DL-Lite in the light of first-order logic

A. Artale, D. Calvanese, R. Kontchakov and M. Zakharyaschev

March 26, 2007

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 DL 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, and hence local functionality in the TBox.

1 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. 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 [17], the Semantic Web [15], P2P data management [5, 11, 14], ontology-based data access [7, 9], 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 [4, 3] and of the *DL-Lite* family [8, 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) [1] can be done in LogSpace in data complexity [8]. 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 the data complexity of query answering stays within LogSpace. For such DLs one can then devise query answering techniques that leverage on RDB technology, and thus guarantee performance and scalability.

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 $\mathit{DL\text{-}Lite}_{bool}$, and we introduce two sublanguages of it, $\mathit{DL\text{-}Lite}_{krom}$ and $\mathit{DL\text{-}Lite}_{horn}$. Notably, the latter DL 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*.

We show that KB satisfiability (or subsumption w.r.t. a KB) is NLogSpace-complete for $\mathit{DL\text{-}Lite}_{krom}$, P-complete for $\mathit{DL\text{-}Lite}_{horn}$, and NP-complete (resp. coNP-complete) for $\mathit{DL\text{-}Lite}_{bool}$. We prove that data complexity of both satisfiability and instance checking is LogSpace for $\mathit{DL\text{-}Lite}_{bool}$. 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 $\mathit{DL\text{-}Lite}_{horn}$. Note that already for $\mathit{DL\text{-}Lite}_{krom}$, and hence also for $\mathit{DL\text{-}Lite}_{bool}$, it directly follows from previous results that the problem is coNP-complete [12, 10, 18].

The rest of the paper is structured as follows. In the next section we introduce the three variants of *DL-Lite* mentioned above, 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.

2 The *DL-Lite* family

We begin by considering the following extension $\mathit{DL\text{-}Lite}_{bool}$ of the description logic *DL-Lite* introduced in [8, 10]. The language of $\mathit{DL\text{-}Lite}_{bool}$ contains *object names* $a_0, a_1, \dots,$ *atomic concept names* $A_0, A_1, \dots,$ and *atomic role names* $P_0, P_1, \dots$. Complex *roles* R and *concepts* C of $\mathit{DL\text{-}Lite}_{bool}$ are defined as follows:

$$\begin{array}{lcl} R & ::= & P_i \quad | \quad P_i^-, \\ B & ::= & \bot \quad | \quad A_i \quad | \quad \geq q\, R, \\ C & ::= & B \quad | \quad \neg C \quad | \quad C_1 \sqcap C_2, \end{array}$$

where $q \geq 1$. Concepts of the form B will be called *basic*. A $\mathit{DL\text{-}Lite}_{bool}$ *TBox*, $\mathcal{T}$, consists of axioms of the form

$$C_1 \sqsubseteq C_2,$$

and an *ABox*, $\mathcal{A}$, of assertions of the form

$$A_k(a_i), \qquad P_k(a_i, a_j).$$

Together $\mathcal{T}$ and $\mathcal{A}$ constitute the *DL-Lite$_{bool}$ knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. (Note that assertions involving complex concepts $C(a_i)$ and inverse roles $P_k^-(a_i, a_j)$ can be expressed as

$$A_C(a_i), \qquad A_C \sqsubseteq C \qquad \text{and} \qquad P_k(a_j, a_i),$$

respectively, where A_C is a fresh atomic concept.)

A *DL-Lite$_{bool}$ interpretation* is a structure of the form

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

where Δ is a nonempty set, $a_i^{\mathcal{I}} \in \Delta$, $A_i^{\mathcal{I}} \subseteq \Delta$, $P_i^{\mathcal{I}} \subseteq \Delta \times \Delta$, and

(una) $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for all $i \neq j$.

The role and concept constructors are interpreted in $\mathcal{I}$ as usual:

$$\begin{aligned}
(P_i^-)^{\mathcal{I}} &= \{(y, x) \in \Delta \times \Delta \mid (x, y) \in P_i^{\mathcal{I}}\},\\
\bot^{\mathcal{I}} &= \emptyset,\\
(\geq q\, R)^{\mathcal{I}} &= \{x \in \Delta \mid \exists y_1, \dots, y_q \in \Delta \text{ such that } (x, y_i) \in R^{\mathcal{I}} \text{ and } y_i \neq y_j, \text{ for } i \neq j\},\\
(\neg C)^{\mathcal{I}} &= \Delta \setminus C^{\mathcal{I}},\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}.
\end{aligned}$$

The standard abbreviations $\top := \neg\bot$, $\exists R := (\geq 1\, R)$ and $\leq q\, R := \neg(\geq q + 1\, R)$ we need in what follows are self-explanatory and correspond to the intended semantics.

