DL-Lite in the Light of First-Order Logic

A. Artale and **D. Calvanese**
Faculty of Computer Science
Free University of Bozen-Bolzano
I-39100 Bozen-Bolzano, Italy
{artale, calvanese}@inf.unibz.it

R. Kontchakov and **M. Zakharyaschev**
School of Computer Science and Information Systems
Birkbeck College
London WC1E 7HX, U.K.
{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. a trade-off between expressive power of a DL and 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, and hence local functionality in the TBox.

Introduction

Description Logics (DLs) provide the formal foundation for ontologies (`http://owl1_1.cs.manchester.ac.uk/`), and the tasks related to the use of ontologies in various application domains are posing new and challenging requirements w.r.t. a trade-off between expressive power of a DL and 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 (Lenzerini 2002), the Semantic Web (Heflin & Hendler 2001), P2P data management (Bernstein *et al.* 2002; Calvanese *et al.* 2004; Franconi *et al.* 2004), ontology-based data access (Borgida *et al.* 1989; Calvanese *et al.* 2005b), 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 (Baader, Brandt, & Lutz 2005; Baader, Lutz, & Suntisrivaraporn 2005) and of the *DL-Lite* family (Calvanese *et al.* 2005a; 2006).

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*: for the various versions of *DL-Lite*, answering unions of conjunctive queries (UCQs) (Abiteboul, Hull, & Vianu 1995) can be done in LOGSPACE in data complexity (Calvanese *et al.* 2005a). 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 (Calvanese *et al.* 2005a)). 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*$_{bool}$ (Calvanese *et al.* 2006).

The *DL-Lite*$_{bool}$ family has been shown to be expressive enough to capture conceptual data models like UML and Extended ER (Artale *et al.* 2007). 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`.)

The *DL-Lite* Family

We begin by introducing the following extension *DL-Lite*$_{bool}$ of the description logic *DL-Lite* (Calvanese *et al.* 2005a; 2006). The language of *DL-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-Lite*$_{bool}$ are defined as follows:

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

where $q \geq 1$. Concepts of the form B are called *basic concepts*. A *DL-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-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-Lite*$_{bool}$ *interpretation* is a structure of the form

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

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:

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

We make use of the standard abbreviations $\exists R \equiv \geq 1\, R$, $\top \equiv \neg\bot$, 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-Lite*$_{bool}$ (in the following, the B_i and B are basic concepts):

(Krom fragment) A TBox of a *DL-Lite*$_{krom}$ KB only contains axioms of the form $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \neg B_2$ or $\neg B_1 \sqsubseteq B_2$. KBs with such TBoxes are called *Krom KBs*.

(Horn fragment) A TBox of a *DL-Lite*$_{horn}$ KB only contains axioms of the form $\bigsqcap_k B_k \sqsubseteq B$. KBs with such TBoxes are called *Horn KBs*.

Note that the restricted negation of the original variants of *DL-Lite* and *DL-Lite*$_{\mathcal{F},\sqcap}$ (Calvanese *et al.* 2005a; 2006) can only express disjointness of basic concepts, while the full negation in *DL-Lite*$_{bool}$ allows one to define a concept as the complement of another one. In *DL-Lite*$_{horn}$ one can express disjointness of basic concepts B_k by $\bigsqcap_k B_k \sqsubseteq \bot$. The explicit functionality axioms of *DL-Lite*$_{\mathcal{F},\sqcap}$ stating that a role R is globally functional can be represented in both *DL-Lite*$_{krom}$ and *DL-Lite*$_{horn}$ as $\geq 2\, R \sqsubseteq \bot$. Moreover, the two languages are capable of expressing *local functionality* of a role, i.e., functionality restricted to a (basic) concept B: $B \sqsubseteq \neg(\geq 2\, R)$ in *DL-Lite*$_{krom}$ and $B \sqcap \geq 2\, R \sqsubseteq \bot$ in *DL-Lite*$_{horn}$. Therefore, *DL-Lite*$_{horn}$ strictly extends *DL-Lite* and *DL-Lite*$_{\mathcal{F},\sqcap}$ with local functionality of roles and, more generally, with number restrictions.

