acknowledgements.** Authors partially supported by the U.K. EPSRC grant GR/S63175, the FET project TONES (Thinking ONtologiES), funded within the EU 6th Framework Programme under contract FP6-7603, the KnowledgeWeb and InterOp EU funded projects and by the PRIN 2006 project NGS (New Generation Search), funded by MIUR.

access [5, 6], and biological data management. These new requirements have led to the proposal of novel DLs with PTime algorithms for reasoning over KBs (composed of a TBox storing intensional information, and an ABox representing the extensional data), such as those of the $\mathcal{EL}$-family [7, 8] and of the *DL-Lite* family [9, 10].

The logics of the *DL-Lite* family, in addition to having inference that is polynomial in the size of the whole KB, have been designed with the aim of providing efficient access to large data repositories. The data that need to be accessed are assumed to be stored in a standard relational database (RDB), and one is interested in expressing, through the ontology, sufficiently complex queries to such data that go beyond the simple *instance checking* case (i.e., asking for instances of single concepts and roles). The logics of the *DL-Lite* family are tailored towards such a task, in other words, they are specifically optimized w.r.t. *data complexity*. More precisely, for the various versions of *DL-Lite*, answering conjunctive queries or their union (UCQs) [11] can be done in LogSpace in data complexity [9]. Indeed, the aim of the original line of research on the *DL-Lite* family was precisely to establish the maximal subset of DLs constructs for which one can devise query answering techniques that leverage on RDB technology, and thus guarantee performance and scalability (see FOL-reducibility in [9]). Clearly, a requirement for this is that the data complexity of query answering stays within LogSpace.

In this paper, we pursue a similar objective and aim at providing useful insights for the investigation of the computational properties of the logics in the *DL-Lite* family. We extend the basic *DL-Lite* with full Booleans and number restrictions, obtaining the logic we call *DL-Lite*$_{bool}$, and introduce two sublanguages of it, *DL-Lite*$_{krom}$ and *DL-Lite*$_{horn}$. Notably, the latter strictly extends basic *DL-Lite* with number restrictions, and hence *local* (as opposed to global) functionality. We then characterize the first-order logic nature of this class of newly introduced DLs by showing their strong connection with the *one variable fragment* $\mathcal{QL}^1$ of first-order logic. The gained understanding allows us also to derive novel results on the computational complexity of inference for the newly introduced variants of *DL-Lite*.

Specifically, we show that KB satisfiability (or subsumption w.r.t. a KB) is NLogSpace-complete for *DL-Lite*$_{krom}$, P-complete for *DL-Lite*$_{horn}$, and NP-complete (resp. coNP-complete) for *DL-Lite*$_{bool}$. We prove that data complexity of both satisfiability and instance checking is in LogSpace for *DL-Lite*$_{bool}$. We then look into the data complexity of answering *positive existential queries*, which extend the well-known class of UCQs by allowing for an unrestricted interaction of conjunction and disjunction. We extend the LogSpace upper bound already known for UCQs in *DL-Lite* to positive existential queries in *DL-Lite*$_{horn}$. Due essentially to the presence of disjunction, the problem is coNP-hard for *DL-Lite*$_{krom}$, and hence for *DL-Lite*$_{bool}$ [10].

The *DL-Lite*$_{bool}$ family has been shown to be expressive enough to capture conceptual data models like UML and Extended ER [12]. Such correspondence provided new complexity results for reasoning over various fragments of the Extended ER language.

The rest of the paper is structured as follows. In the next section we introduce the three variants of *DL-Lite* mentioned above. Then we exhibit the translation to $\mathcal{QL}^1$ and derive the complexity results for satisfiability and subsumption. We proceed with the analysis of data complexity, and conclude with techniques and data complexity results for answering positive existential queries. (All proofs can be found at `http://www.dcs.bbk.ac.uk/~roman`.)