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

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

A *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is *satisfiable* if there is a model satisfying all the members of $\mathcal{T}$ and $\mathcal{A}$.

We also consider two sublanguages of *DL-Lite$_{bool}$*: the *Krom fragment*, *DL-Lite$_{krom}$*, and the *Horn fragment*, *DL-Lite$_{horn}$*.

(Krom) A TBox of a *DL-Lite$_{krom}$* KB only contains axioms of the following form (where the B_i are basic concepts):

$$B_1 \sqsubseteq B_2 \qquad \text{or} \qquad B_1 \sqsubseteq \neg B_2 \qquad \text{or} \qquad \neg B_1 \sqsubseteq B_2.$$

KBs with such TBoxes will be called *Krom KBs*.

(Horn) 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* [8, 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$_{bool}$* into the one-variable fragment of first-order logic

Our main aim in this section is to show that the satisfiability problem for *DL-Lite$_{bool}$* knowledge bases can be polynomially reduced to the satisfiability problem for the *one-variable fragment* $\mathcal{QL}^1$ of first-order logic without equality. (Recall that the satisfiability problem for $\mathcal{QL}^1$-formulas is NP-complete; see, e.g., [6].)

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a *DL-Lite$_{bool}$* knowledge base. Denote by $\mathit{role}(\mathcal{K})$ the set of atomic roles occurring in $\mathcal{T}$ and $\mathcal{A}$, by $\mathit{role}^\pm(\mathcal{K})$ the set $\{P_k, P_k^- \mid P_k \in \mathit{role}(\mathcal{K})\}$, and by $\mathit{ob}(\mathcal{A})$ the set of object names in $\mathcal{A}$. Let $q_\mathcal{T}$ be the maximum numerical parameter in $\mathcal{T}$. Note that if the functionality axiom $(\geq 2\,R \sqsubseteq \bot)$ is present in $\mathcal{T}$ then $q_\mathcal{T} \geq 2$.

With every object name $a_i \in \mathit{ob}(\mathcal{A})$ we associate the individual constant a_i of $\mathcal{QL}^1$ and with every concept name A_i the unary predicate $A_i(x)$ from the signature of $\mathcal{QL}^1$. For each role $R \in \mathit{role}^\pm(\mathcal{K})$, we also introduce $q_\mathcal{T}$ fresh unary predicates

$$E_qR(x), \qquad \text{for } 1 \leq q \leq q_\mathcal{T}.$$

The intended meaning of these predicates is as follows: for a role name P_k,

- $E_1P_k(x)$ and $E_1P_k^-(x)$ represent the domain and range of P_k, respectively; in other words, $E_1P_k(x)$ 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;
- $E_qP_k(x)$ 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 pair of roles $P_k, P_k^- \in \mathit{role}^\pm(\mathcal{K})$, we take two fresh individual constants

$$dp_k \qquad \text{and} \qquad dp_k^-$$

of $\mathcal{QL}^1$ which will serve as 'representatives' of the points from the domains of P_k and P_k^-, respectively (provided that they are not empty). Furthermore, for each pair of object names $a_i, a_j \in \mathit{ob}(\mathcal{A})$ and each role $R \in \mathit{role}^\pm(\mathcal{K})$, 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^*:

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

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

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

It should be also clear how to translate a *DL-Lite$_{bool}$* ABox $\mathcal{A}$ into $\mathcal{QL}^1$:

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

The following formulas express some natural properties of the role domains and ranges. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. For every role $R \in \mathit{role}^\pm(\mathcal{K})$, we need two $\mathcal{QL}^1$-sentences:

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

where

$$\mathit{inv}(E_q R) \ = \ \begin{cases} E_q P_k^-, & \text{if } R = P_k,\\ E_q P_k, & \text{if } R = P_k^-, \end{cases} \qquad \text{and} \qquad \mathit{inv}(dr) = \begin{cases} dp_k^-, & \text{if } R = P_k,\\ dp_k, & \text{if } R = P_k^-. \end{cases}$$

Sentence (4) says that if the domain of R is not empty then its range is not empty either: it contains the representative $\mathit{inv}(dr)$. The meaning of (5) should be obvious.