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 $\{P_k, P_k^- \mid P_k \in \textit{role}(\mathcal{K})\}$, 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 $R \in \textit{role}^{\pm}(\mathcal{K})$, we introduce $q_{\mathcal{T}}$ fresh unary predicates $E_q R(x)$, for $1 \leq q \leq q_{\mathcal{T}}$. Intuitively, $E_1 P_k(x)$ and $E_1 P_k^-(x)$ represent the domain and range of P_k—i.e., $E_1 P_k(x)$ and $E_1 P_k^-(x)$ are the sets of points with *at least one* P_k-successor and *at least one* P_k-predecessor, respectively. Predicates $E_q P_k(x)$ and $E_q P_k^-(x)$ 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 $P_k \in \textit{role}(\mathcal{K})$, 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 $a_i, a_j \in \mathit{ob}(\mathcal{A})$ and each $R \in \mathit{role}^{\pm}(\mathcal{K})$, 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^*:

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

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

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

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

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

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

$$\varepsilon(R) \;=\; \forall x \, \big(E_1 R(x) \to \mathit{inv}(E_1 R(dr))\big), \tag{4}$$
$$\delta(R) \;=\; \textstyle\bigwedge_{q=1}^{q_\mathcal{T}-1} \forall x \, \big(E_{q+1} R(x) \to E_q R(x)\big), \tag{5}$$

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

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

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

where $\mathit{inv}(R)a_ja_i$ 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

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

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.*

Proof. ($\Leftarrow$) Let $\mathfrak{M}$ be an Herbrand model (in the signature of $\mathcal{K}^\dagger$) satisfying $\mathcal{K}^\dagger$; for details see, e.g., (Rautenberg 2006). We denote the domain of $\mathfrak{M}$ by D (it consists of all the constants occurring in $\mathcal{K}^\dagger$), and the interpretations of (unary) predicates A, propositional variables R and constants a of $\mathcal{QL}^1$ in $\mathfrak{M}$ by $A^\mathfrak{M}$, $R^\mathfrak{M}$ and $a^\mathfrak{M}$, respectively. We construct inductively a *DL-Lite*$_{bool}$ model $\mathcal{I}$ based on some domain $\Delta \supseteq D$, which will be defined as the union $\Delta = \bigcup_{m=0}^\infty W_m$ with $W_0 = D$. Each set W_{m+1}, for $m \geq 0$, is constructed by adding to W_m some new elements that are fresh *copies* of certain elements from W_0 (i.e., $W_m \subseteq W_{m+1}$ for $m \geq 0$). If such a new element w' is a copy of $w \in W_0$ then we write $cp(w') = w$, while, for $w \in W_0$, we let $cp(w) = w$ (thus, $cp\colon \Delta \to W_0$). The set $W_m \setminus W_{m-1}$, for $m \geq 0$, will be denoted by V_m (for convenience, let $W_{-1} = \emptyset$ so that $V_0 = D$). The interpretations of the objects a in $\mathcal{I}$ are given by their interpretations in $\mathfrak{M}$, namely, $a^\mathcal{I} = a^\mathfrak{M} \in W_0$. The interpretations $A^\mathcal{I}$ of concept names A in $\mathcal{I}$ are defined by taking

$$A^\mathcal{I} \;=\; \{w \in \Delta \mid \mathfrak{M} \models A^*[cp(w)]\}. \tag{8}$$

The interpretation $P_k^\mathcal{I}$ of a role P_k in $\mathcal{I}$ will be defined inductively as the union

$$P_k^\mathcal{I} \;=\; \textstyle\bigcup_{m=0}^\infty P_k^m, \qquad \text{where} \quad P_k^m \subseteq W_m \times W_m,$$

along with the construction of Δ. For $R \in \mathit{role}^{\pm}(\mathcal{K})$, we define the *required R-rank $r(R,d)$ of a point $d \in D$* by taking

$$r(R,d) = \begin{cases} 0, & \text{if } \mathfrak{M} \models \neg E_1 R[d], \\ q, & \text{if } \mathfrak{M} \models (E_q R \wedge \neg E_{q+1} R)[d], 1 \leq q < q_\mathcal{T}, \\ q_\mathcal{T}, & \text{if } \mathfrak{M} \models E_{q_\mathcal{T}} R[d]. \end{cases}$$