2 The *DL-Lite* family

We begin by introducing the following extension $DL\text{-}Lite_{bool}$ of the description logic *DL-Lite* [9, 10]. The language of $DL\text{-}Lite_{bool}$ contains *object names* $a_0, a_1, \ldots$, *concept names* $A_0, A_1, \ldots$ and *role names* $P_0, P_1, \ldots$. Complex *roles* R and *concepts* C of $DL\text{-}Lite_{bool}$ are defined as follows:

$$R \;::=\; P_k \;\mid\; P_k^-, \qquad\qquad B \;::=\; \bot \;\mid\; A_k \mid\; \geq q\,R,$$
$$C \;::=\; B \;\mid\; \neg C \;\mid\; C_1 \sqcap C_2,$$

where $q \geq 1$. Concepts of the form B are called *basic concepts*. A $DL\text{-}Lite_{bool}$ *TBox*, $\mathcal{T}$, consists of axioms of the form $C_1 \sqsubseteq C_2$, and an *ABox*, $\mathcal{A}$, of assertions of the form $A_k(a_i)$ or $P_k(a_i, a_j)$. Together $\mathcal{T}$ and $\mathcal{A}$ constitute a $DL\text{-}Lite_{bool}$ *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. (Note that, assertions involving complex concepts $C(a_i)$ and inverse roles $P_k^-(a_i, a_j)$ can be expressed as $A_C(a_i), A_C \sqsubseteq C$ and $P_k(a_j, a_i)$, respectively, where A_C is a fresh concept name.)

A $DL\text{-}Lite_{bool}$ *interpretation* is a structure of the form

$$\mathcal{I} \;=\; (\Delta, a_0^{\mathcal{I}}, a_1^{\mathcal{I}}, \ldots, A_0^{\mathcal{I}}, A_1^{\mathcal{I}}, \ldots, P_0^{\mathcal{I}}, P_1^{\mathcal{I}}, \ldots), \tag{1}$$

where $\Delta \neq \emptyset$, $a_i^{\mathcal{I}} \in \Delta$, $A_k^{\mathcal{I}} \subseteq \Delta$, $P_k^{\mathcal{I}} \subseteq \Delta \times \Delta$, and $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all $i \neq j$. The role and concept constructors are interpreted in $\mathcal{I}$ as usual:

$$(P_k^-)^{\mathcal{I}} = \{(y, x) \in \Delta \times \Delta \mid (x, y) \in P_k^{\mathcal{I}}\}, \qquad (\bot)^{\mathcal{I}} = \emptyset, \qquad (\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}},$$
$$(\geq q\,R)^{\mathcal{I}} = \{x \in \Delta \mid \sharp\{y \in \Delta \mid (x, y) \in R^{\mathcal{I}}\} \geq q\}, \qquad (C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}.$$

We also use the standard abbreviations $\exists R \equiv\; \geq 1\,R$ and $\leq q\,R \equiv \neg(\geq q+1\,R)$.

The *satisfaction relation* $\models$ is defined in the standard way:

$$\mathcal{I} \models C_1 \sqsubseteq C_2 \quad \text{iff} \quad C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}},$$
$$\mathcal{I} \models A_k(a_i) \quad \text{iff} \quad a_i^{\mathcal{I}} \in A_k^{\mathcal{I}},$$
$$\mathcal{I} \models P_k(a_i, a_j) \quad \text{iff} \quad (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}}.$$

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *satisfiable* if there is an interpretation, called a *model for* $\mathcal{K}$, satisfying all axioms of $\mathcal{T}$ and $\mathcal{A}$.

We also consider two sublanguages of $DL\text{-}Lite_{bool}$: the *Krom fragment*, $DL\text{-}Lite_{krom}$, and the *Horn fragment*, $DL\text{-}Lite_{horn}$ (in the following, B_i, B are basic concepts).

– A TBox of a *DL-Lite*$_{krom}$ KB only contains axioms of the form: $B_1 \sqsubseteq B_2$, or $B_1 \sqsubseteq \neg B_2$, or $\neg B_1 \sqsubseteq B_2$. KBs with such TBoxes will be called *Krom KBs*.
– A TBox of a *DL-Lite*$_{horn}$ KB only contains axioms of the form: $\bigsqcap_k B_k \sqsubseteq B$. KBs with such TBoxes will be called *Horn KBs*.