We also need formulas representing the relation of the $R a_i a_j$ with the unary predicates for the role domain and range. For a role $R \in \mathit{role}^\pm(\mathcal{K})$, let $R^\dagger$ be the conjunction of the $\mathcal{QL}^1$-sentences

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

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

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 the *DL-Lite$_{bool}$* knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we set

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

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

Theorem 1. *A DL-Lite$_{bool}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ 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., [13, 20]. 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 R, propositional variables p and constants a of $\mathcal{QL}^1$ in $\mathfrak{M}$ by $R^{\mathfrak{M}}$, $p^{\mathfrak{M}}$ and $a^{\mathfrak{M}}$, respectively.

Now we construct inductively a *DL-Lite*$_{bool}$ model $\mathcal{I}$ based on some domain $\Delta \supseteq D$. This domain Δ will be (inductively) defined as the union

$$\Delta \;=\; \bigcup_{m=0}^{\infty} W_m, \qquad \text{where} \quad W_0 \;=\; D.$$

The interpretations of the object names a in $\mathcal{I}$ are given by their interpretations in $\mathfrak{M}$, namely, $a^{\mathcal{I}} = a^{\mathfrak{M}} \in W_0$. 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. 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 is a function from Δ onto 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 $A^{\mathcal{I}}$ of concept names A in $\mathcal{I}$ are defined by taking

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

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

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

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

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

It follows from (5) that $r(R,d)$ is a function and that if $d \in D$ and $r(R,d) = q$ then we have $\mathfrak{M} \models E_{q'} R[d]$ whenever $1 \leq q' \leq q$, and $\mathfrak{M} \models \neg E_{q'} R[d]$ whenever $q < q' \leq q_{\mathcal{T}}$.

We also define the *actual R-rank $r_m(R,w)$ of a point $w \in \Delta$ at step m* by taking

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

where $R^m = P_k^m$ if $R = P_k$ and $R^m = (P_k^m)^-$ if $R = P_k^-$, $(P_k^m)^- = \{(w',w) \mid (w,w') \in P_k^m\}$ and, for $W \subseteq \Delta$, $R \subseteq \Delta \times \Delta$ and $0 \leq q \leq q_{\mathcal{T}}$,

$${\geq} q\, R.W \;=\; \bigl\{w \in W \mid \exists w_1, \dots, w_q \in W \text{ with } (w, w_i) \in R \text{ and } w_i \neq w_j \text{ for } i \neq j\bigr\}.$$

For the basis of induction we set, for each atomic role $P_k \in role(\mathcal{K})$,

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

Observe that, by (7) and (6), we have, for all $R \in role^{\pm}(\mathcal{K})$ 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 roles $R \in role^{\pm}(\mathcal{K})$ and points $w \in W_m$, then the model $\mathcal{I}$ we need would be constructed. 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 an atomic role $P_k \in role(\mathcal{K})$, consider the following two sets of defects in P_k^m:

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

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+1} and 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 $d = cp(w)$. We have $\mathfrak{M} \models E_{q'} P_k[d]$ for some $q' \geq q > 0$. Then, by (5), $\mathfrak{M} \models E_1 P_k[d]$ and, by (4), $\mathfrak{M} \models E_1 P_k^-[dp_k^-]$.[1] In this case we take q *fresh* copies $w'_1, \dots, w'_q$ of dp_k^- (and set $cp(w'_i) = dp_k^-$, for $1 \leq i \leq q$), add them to W_{m+1} and add the pairs (w, w'_i), $1 \leq i \leq q$, to P_k^{m+1}.

(Λ_k^{m-}) Let $w \in \Lambda_k^{m-}$, $q = r(P_k^-, cp(w)) - r_m(P_k^-, w)$ and $d = cp(w)$. We have $\mathfrak{M} \models E_{q'} P_k^-[d]$ 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 (and set $cp(w'_i) = dp_k$, for $1 \leq i \leq q$), add them to W_{m+1} and add the pairs (w'_i, w), $1 \leq i \leq q$, to P_k^{m+1}.

The reader can find a concrete example illustrating this construction in Fig. 1.