It follows from (5) that $r(R,d)$ is indeed a function, and if $d \in D$ and $r(R,d) = q$ then $\mathfrak{M} \models E_{q'} R[d]$, for $1 \leq q' \leq q$, and $\mathfrak{M} \models \neg E_{q'} R[d]$, for $q < q' \leq q_\mathcal{T}$. We also define the *actual R-rank $r_m(R,w)$ of $w \in \Delta$ at step m* by taking

$$r_m(R,w) = \begin{cases} q, & \text{if } w \in (\geq q\, R^m.W_m) \setminus (\geq q{+}1\, R^m.W_m) \\ & \qquad \text{and } 0 \leq q < q_\mathcal{T}, \\ q_\mathcal{T}, & \text{if } w \in (\geq q_\mathcal{T}\, R^m.W_m), \end{cases}$$

where $R^m = P_k^m$ if $R = P_k$, $R^m = (P_k^m)^-$ if $R = P_k^-$, and, for $W \subseteq \Delta$, $R \subseteq \Delta \times \Delta$ and $0 \leq q \leq q_\mathcal{T}$, $(\geq q\, R.W) \;=\; \big\{w \in W \mid \sharp\{v \mid (w,v) \in R\} \geq q\big\}$. For the basis of induction we set, for each $P_k \in \mathit{role}(\mathcal{K})$,

$$P_k^0 \;=\; \{(a_i^\mathfrak{M}, a_j^\mathfrak{M}) \in W_0 \times W_0 \mid \mathfrak{M} \models P_k a_i a_j\}. \tag{9}$$

Note that, by (6) and (7), for all $R \in \mathit{role}^{\pm}(\mathcal{K})$ and $w \in W_0$,

$$r_0(R,w) \;\leq\; r(R, cp(w)). \tag{10}$$

Suppose now that W_m and the P_k^m, for $m \geq 0$, have already been defined. If we had $r_m(R,w) = r(R, cp(w))$, for all $R \in \mathit{role}^{\pm}(\mathcal{K})$ and $w \in W_m$, then the model would be as required. However, in general this is not the case because there may be some 'defects' in the sense that the actual rank of some points is smaller than the required rank. For a role $P_k \in \mathit{role}(\mathcal{K})$, consider the following sets of defects in P_k^m:

$$\Lambda_k^m \;=\; \{w \in V_m \mid r_m(P_k, w) < r(P_k, cp(w))\},$$
$$\Lambda_k^{m-} \;=\; \{w \in V_m \mid r_m(P_k^-, w) < r(P_k^-, cp(w))\}.$$

The purpose of, say, Λ_k^m is to identify those 'defective' points $w \in V_m$ from which precisely $r(P_k, cp(w))$ distinct P_k-arrows should start (according to $\mathfrak{M}$), but some arrows are still missing (only $r_m(P_k, w)$ many arrows exist). To 'cure' these defects, we extend W_m to W_{m+1} and P_k^m to P_k^{m+1} according to the following rules:

(Λ_k^m) Let $w \in \Lambda_k^m$, $q = r(P_k, cp(w)) - r_m(P_k, w)$ and $d = cp(w)$. So $\mathfrak{M} \models E_{q'} P_k[d]$, for some $q' \geq q > 0$. Then, by (5), $\mathfrak{M} \models E_1 P_k[d]$ and, by (4), $\mathfrak{M} \models E_1 P_k^-[dp_k^-]$. In this case we take q *fresh* copies $w_1', \dots, w_q'$ of dp_k^-, set $cp(w_i') = dp_k^-$, add them to W_{m+1} and add the pairs (w, w_i') to P_k^{m+1}.

(Λ_k^{m-}) is the mirror image of the above.

Observe the following important property of the construction: for all $m_0 \geq 0$, $w \in V_{m_0}$ and $R \in role^{\pm}(\mathcal{K})$,

$$r_m(R, w) = \begin{cases} 0, & \text{if } m < m_0, \\ q, & \text{if } m = m_0,\ q \leq r(R, cp(w)), \\ r(R, cp(w)), & \text{if } m > m_0. \end{cases} \quad (11)$$

This claim can be proved by considering all possible cases for the relationship between m and m_0.