Note that the restricted negation of the original variants of *DL-Lite* [9, 10] can only express disjointness of basic concepts, while full negation in *DL-Lite*$_{bool}$ allows one to define a concept as the complement of another one. In *DL-Lite*$_{horn}$ we can express disjointness of basic concepts by using $\bot$ in the right-hand side of axioms. Also, the explicit functionality assertions of *DL-Lite* (and *DL-Lite*$_{\mathcal{F},\sqcap}$ in [10]) stating that some roles R are globally functional can be expressed in *DL-Lite*$_{bool}$ and its sublanguages *DL-Lite*$_{horn}$ and *DL-Lite*$_{krom}$ as $\geq 2\,R \sqsubseteq \bot$. Moreover, *local functionality* of a role, i.e., functionality restricted to a (basic) concept B, can be expressed in *DL-Lite*$_{bool}$ and *DL-Lite*$_{krom}$ as $B \sqsubseteq \neg(\geq 2\,R)$, and in *DL-Lite*$_{horn}$ as $B \sqcap \geq 2\,R \sqsubseteq \bot$. Thus, *DL-Lite*$_{horn}$ strictly extends *DL-Lite* and *DL-Lite*$_{\mathcal{F},\sqcap}$ with local functionality of roles and, more generally, with number restrictions.

3 Embedding *DL-Lite* into the one-variable fragment of first-order logic

Our main aim in this section is to show that satisfiability for *DL-Lite*$_{bool}$ KBs can be polynomially reduced to the satisfiability problem for the *one-variable fragment* $\mathcal{QL}^1$ of first-order logic without equality and function symbols.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a *DL-Lite*$_{bool}$ KB. Denote by $role(\mathcal{K})$ the set of role names occurring in $\mathcal{T}$ and $\mathcal{A}$, by $role^{\pm}(\mathcal{K})$ the set $\{P_k, P_k^- \mid P_k \in role(\mathcal{K})\}$, and by $ob(\mathcal{A})$ the set of object names in $\mathcal{A}$. Let $q_\mathcal{T}$ be the maximum numerical parameter in $\mathcal{T}$. Note that $q_\mathcal{T} \geq 2$ if the functionality axiom $(\geq 2\,R \sqsubseteq \bot)$ is present in $\mathcal{T}$. With every object name a_i in $ob(\mathcal{A})$ we associate the individual constant a_i of $\mathcal{QL}^1$ and with each concept name A_k the unary predicate $A_k(x)$ from the signature of $\mathcal{QL}^1$. For each role $R \in role^{\pm}(\mathcal{K})$, we also introduce $q_\mathcal{T}$ fresh unary predicates $E_qR(x)$, for $1 \leq q \leq q_\mathcal{T}$. Intuitively, $E_1P_k(x)$ and $E_1P_k^-(x)$ represent the domain and range of P_k—i.e., $E_1P_k(x)$ and $E_1P_k^-(x)$ are the sets of points with *at least one* P_k-successor and *at least one* P_k-predecessor, respectively. Predicates $E_qP_k(x)$ and $E_qP_k^-(x)$ represent the sets of points with *at least* q distinct P_k-successors and *at least* q distinct P_k-predecessors, respectively. Additionally, for every $P_k \in role(\mathcal{K})$, we take two fresh individual constants dp_k and dp_k^- of $\mathcal{QL}^1$ which will serve as 'representatives' of the points from the domain of P_k and P_k^-, respectively (provided that they are not empty). Furthermore, for each pair of objects $a_i, a_j \in ob(\mathcal{A})$ and each $R \in role^{\pm}(\mathcal{K})$, we take a fresh *propositional variable* Ra_ia_j of $\mathcal{QL}^1$ to encode $R(a_i, a_j)$. By induction on the construction of a *DL-Lite*$_{bool}$ concept C we define the $\mathcal{QL}^1$-formula C^*:

$$(\bot)^* = \bot, \quad (A_k)^* = A_k(x), \quad (\geq q\,R)^* = E_qR(x),$$
$$(\neg C)^* = \neg C^*(x), \qquad (C_1 \sqcap C_2)^* = C_1^*(x) \wedge C_2^*(x),$$

where A_k is a concept name and R is a role. Then a *DL-Lite$_{bool}$* TBox $\mathcal{T}$ corresponds to the $\mathcal{QL}^1$-sentence

$$\mathcal{T}^* = \bigwedge_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \forall x \left(C_1^*(x) \to C_2^*(x)\right). \tag{2}$$