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

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

To prove this property, consider all possible cases:

- If $m < m_0$ then the point w has not been added to W_m yet, i.e., $w \notin W_m$, and so we have $r_m(R, w) = 0$.
- If $m = m_0$ and $m_0 = 0$ then $r_m(R, w) \leq r(R, cp(w))$ follows from (10).
- If $m = m_0$ and $m_0 > 0$ then w was added at step m_0 to cure a defect of some point $w' \in W_{m_0-1}$. This means that there is $P_k \in role(\mathcal{K})$ such that either $(w', w) \in P_k^{m_0}$ and $w' \in \Lambda_k^{m_0-1}$ or $(w, w') \in P_k^{m_0}$ and $w' \in \Lambda_k^{(m_0-1)-}$. Consider the former case. We have $cp(w) = dp_k^-$. Since *fresh* witnesses are picked up every time the rule ($\Lambda_k^{m_0-1}$) is applied, $r_{m_0}(P_k^-, w) = 1$, $r_{m_0}(P_k, w) = 0$ and $r_{m_0}(R, w) = 0$, for every $R \neq P_k, P_k^-$. So it suffices to show that $r(P_k^-, dp_k^-) \geq 1$. Indeed, as $\mathfrak{M} \models E_q P_k[cp(w')]$ for some $q \geq 1$, we have, by (5), $\mathfrak{M} \models E_1 P_k[cp(w')]$, and so, by (4), $\mathfrak{M} \models E_1 P_k^-[dp_k^-]$. By the definition of r, we have $r(P_k^-, dp_k^-) \geq 1$. The latter case is considered analogously.

[1] Here and below we slightly abuse notation and write dp_k^- instead of $(dp_k^-)^{\mathfrak{M}}$.

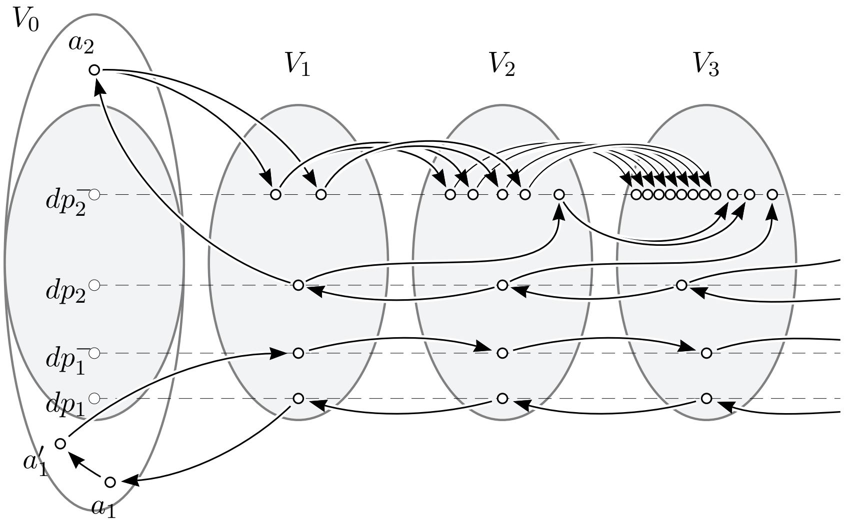

Figure 1: The lower half of the figure shows part of the constructed model for $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ with $\mathcal{T}_1 = \{A_1 \sqsubseteq \exists P_1, A_1 \sqsubseteq \exists P_1^-, \top \sqsubseteq \leq 1\, P_1, \top \sqsubseteq \leq 1\, P_1^-, \exists P_1 \sqsubseteq A_1, \exists P_1^- \sqsubseteq A_1\}$ and $\mathcal{A}_1 = \{A_1(a_1), A_1(a_1'), P_1(a_1, a_1')\}$. The upper half shows part of the model for $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A}_2)$ with $\mathcal{T}_2 = \{A_2 \sqsubseteq \exists P_2^-, A_2 \sqsubseteq \geq 2\, P_2, \top \sqsubseteq \leq 1\, P_2^-, \exists P_2 \sqsubseteq A_2, \exists P_2^- \sqsubseteq A_2\}$ and $\mathcal{A}_2 = \{A_2(a_2)\}$. The full model for $\mathcal{K}_1 \cup \mathcal{K}_2$ consists of three isomorphic copies of the depicted part: one (depicted) is built around the a_i, and the other two are built around the dp_i and the dp_i^-, respectively.