It follows from this property that, for all $R \in role^{\pm}(\mathcal{K})$, $1 \leq q \leq q_{\mathcal{T}}$ and $w \in \Delta$,

$$\mathfrak{M} \models E_q R[cp(w)] \qquad \text{iff} \qquad w \in (\geq q\, R^{\mathcal{I}}.\Delta). \quad (12)$$

Now we show by induction on the construction of concepts C in $\mathcal{K}$ that, for every $w \in \Delta$,

$$\mathfrak{M} \models C^*[cp(w)] \qquad \text{iff} \qquad w \in C^{\mathcal{I}}. \quad (13)$$

The basis of induction is trivial for $C = \bot$, follows from (8) for $C = A_k$ and from (12) for $C = \geq q\, R$. The induction step for the Booleans ($C = \neg C_1$ and $C = C_1 \sqcap C_2$) easily follows from the induction hypothesis. Finally, we show that for each statement $\psi \in \mathcal{T} \cup \mathcal{A}$,

$$\mathfrak{M} \models \psi^{\dagger} \qquad \text{iff} \qquad \mathcal{I} \models \psi. \quad (14)$$

The case $\psi = C_1 \sqsubseteq C_2$ follows from (13) and $\psi = A_k(a_i)$ from the definition of $A_k^{\mathcal{I}}$. For $\psi = P_k(a_i, a_j)$, we have $(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}}$ iff, by construction of $P_k^{\mathcal{I}}$, $(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^0$ iff, by (9), $\mathfrak{M} \models P_k a_i a_j$. Therefore, $\mathcal{I} \models \mathcal{K}$.

The proof of ($\Rightarrow$) is straightforward. □

The translation $\mathcal{K}^{\dagger}$ of $\mathcal{K}$ is too lengthy to provide us with reasonably low complexity results: $|\mathcal{K}^{\dagger}| \leq const \cdot |\mathcal{K}| + |ob(\mathcal{A})|^{q_{\mathcal{T}}}$. Let us now define a more concise translation of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ into $\mathcal{QL}^1$. For $R \in role^{\pm}(\mathcal{K})$, let $Q_{\mathcal{T}}^R$ be the subset of the natural numbers containing 1 and all the numerical parameters q for which the concept $\geq q\, R$ occurs in $\mathcal{T}$ (recall that the ABox does not contain numerical parameters). Then we set

$$\mathcal{K}^{\flat} \;=\; \Big[\mathcal{T}^* \;\wedge\; \bigwedge\nolimits_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^{\flat}(R)\big)\Big] \quad \wedge \quad \mathcal{A}^{\flat},$$

where $\mathcal{T}^*$ and $\varepsilon(R)$ are as before (see (2) and (4)), and

$$\delta^{\flat}(R) \;=\; \bigwedge_{\substack{q, q' \in Q_{\mathcal{T}}^R,\ q' > q \text{ and} \\ q' > q'' > q \text{ for no } q'' \in Q_{\mathcal{T}}^R}} \forall x \big(E_{q'} R(x) \rightarrow E_q R(x)\big), \quad (15)$$

$$\mathcal{A}^{\flat} \;=\; \bigwedge_{A(a_i) \in \mathcal{A}} A(a_i) \quad \wedge \quad \bigwedge_{R \in role^{\pm}(\mathcal{K}), a \in ob(\mathcal{A})} E_{q_{R,a}} R(a), \quad (16)$$

where $q_{R,a}$ is the maximum number from $Q_{\mathcal{T}}^R$ such that there are $q_{R,a}$ many distinct a_i with $P_k(a, a_i) \in \mathcal{A}$, for $R = P_k$, and $P_k(a_i, a) \in \mathcal{A}$, for $R = P_k^-$. Note that $\mathcal{K}^{\flat}$, unlike $\mathcal{K}^{\dagger}$, does not contain propositional variables $Ra_i a_j$; indeed, the actual connections between named objects (stated in the ABox) are of no importance at all; what really matters are the unary 'types' of named objects $a \in ob(\mathcal{A})$, that is, the sets of all concepts C from $\mathcal{K}$ such that $\mathcal{K} \models C(a)$. This information is enough to restore the relations between the named objects required by $\mathcal{K}$.

Now both the size of $\mathcal{A}^{\flat}$ and the size of $\mathcal{K}^{\flat}$ are linear in the size of $\mathcal{A}$ and $\mathcal{K}$, respectively, *no matter whether the numerical parameters are coded in unary or in binary*.

From the fact that $\mathcal{K}^{\dagger}$ is satisfiable iff $\mathcal{K}^{\flat}$ is satisfiable the following corollary holds.