It should be also clear how to translate an ABox $\mathcal{A}$ into $\mathcal{QL}^1$:

$$\mathcal{A}^\dagger \;=\; \bigwedge_{A_k(a_i) \in \mathcal{A}} A_k(a_i) \;\wedge\; \bigwedge_{P_k(a_i,a_j) \in \mathcal{A}} P_k a_i a_j. \tag{3}$$

The following $\mathcal{QL}^1$-sentences express some natural properties of the role domains and ranges: for every $R \in \mathit{role}^\pm(\mathcal{K})$,

$$\varepsilon(R) \;=\; \forall x \left(E_1 R(x) \to \mathit{inv}(E_1 R(dr))\right), \tag{4}$$

$$\delta(R) \;=\; \bigwedge_{q=1}^{q_{\mathcal{T}}-1} \forall x \left(E_{q+1} R(x) \to E_q R(x)\right), \tag{5}$$

where $\mathit{inv}(E_1 R(dr))$ is $E_1 P_k^-(dp_k^-)$ if $R = P_k$, and $E_1 P_k(dp_k)$ if $R = P_k^-$. Sentence (4) says that if the domain of, say, P_k is not empty then its range is not empty either: it contains the representative dp_k^-. We also need formulas relating each Ra_ia_j to the unary predicates for the role domain and range. For each $R \in \mathit{role}^\pm(\mathcal{K})$, let $R^\dagger$ be the conjunction of the following $\mathcal{QL}^1$-sentences

$$\bigwedge_{q=1}^{q_{\mathcal{T}}} \quad \bigwedge_{\substack{a, a_{j_1}, \dots, a_{j_q} \in ob(\mathcal{A}) \\ j_i \neq j_{i'} \text{ for } i \neq i'}} \Big(\bigwedge_{i=1}^{q} R a a_{j_i} \to E_q R(a)\Big), \tag{6}$$

$$\bigwedge_{a_i, a_j \in ob(\mathcal{A})} \left(R a_i a_j \to \mathit{inv}(R) a_j a_i\right), \tag{7}$$

where $\mathit{inv}(R) a_j a_i$ is the propositional variable $P_k^- a_j a_i$ if $R = P_k$, and $P_k a_j a_i$ if $R = P_k^-$. Finally, for $\mathcal{K}$, we set

$$\mathcal{K}^\dagger \;=\; \Big[\mathcal{T}^* \;\wedge \bigwedge_{R \in \mathit{role}^\pm(\mathcal{K})} \left(\varepsilon(R) \wedge \delta(R)\right)\Big] \;\wedge\; \Big[\mathcal{A}^\dagger \;\wedge \bigwedge_{R \in \mathit{role}^\pm(\mathcal{K})} R^\dagger\Big].$$

It is worth noting that all of the conjuncts of $\mathcal{K}^\dagger$ are *universal* sentences.

Theorem 1. *A DL-Lite$_{bool}$ KB $\mathcal{K}$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^\dagger$ is satisfiable.*

The translation $\mathcal{K}^\dagger$ of $\mathcal{K}$ is too lengthy to provide us with reasonably low complexity results. However, we can define a more concise equi-satisfiable translation $\mathcal{K}^\flat$ of $\mathcal{K}$, whose size is linear in the size of $\mathcal{K}$, *no matter whether the numerical parameters are coded in unary or in binary* (see `http://www.dcs.bbk.ac.uk/~roman` for the details).