- If $m = m_0 + 1$ then, for each role name P_k, all defects of w are cured at step $m_0 + 1$ by applying the rules $(\Lambda_k^{m_0})$ and $(\Lambda_k^{m_0-})$. Therefore, $r_{m_0+1}(R, w) = r(R, cp(w))$.
- If $m > m_0 + 1$ then (11) follows from the observation that new arrows involving w can only be added at step $m_0 + 1$, that is, for all $m \geq 0$, and each *role name* $P_k \in role(\mathcal{K})$,

$$P_k^{m+1} \setminus P_k^m \quad \subseteq \quad V_m \times V_{m+1} \quad \cup \quad V_{m+1} \times V_m. \tag{12}$$

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

$$\mathfrak{M} \models E_q R[cp(w)] \qquad \text{iff} \qquad w \in\, \geq q\, R^{\mathcal{I}}.\Delta. \tag{13}$$

Indeed, if $\mathfrak{M} \models E_q R[cp(w)]$ then, by definition, $r(R, cp(w)) \geq q$. Let $w \in V_{m_0}$. Then, by (11), $r_m(R, w) = r(R, cp(w)) \geq q$, for all $m > m_0$. It follows from the definition of $r_m(R, w)$ and $R^{\mathcal{I}}$ that $w \in (\geq q\, R^{\mathcal{I}}.\Delta)$. Conversely, let $w \in (\geq q\, R^{\mathcal{I}}.\Delta)$ and $w \in V_{m_0}$. Then, by (11), $q \leq r_m(R, w) = r(R, cp(w))$, for all $m > m_0$. So, by the definition of $r(R, cp(w))$ and (5), $\mathfrak{M} \models E_q R[cp(w)]$.

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

$$\mathfrak{M} \models C^*[cp(w)] \qquad \text{iff} \qquad w \in C^{\mathcal{I}}. \tag{14}$$

The basis of induction is trivial for $B = \bot$ and follows from (8) for $B = A_i$ and from (13) for $B = \geq q\, R$. The induction step for the Booleans ($C = \neg C_1$ and $C = C_1 \sqcap C_2$) follows from the induction hypothesis.

Finally, we show that for each statement $\psi \in \mathcal{T} \cup \mathcal{A}$,

$$\mathfrak{M} \models \psi^\dagger \qquad \text{iff} \qquad \mathcal{I} \models \psi. \tag{15}$$

The case $\psi = C_1 \sqsubseteq C_2$ follows from (14) 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 (12), $(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^0$ iff, by (9), $\mathfrak{M} \models P_k a_i a_j$.

Therefore $\mathcal{I} \models \mathcal{K}$.

($\Rightarrow$) Let $\mathcal{I}$ be an interpretation of the form (1) such that $\mathcal{I} \models \mathcal{K}$. We construct a model $\mathfrak{M}$ for $\mathcal{K}^\dagger$ based on the same domain Δ as $\mathcal{I}$. For every $a_i \in ob(\mathcal{A})$, we let $a_i^{\mathfrak{M}} = a_i^{\mathcal{I}}$ and, for every $R \in role^{\pm}(\mathcal{K})$, we take some $d \in (\geq 1\, R)^{\mathcal{I}}$ if $(\geq 1\, R)^{\mathcal{I}} \neq \emptyset$ and an arbitrary element $d \in \Delta$ otherwise, and let

$$dr^{\mathfrak{M}} \;=\; d.$$

Next, for every concept name A_i, we let $A_i^{\mathfrak{M}} = A_i^{\mathcal{I}}$ and, for every role $R \in role^{\pm}(\mathcal{K})$, we put

$$E_q R^{\mathfrak{M}} \;=\; (\geq q\, R)^{\mathcal{I}}.$$

Finally, for every role name $P_k \in role(\mathcal{K})$ and every pair of objects $a_i, a_j \in ob(\mathcal{A})$, we define $(P_k a_i a_j)^{\mathfrak{M}}$ to be true iff $\mathcal{I} \models P_k(a_i, a_j)$.

It is readily checked that $\mathfrak{M} \models \mathcal{K}^\dagger$. $\square$

The translation $\mathcal{K}^\dagger$ of $\mathcal{K}$ is obviously too lengthy to provide us with reasonably low complexity results. However, it follows from the proof above that in fact a lot of information in this translation is redundant and can be safely omitted.