Corollary 2. *A DL-Lite$_{bool}$ KB $\mathcal{K}$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^{\flat}$ is satisfiable.*

As a consequence of Corollary 2 we obtain the following:

Theorem 3. *The satisfiability problem is* NP*-complete for DL-Lite$_{bool}$ KBs,* NLOGSPACE*-complete for DL-Lite$_{krom}$ KBs and* P*-complete for DL-Lite$_{horn}$ KBs.*

Proof. As $\mathcal{K}^{\flat}$ contains no function symbols, its Herbrand universe consists of all constants occurring in it, i.e., $ob(\mathcal{A})$ and the dr, $R \in role^{\pm}(\mathcal{K})$. Therefore, satisfiability of $\mathcal{K}$ is polynomially reducible to satisfiability of a set of propositional formulas, namely, the formulas obtained from $\mathcal{K}^{\flat}$ by replacing x with each of the constants occurring in $\mathcal{K}^{\flat}$ (because $\mathcal{K}^{\flat}$ is a universal formula). It remains to recall that the satisfiability of Boolean formulas is in NP, of 2-CNFs in NLOGSPACE, and of propositional Horn formulas in P (Papadimitriou 1994; Kozen 2006). The matching lower bounds also follow from the complexity of the respective fragments of propositional Boolean logic. □

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 $\mathcal{K} \models C \sqsubseteq D$. Since subsumption and non-satisfiability of KBs are reducible to each other and CONLOGSPACE=NLOGSPACE by the Immerman-Szelepcsényi theorem (see, e.g., Kozen, 2006) the following holds:

Theorem 4. *The subsumption problem is* CONP*-complete for DL-Lite$_{bool}$,* NLOGSPACE*-complete for DL-Lite$_{krom}$ and* P*-complete for DL-Lite$_{horn}$.*

Other reasoning tasks are analysed in the same way. In particular, a reduction for the instance checking problem can be found in the next section.

Data Complexity

In terms of the classification suggested in (Vardi 1982), 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. We remind the reader (see e.g., Kozen, 2006) that a problem belongs to the complexity class LOGSPACE if there is a two-tape Turing machine M such that, starting with an input of length n written on the *read-only input tape*, M stops in an accepting or rejecting state having used at most $\log n$ cells of the (initially blank) *read/write work tape*.

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 $\mathit{role}^{\pm}(\mathcal{T})$ instead of $\mathit{role}^{\pm}(\mathcal{K})$. Let $\Sigma(\mathcal{T})$ be the set $\{E_1 R(dr) \mid R \in \mathit{role}^{\pm}(\mathcal{T})\}$ and, for $\Sigma_0 \subseteq \Sigma(\mathcal{T})$,

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

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

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 5. *$\mathcal{K}^{\flat}$ is satisfiable iff there is $\Sigma_0 \subseteq \Sigma(\mathcal{T})$ such that $\mathit{core}_{\Sigma_0}(\mathcal{T})$ and the $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$, for $a \in \mathit{ob}(\mathcal{A})$, are all satisfiable.*

Note that $\mathit{core}_{\Sigma_0}(\mathcal{T})$ and the $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$, for $a \in \mathit{ob}(\mathcal{A})$, are in essence propositional Boolean formulas and their size does not depend on the size of $\mathcal{A}$. This is clearly the case for $\mathit{core}_{\Sigma_0}(\mathcal{T})$ and the first three conjuncts of $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$. As for the last conjunct of $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$, its length does not exceed the number of concept names in $\mathcal{T}$ plus $q_{\mathcal{T}} \cdot |\mathit{role}^{\pm}(\mathcal{T})|$ 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 6. *The data complexity of satisfiability and instance checking for DL-Lite*$_{bool}$ *KBs is in* LOGSPACE.

Proof. The instance checking problem is reducible to the (un)satisfiability problem: an object a is an instance of a basic concept B in every model of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff the KB $\big(\mathcal{T} \cup \{A_{\neg B} \sqsubseteq \neg B\},\ \mathcal{A} \cup \{A_{\neg B}(a)\}\big)$ is not satisfiable, where $A_{\neg B}$ is a fresh concept name. The following deterministic algorithm checks whether a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfiable:

- for every subset Σ_0 of $\Sigma(\mathcal{T})$, repeat the following:

(c) compute $\mathit{core}_{\Sigma_0}(\mathcal{T})$ and check whether it is satisfiable;

(p) for every object name $a \in \mathit{ob}(\mathcal{A})$,

- compute the $q_{R,a}$, for $R \in \mathit{role}^{\pm}(\mathcal{T})$,
- compute $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$, check whether it is satisfiable.