Theorem 2. *Satisfiability is* NP*-complete for DL-Lite$_{bool}$ KBs,* NLogSpace*-complete for DL-Lite$_{krom}$ KBs and* P*-complete for DL-Lite$_{horn}$ KBs.*

Many other reasoning tasks are reducible to the satisfiability problem. Consider, for example, the *subsumption problem*: given a KB $\mathcal{K}$ and two concepts C and D, decide whether $\mathcal{K} \models C \sqsubseteq D$. The following complexity results hold:

Theorem 3. *The subsumption problem is* coNP*-complete for DL-Lite$_{bool}$,* NLogSpace*-complete for DL-Lite$_{krom}$ and* P*-complete for DL-Lite$_{horn}$.*

4 Data complexity

In terms of the classification suggested in [13], so far we have been considering the *combined complexity* of the satisfiability problem. When the size of data is the crucial parameter (as in ontologies for huge data sets) the most relevant complexity measure becomes *data* (or *ABox*) *complexity*, where the complexity is only measured in terms of the size of the ABox $\mathcal{A}$, while the knowledge in the TBox $\mathcal{T}$ is assumed to be fixed. In this section we show that as far as data complexity is concerned, reasoning problems for *DL-Lite*$_{bool}$ KBs can be solved using only logarithmic space in the size of the ABox.

In what follows, without loss of generality, we assume that all role names of a given KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ occur in its TBox and write $role^{\pm}(\mathcal{T})$ instead of $role^{\pm}(\mathcal{K})$. Let $\Sigma(\mathcal{T})$ be the set $\{E_1 R(dr) \mid R \in role^{\pm}(\mathcal{T})\}$ and, for $\Sigma_0 \subseteq \Sigma(\mathcal{T})$,

$$core_{\Sigma_0}(\mathcal{T}) = \bigwedge_{E_1 R(dr) \in \Sigma_0} E_1 R(dr) \quad \wedge \bigwedge_{R \in role^{\pm}(\mathcal{T})} \Big(\mathcal{T}^*[dr] \;\wedge \bigwedge_{R' \in role^{\pm}(\mathcal{T})} \big(\varepsilon(R')[dr] \;\wedge\; \delta^{\flat}(R')[dr]\big)\Big),$$

$$proj_{\Sigma_0}(\mathcal{K}, a) = \bigwedge_{inv(E_1 R(dr)) \in \Sigma(\mathcal{T}) \setminus \Sigma_0} \neg E_1 R(a) \quad \wedge\; \mathcal{T}^*[a] \quad \wedge \bigwedge_{R' \in role^{\pm}(\mathcal{T})} \delta^{\flat}(R')[a] \;\wedge\; \mathcal{A}^{\flat}(a),$$

where $\mathcal{T}^*[c]$, $\varepsilon(R')[c]$ and $\delta^{\flat}(R')[c]$ are instantiations of the universal quantifier in the respective formulas with the constant c, and $\mathcal{A}^{\flat}(a)$ is the maximal subformula of $\mathcal{A}^{\flat}$ containing only occurrences of predicates with a as their parameter.

Lemma 1. *$\mathcal{K}^{\flat}$ is satisfiable iff there is $\Sigma_0 \subseteq \Sigma(\mathcal{T})$ such that $core_{\Sigma_0}(\mathcal{T})$ and the $proj_{\Sigma_0}(\mathcal{K}, a)$, for $a \in ob(\mathcal{A})$, are all satisfiable.*

Note that $core_{\Sigma_0}(\mathcal{T})$ and the $proj_{\Sigma_0}(\mathcal{K}, a)$, for $a \in ob(\mathcal{A})$, are in essence propositional Boolean formulas and their size does not depend on the size of $\mathcal{A}$. This is clearly the case for $core_{\Sigma_0}(\mathcal{T})$ and the first three conjuncts of $proj_{\Sigma_0}(\mathcal{K}, a)$. As for the last conjunct of $proj_{\Sigma_0}(\mathcal{K}, a)$, its length does not exceed the number of concept names in $\mathcal{T}$ plus $q_{\mathcal{T}} \cdot |role^{\pm}(\mathcal{T})|$ and, therefore, only depends on the structure of $\mathcal{T}$. The above lemma states that satisfiability of a *DL-Lite*$_{bool}$ KB can be checked locally: first, for the elements dr representing the domains and ranges of all roles, and second, for every object name in the ABox. This observation suggests a high degree of parallelism in the satisfiability check.