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 set of 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

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

where $\varepsilon(R)$ is as before (see (4)),

$$\delta^\flat(R) \;=\; \bigwedge_{\substack{q,q' \in Q_{\mathcal{T}}^R \\ q' > q \\ q' > q'' > q \text{ for no } q'' \in Q_{\mathcal{T}}^R}} \forall x \big(E_{q'} R(x) \to E_q R(x)\big), \tag{16}$$

(cf. (5)), and

$$\mathcal{A}^\flat \;=\; \bigwedge_{A(a_i) \in \mathcal{A}} A(a_i) \qquad \wedge \bigwedge_{\substack{R \in role^{\pm}(\mathcal{K}) \\ a \in ob(\mathcal{A})}} E_{q_{R,a}} R(a), \tag{17}$$

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 $R = P_k^-$. 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.*

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

Proof. Follows from the fact that $\mathcal{K}^\dagger$ is satisfiable iff $\mathcal{K}^\flat$ is satisfiable. Indeed, if $\mathfrak{M} \models \mathcal{K}^\dagger$ then clearly $\mathfrak{M} \models \mathcal{K}^\flat$. Conversely, if $\mathfrak{M} \models \mathcal{K}^\flat$ then one can construct a new model $\mathfrak{M}'$ based on the same domain D as $\mathfrak{M}$ by taking

- $A^{\mathfrak{M}'} = A^{\mathfrak{M}}$, for all concept names A;
- $E_q R^{\mathfrak{M}'} = E_{q'} R^{\mathfrak{M}}$, for all $R \in \mathit{role}^\pm(\mathcal{K})$ and $1 \le q \le q_\mathcal{T}$, where q' is the maximum number from $Q^R_\mathcal{T}$ with $q' \le q$;
- Ra_ia_j to be true in $\mathfrak{M}'$ iff $R(a_i, a_j) \in \mathcal{A}$ or $\mathit{inv}(R)(a_j, a_i) \in \mathcal{A}$;
- $a^{\mathfrak{M}'} = a^{\mathfrak{M}}$, for all $a \in \mathit{ob}(\mathcal{A})$;
- $dr^{\mathfrak{M}'} = dr^{\mathfrak{M}}$, for all $R \in \mathit{role}^\pm(\mathcal{K})$.

It follows immediately from the definition that we have $\mathfrak{M}' \models \mathcal{K}^\dagger$. (For example, $\mathfrak{M}' \models \mathcal{T}^*$ follows from the fact that for every concept $(\geq q\, R)$ from $\mathcal{T}$ we have $E_q R^{\mathfrak{M}'} = E_q R^{\mathfrak{M}}$.) □

Remark 3. Note that the cardinality (and functionality) constraints from $\mathcal{K}^\flat$ are only checked for the named objects. Since $\mathcal{K}^\flat$ only contains *unary* predicates and, unlike $\mathcal{K}^\dagger$, does not contain propositional variables Ra_ia_j, the actual connections between named objects (stated in the ABox) are of no importance at all; what really matters is the unary 'types' of named objects $a \in \mathit{ob}(\mathcal{A})$, 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 $\mathcal{K}$.

As an immediate consequence of Corollary 2 and the fact that the satisfiability problem for $\mathcal{QL}^1$-formulas is NP-complete (and that *DL-Lite*$_{bool}$ contains the Booleans, and so can encode full propositional classical logic), we obtain the following:

Theorem 4. *The satisfiability problem for DL-Lite*$_{bool}$ *knowledge bases is* NP*-complete.*

Let us now observe that if $\mathcal{K}$ is a Krom KB then $\mathcal{K}^\flat$ belongs to the Krom fragment[2] of $\mathcal{QL}^1$. As the satisfiability problem for Krom formulas with the prefix of the form $\forall x$ (as in $\mathcal{K}^\flat$) is NLogSpace-complete (see, e.g., [6, Exercise 8.3.7]), we obtain the following:

Theorem 5. *The satisfiability problem for Krom knowledge bases is* NLogSpace*-complete.*

If $\mathcal{K}$ is a Horn KB then $\mathcal{K}^\flat$ belongs to the universal Horn fragment of $\mathcal{QL}^1$.

Theorem 6. *The satisfiability problem for Horn knowledge bases is* P*-complete.*

Proof. As $\mathcal{QL}^1$ contains no function symbols and $\mathcal{K}^\flat$ is universal, satisfiability of $\mathcal{K}^\flat$ is polynomially reducible to satisfiability of a set of propositional Horn formulas, namely, the formulas that are obtained from $\mathcal{K}^\flat$ by replacing x with each of the constants occurring in $\mathcal{K}^\flat$. It remains to recall that the satisfiability problem for propositional Horn formulas is P-complete (see, e.g., [19]). □

[2]The Krom fragment consists of all formulas in prenex normal form whose quantifier-free part is a conjunction of binary clauses.