This algorithm requires space bounded by a logarithmic function in the size $|\mathcal{A}|$ of the ABox. Indeed, in order to enumerate all subsets Σ_0 of $\Sigma(\mathcal{T})$ one needs $|\mathit{role}^{\pm}(\mathcal{T})|$ cells of the work tape—this does not depend on $|\mathcal{A}|$. At step **(c)**, the size of $\mathit{core}_{\Sigma_0}(\mathcal{T})$ does not depend on $|\mathcal{A}|$ either, and whether this formula is satisfiable can be checked *deterministically* (though in time exponential and in space linear in the length of the formula). At step **(p)** we enumerate all elements of $\mathit{ob}(\mathcal{A})$, and this requires $\log |\mathcal{A}|$ cells on the working tape. Next, the $q_{R,a}$, for $R \in \mathit{role}^{\pm}(\mathcal{T})$, can be computed using $q_{\mathcal{T}} \cdot \log |\mathcal{A}|$ of extra space: for every $1 \leq q \leq q_{\mathcal{T}}$, one enumerates all q-tuples $(a_{i_1}, \ldots, a_{i_q})$ of distinct objects in $\mathit{ob}(\mathcal{A})$ and checks whether, for every $1 \leq j \leq q$, $P_k(a, a_{i_j}) \in \mathcal{A}$, if $R = P_k$, and $P_k(a_{i_j}, a) \in \mathcal{A}$, if $R = P_k^-$. The maximum such q is the required number $q_{R,a}$ (cf. (16)). Finally, for each $a \in \mathit{ob}(\mathcal{A})$, the size of $\mathit{proj}_{\Sigma_0}(\mathcal{K}, a)$ does not depend on $|\mathcal{A}|$ and its satisfiability can be checked deterministically. □

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.,

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

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

$$\begin{aligned}
\mathcal{I} &\models^{\mathfrak{a}} A_k(t) && \text{iff} \quad t^{\mathcal{I},\mathfrak{a}} \in A_k^{\mathcal{I}},\\
\mathcal{I} &\models^{\mathfrak{a}} P_k(t_1, t_2) && \text{iff} \quad (t_1^{\mathcal{I},\mathfrak{a}}, t_2^{\mathcal{I},\mathfrak{a}}) \in P_k^{\mathcal{I}},\\
\mathcal{I} &\models^{\mathfrak{a}} q_1 \wedge q_2 && \text{iff} \quad \mathcal{I} \models^{\mathfrak{a}} q_1 \text{ and } \mathcal{I} \models^{\mathfrak{a}} q_2,\\
\mathcal{I} &\models^{\mathfrak{a}} q_1 \vee q_2 && \text{iff} \quad \mathcal{I} \models^{\mathfrak{a}} q_1 \text{ or } \mathcal{I} \models^{\mathfrak{a}} q_2,\\
\mathcal{I} &\models^{\mathfrak{a}} \exists y_i\, q && \text{iff} \quad \mathcal{I} \models^{\mathfrak{b}} q, \text{ for some } \mathfrak{b} \text{ that may differ from } \mathfrak{a} \text{ on } y_i.
\end{aligned}$$

For a ground query $q(\vec{a})$ the satisfaction relation does not depend on the assignment $\mathfrak{a}$, thus we write $\mathcal{I} \models q(\vec{a})$ instead of $\mathcal{I} \models^{\mathfrak{a}} q(\vec{a})$. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that a tuple $\vec{a}$ of objects from $\mathit{ob}(\mathcal{A})$ is an *answer* to $q(\vec{x})$ and write $\mathcal{K} \models q(\vec{a})$ if $\mathcal{I} \models q(\vec{a})$ 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 $q(\vec{x})$ and a tuple $\vec{a}$ of object names from $ob(\mathcal{A})$, decide whether $\mathcal{K} \models q(\vec{a})$. The variant of this problem requiring to 'list all the answers $\vec{a}$ to $q(\vec{x})$ with respect to $\mathcal{K}$' is LOGSPACE-equivalent to the previous one (Abiteboul, Hull, & Vianu 1995, Exercise 16.13). We are interested in the *data complexity* of the query answering problem.