Theorem 4. *The data complexity of the satisfiability and instance checking problems for DL-Lite$_{bool}$ KBs is in* LogSpace.

5 Query answering

By a *positive existential query* $q(x_1, \ldots, x_n)$ we understand any first-order formula constructed by means of conjunction, disjunction and existential quantification starting from atoms of the from $A_k(t)$ and $P_k(t_1, t_2)$, where A_k is a concept name, P_k is a role name, and t, t_1, t_2 are *terms* taken from the list of variables $y_0, y_1, \ldots$ and the list of object names $a_0, a_1, \ldots$, i.e.,

$$q \quad ::= \quad A_k(t) \quad | \quad P_k(t_1, t_2) \quad | \quad q_1 \wedge q_2 \quad | \; q_1 \vee q_2 \quad | \quad \exists y_i \, q.$$

The free variables of q are called its *distinguished variables* and the bound ones its *non-distinguished variables*. We write $q(x_1, \dots, x_n)$ for a query with distinguished variables $x_1, \dots, x_n$. A *conjunctive query* (CQ) is a positive existential query which contains no disjunction—that is, constructed from atoms by means of conjunction and existential quantification. Given a query $q(\boldsymbol{x})$, with $\boldsymbol{x} = x_1, \dots, x_n$, and an n-tuple $\boldsymbol{a}$ of object names, we write $q(\boldsymbol{a})$ for the result of replacing every occurrence of x_i in $q(\boldsymbol{x})$ with the ith member of $\boldsymbol{a}$. Queries containing no distinguished variables will be called *ground*.

Let $\mathcal{I}$ be a *DL-Lite$_{bool}$* model of the form (1). An *assignment* $\mathfrak{a}$ in Δ is a function associating with every variable y an element $\mathfrak{a}(y)$ of Δ. We will use the following notation: $a_i^{\mathcal{I},\mathfrak{a}} = a_i^{\mathcal{I}}$ and $y^{\mathcal{I},\mathfrak{a}} = \mathfrak{a}(y)$. Define the *satisfaction relation* for positive existential formulas with respect to a given assignment $\mathfrak{a}$:

$$\mathcal{I} \models^{\mathfrak{a}} A_k(t) \text{ iff } t^{\mathcal{I},\mathfrak{a}} \in A_k^{\mathcal{I}}, \qquad \mathcal{I} \models^{\mathfrak{a}} q_1 \land q_2 \text{ iff } \mathcal{I} \models^{\mathfrak{a}} q_1 \text{ and } \mathcal{I} \models^{\mathfrak{a}} q_2,$$

$$\mathcal{I} \models^{\mathfrak{a}} P_k(t_1, t_2) \text{ iff } (t_1^{\mathcal{I},\mathfrak{a}}, t_2^{\mathcal{I},\mathfrak{a}}) \in P_k^{\mathcal{I}}, \quad \mathcal{I} \models^{\mathfrak{a}} q_1 \lor q_2 \text{ iff } \mathcal{I} \models^{\mathfrak{a}} q_1 \text{ or } \mathcal{I} \models^{\mathfrak{a}} q_2,$$

$$\mathcal{I} \models^{\mathfrak{a}} \exists y_i\, q \text{ iff } \mathcal{I} \models^{\mathfrak{b}} q, \text{ for some } \mathfrak{b} \text{ that may differ from } \mathfrak{a} \text{ on } y_i.$$

For a ground query $q(\boldsymbol{a})$ the satisfaction relation does not depend on the assignment $\mathfrak{a}$, thus we write $\mathcal{I} \models q(\boldsymbol{a})$ instead of $\mathcal{I} \models^{\mathfrak{a}} q(\boldsymbol{a})$. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that a tuple $\boldsymbol{a}$ of objects from $ob(\mathcal{A})$ is an *answer* to $q(\boldsymbol{x})$ and write $\mathcal{K} \models q(\boldsymbol{a})$ if $\mathcal{I} \models q(\boldsymbol{a})$ whenever $\mathcal{I} \models \mathcal{K}$.