As is well known, many other reasoning tasks for description logics are reducible to the satisfiability problem. Consider, for example, the *subsumption problem*: given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and two concepts C and D, decide whether $\mathcal{K} \models C \sqsubseteq D$, that is, we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{K}$. To reduce this problem to satisfiability, we take some fresh atomic concept A, an object name a, and set

$$\mathcal{K}' = (\mathcal{T}', \mathcal{A}'), \quad \text{where} \quad \mathcal{T}' = \mathcal{T} \cup \{A \sqsubseteq C,\ A \sqsubseteq \neg D\}, \quad \mathcal{A}' = \mathcal{A} \cup \{A(a)\}.$$

It is easy to see that $\mathcal{K} \models C \sqsubseteq D$ iff $\mathcal{K}'$ is *not* satisfiable. It follows that the subsumption problem for *DL-Lite*$_{bool}$ knowledge bases is coNP-complete. In the case of Krom knowledge bases we should assume that $C \sqsubseteq D$ belongs to the Krom fragment, and so $\mathcal{T}'$ is a Krom KB as well. By the Immerman–Szelepcsényi theorem, NLogSpace = coNLogSpace, and so the subsumption problem for Krom KBs is NLogSpace-complete. For Horn KBs, $C \sqsubseteq D$ should be a Horn subsumption of the form $\bigsqcap_{k=1}^{n} B_k \sqsubseteq B$. In this case we slightly change the reduction above by taking

$$\mathcal{T}' = \mathcal{T} \cup \{A \sqsubseteq B_1, \dots, A \sqsubseteq B_n, A \sqcap B \sqsubseteq \bot\}, \quad \mathcal{A}' = \mathcal{A} \cup \{A(a)\}.$$

Again we have $\mathcal{K} \models C \sqsubseteq D$ iff $\mathcal{K}'$ is not satisfiable. This means that the subsumption problem for Horn KBs is P-complete. Other reasoning tasks are analysed in the same way (a reduction for the instance checking problem can be found in the next section, and query answering will be considered in Section 5).

So far we have assumed the whole knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ to be the input for the satisfiability problem (together with the concepts C and D in the case of the subsumption problem). According to the classification suggested by Vardi [22], we have been considering the *combined complexity* of the satisfiability problem. Two other types of complexity for knowledge bases are

- the *program* (or *TBox*) *complexity*, where only the TBox $\mathcal{T}$ is regarded to be the input, while the ABox $\mathcal{A}$ is assumed to be fixed, and

- the *data* (or *ABox*) *complexity*, where the knowledge in the TBox $\mathcal{T}$ is fixed, while the input data ABox $\mathcal{A}$ can vary.

It is easy to see that the program complexity of the satisfiability and subsumption problems for *DL-Lite*$_{bool}$ and its fragments considered above coincides with the corresponding combined complexity.

Let us consider first the data complexity.

4 Data complexity

In this section we show that as far as data complexity is concerned, reasoning problems for *DL-Lite*$_{bool}$ knowledge bases can be solved using only logarithmic space in the size of the ABox. We remind the reader (for more details see, e.g., [16]) 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 knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ occur in its TBox and write $role^{\pm}(\mathcal{T})$ instead of $role^{\pm}(\mathcal{K})$. Let

$$\Sigma(\mathcal{T}) \;=\; \{E_1R(dr) \mid R \in role^{\pm}(\mathcal{T})\},$$

and, for $\Sigma_0 \subseteq \Sigma(\mathcal{T})$, let

$$core_{\Sigma_0}(\mathcal{T}) = \bigwedge_{E_1R(dr)\in\Sigma_0} E_1R(dr) \quad\wedge \bigwedge_{R\in role^{\pm}(\mathcal{T})} \Big(\mathcal{T}^*[dr] \;\wedge\; \bigwedge_{R'\in role^{\pm}(\mathcal{T})} \big(\varepsilon(R')[dr] \;\wedge\; \delta^{\flat}(R')[dr]\big)\Big), \quad (18)$$

$$proj_{\Sigma_0}(\mathcal{K}, a) = \bigwedge_{inv(E_1R)(inv(dr))\in\Sigma(\mathcal{T})\setminus\Sigma_0} \neg E_1R(a) \quad\wedge\quad \mathcal{T}^*[a] \;\wedge \bigwedge_{R'\in role^{\pm}(\mathcal{T})} \delta^{\flat}(R')[a] \quad\wedge\quad \mathcal{A}^{\flat}(a), \quad (19)$$