We first recall known results (Donini *et al.* 1994; Calvanese *et al.* 2006; Ortiz, Calvanese, & Eiter 2006) for the case of conjunctive queries and obtain the following:

Theorem 7. *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.*

Proof. The lower bound was established in (Calvanese *et al.* 2006) by adapting the proof in (Donini *et al.* 1994). The matching upper bound follows from results on data complexity of conjunctive query answering in expressive DLs (Ortiz, Calvanese, & Eiter 2006): every positive existential query q can be transformed into an equivalent UCQ and, although the resulting UCQ may be exponential in the length of q, this clearly does not affect data complexity. □

Next, we show that the LOGSPACE data complexity upper bound (Calvanese *et al.* 2005a; 2006) for conjunctive queries over *DL-Lite* KBs, can be extended to positive existential queries over *DL-Lite*$_{horn}$ KBs:

Theorem 8. *The data complexity of the positive existential query answering problem for DL-Lite*$_{horn}$ *KBs is in* LOGSPACE.

Proof. We only sketch the idea of the proof. (1) We construct a *single*, but possibly infinite, model $\mathcal{I}_0$ which provides all answers to all positive existential queries with respect to a given Horn KB. (2) We show that to find all answers to a given query it is enough to consider some *finite* part of $\mathcal{I}_0$ the size of which does not depend on the given ABox but only on the number of distinguished $\vec{x}$ and non-distinguished variables $\vec{y}$ in the given query as well as the size of the TBox. (3) The LOGSPACE query answering algorithm considers then all proper possible assignments of elements in that finite part of $\mathcal{I}_0$ to the variables $\vec{x}$, $\vec{y}$, computes the corresponding types—the concepts that contain these elements—and, finally, evaluates the query. □

It should be noted that the actual data complexity may be somewhat lower: the upper bounds in Theorems 6 and 8 can be improved to AC_0.

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. (Calvanese *et al.* 2005a). 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.

Acknowledgements. The authors were partially supported by the EU funded projects TONES FP6-7603, KnowledgeWeb, and InterOp and the U.K. EPSRC grant GR/S63175.

References

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

Artale, A.; Calvanese, D.; Kontchakov, R.; Ryzhikov, V.; and Zakharyaschev, M. 2007. Complexity of reasoning in entity relationship models. In *Proc. of DL 2007*.

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

Baader, F.; Lutz, C.; and Suntisrivaraporn, B. 2005. Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? In *Proc. of M4M 2005*.

Bernstein, P. A.; Giunchiglia, F.; Kementsietsidis, A.; Mylopoulos, J.; Serafini, L.; and Zaihrayeu, I. 2002. Data management for peer-to-peer computing: A vision. In *Proc. of WebDB 2002*.

Borgida, A.; Brachman, R. J.; McGuinness, D. L.; and Resnick, L. A. 1989. CLASSIC: A structural data model for objects. In *Proc. of ACM SIGMOD*, 59–67.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2004. Logical foundations of peer-to-peer data integration. In *Proc. of PODS 2004*, 241–251.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005a. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, 602–607.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005b. Tailoring OWL for data intensive ontologies. In *Proc. of OWLED 2005*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006. Data complexity of query answering in description logics. In *Proc. of KR 2006*, 260–270.

Donini, F. M.; Lenzerini, M.; Nardi, D.; and Schaerf, A. 1994. Deduction in concept languages: From subsumption to instance checking. *J. of Log. and Comp.* 4(4):423–452.

Franconi, E.; Kuper, G. M.; Lopatenko, A.; and Zaihrayeu, I. 2004. Queries and updates in the coDB peer to peer database system. In *Proc. of VLDB 2004*, 1277–1280.

Heflin, J., and Hendler, J. 2001. A portrait of the Semantic Web in action. *IEEE Intelligent Systems* 16(2):54–59.

Kozen, D. 2006. *Theory of Computation*. Springer.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, 233–246.

Ortiz, M. M.; Calvanese, D.; and Eiter, T. 2006. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of AAAI 2006*.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison Wesley Publ. Co.

Rautenberg, W. 2006. *A Concise Introduction to Mathematical Logic*. Springer.

Vardi, M. Y. 1982. The complexity of relational query languages. In *Proc. of STOC'82*, 137–146.