The *query answering problem* we analyse here is formulated as follows: given a *DL-Lite$_{bool}$* KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a query $q(\boldsymbol{x})$ and a tuple $\boldsymbol{a}$ of object names from $ob(\mathcal{A})$, decide whether $\mathcal{K} \models q(\boldsymbol{a})$. The variant of this problem requiring to 'list all the answers $\boldsymbol{a}$ to $q(\boldsymbol{x})$ with respect to $\mathcal{K}$' is LogSpace-equivalent to the previous one [11, Exercise 16.13]. We are interested in the *data complexity* of the query answering problem. We first recall known results [14, 10, 15] for the case of conjunctive queries and obtain the following:

Theorem 5. *The data complexity of the conjunctive and positive existential query answering problems for both DL-Lite$_{krom}$ and DL-Lite$_{bool}$ KBs is* coNP-*complete.*

Next, we show that the LogSpace data complexity upper bound for conjunctive queries over *DL-Lite* KBs established in [9, 10], can be extended to positive existential queries over *DL-Lite$_{horn}$* KBs:

Theorem 6. *The data complexity of the positive existential query answering problem for DL-Lite$_{horn}$ KBs is in* LogSpace.

6 Conclusions

The LogSpace data complexity result for query answering provides the basis for the development of algorithms that operate on a KB whose ABox is stored in a relational database (RDB), and that evaluate a query by relying on the query answering capabilities of a RDB management system, cf. [9]. The known algorithms for *DL-Lite* are based on rewriting the original query using the TBox axioms. We aim at developing a similar technique also for answering positive existential queries in *DL-Lite*$_{horn}$.

We are further investigating the complexity of logics obtained by adding further constructs to *DL-Lite*. Preliminary results show that already by adding role inclusion axioms to *DL-Lite*$_{bool}$ the combined complexity raises to ExpTime.

References

1. Lenzerini, M.: Data integration: A theoretical perspective. In: Proc. of the 21st ACM Symp. on Principles of Database Systems (PODS). (2002) 233–246
2. Heflin, J., Hendler, J.: A portrait of the Semantic Web in action. IEEE Intelligent Systems **16**(2) (2001) 54–59
3. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Logical foundations of peer-to-peer data integration. In: Proc. of the 23rd ACM Symp. on Principles of Database Systems (PODS). (2004) 241–251
4. Franconi, E., Kuper, G.M., Lopatenko, A., Zaihrayeu, I.: Queries and updates in the coDB peer to peer database system. In: Proc. of the 30th Int. Conf. on Very Large Data Bases (VLDB). (2004) 1277–1280
5. Borgida, A., Brachman, R.J., McGuinness, D.L., Resnick, L.A.: CLASSIC: A structural data model for objects. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data. (1989) 59–67
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tailoring OWL for data intensive ontologies. In: Proc. of the Workshop on OWL: Experiences and Directions (OWLED). (2005)
7. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI). (2005) 364–369
8. Baader, F., Lutz, C., Suntisrivaraporn, B.: Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? In: Proc. of the Methods for Modalities Workshop (M4M 2005). (2005)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: DL-Lite: Tractable description logics for ontologies. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI). (2005) 602–607
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: 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) 260–270
11. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison Wesley (1995)
12. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Complexity of reasoning in Entity Relationship models. In: Proc. of the 2007 Description Logic Workshop (DL). (2007)
13. Vardi, M.Y.: The complexity of relational query languages. In: Proc. of the 14th ACM SIGACT Symp. on Theory of Computing (STOC). (1982) 137–146
14. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. J. of Logic and Computation **4**(4) (1994) 423–452
15. Ortiz, M.M., Calvanese, D., Eiter, T.: Characterizing data complexity for conjunctive query answering in expressive description logics. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI). (2006)