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 7. *$\mathcal{K}^{\flat}$ is satisfiable iff there is a subset Σ_0 of $\Sigma(\mathcal{T})$ such that*

- *$core_{\Sigma_0}(\mathcal{T})$ is satisfiable;*
- *$proj_{\Sigma_0}(\mathcal{K}, a)$ is satisfiable for every $a \in ob(\mathcal{A})$.*

Proof. ($\Rightarrow$) If there is $\mathfrak{M}$ such that $\mathfrak{M} \models \mathcal{K}^{\flat}$, then we take

$$\Sigma_0 \;=\; \{E_1R(dr) \mid R \in role^{\pm}(\mathcal{T}),\; \mathfrak{M} \models E_1R[dr]\}.$$

It should be clear that we have $\mathfrak{M} \models core_{\Sigma_0}(\mathcal{T})$ and $\mathfrak{M} \models proj_{\Sigma_0}(\mathcal{K}, a)$, for all $a \in ob(\mathcal{A})$.

($\Leftarrow$) Conversely, let $\mathfrak{M}_{\Sigma_0}$ be an Herbrand model of $core_{\Sigma_0}(\mathcal{T})$ and let $\mathfrak{M}_a$ be an Herbrand model of $proj_{\Sigma_0}(\mathcal{K}, a)$, for $a \in ob(\mathcal{A})$. By definition, the domain of $\mathfrak{M}_{\Sigma_0}$ consists of $|role^{\pm}(\mathcal{T})|$ elements and the domains of the $\mathfrak{M}_a$ are singletons. It should be clear that

- $\mathfrak{M}_{\Sigma_0} \models \mathcal{T}^*$ and $\mathfrak{M}_{\Sigma_0} \models \varepsilon(R) \wedge \delta^{\flat}(R)$, for every $R \in role^{\pm}(\mathcal{T})$;
- for every $a \in ob(\mathcal{A})$, we have $\mathfrak{M}_a \models \mathcal{T}^*$ and $\mathfrak{M}_a \models \delta^{\flat}(R)$ for every $R \in role^{\pm}(\mathcal{T})$.

We construct a model $\mathfrak{M}$ by taking the disjoint union of $\mathfrak{M}_{\Sigma_0}$ with all of the $\mathfrak{M}_a$. Let us show that $\mathfrak{M} \models \mathcal{K}^{\flat}$. Indeed,

- We have $\mathfrak{M} \models \mathcal{T}^*$ because $\mathcal{T}^*$ is universal, does not contain constants and is true in every component model.
- By the same argument we have $\mathfrak{M} \models \delta^{\flat}(R)$, for each role $R \in role^{\pm}(\mathcal{T})$.
- Consider now $\varepsilon(R) = \forall x\, \psi(x)$, where $\psi(x) = (E_1R(x) \to inv(E_1R)(inv(dr)))$. We show that, for every d in the domain of $\mathfrak{M}$, we have $\mathfrak{M} \models \psi[d]$. If d is of the form ${dr'}^{\mathfrak{M}}$, for some role $R' \in role^{\pm}(\mathcal{T})$, then clearly $\mathfrak{M} \models \psi[d]$, since $\mathfrak{M}_{\Sigma_0} \models \varepsilon(R)$. If d is of the form $a^{\mathfrak{M}}$, for $a \in ob(\mathcal{A})$, then it trivially holds if $\mathfrak{M}_a \not\models E_1R(a)$. Otherwise, $\mathfrak{M}_a \models E_1R(a)$, and so $inv(E_1R)(inv(dr)) \notin \Sigma(\mathcal{T}) \setminus \Sigma_0$. Therefore, $\mathfrak{M} \models inv(E_1R)(inv(dr))$ and $\mathfrak{M} \models \psi[d]$.
- Finally, $\mathcal{A}^{\flat}$ holds true because every conjunct $B^*(a)$ of it is true in the respective component model $\mathfrak{M}_a$, and so in $\mathfrak{M}$ as well.

This completes the proof of the lemma. □

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 $DL\text{-}Lite_{bool}$ 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 its ABox. This observation suggests a high degree of parallelism in the satisfiability check.

Theorem 8. *The data complexity of the satisfiability and instance checking problems for $DL\text{-}Lite_{bool}$ knowledge bases is in* LogSpace.

Proof. The instance checking problem is reducible to the (un)satisfiability problem: an object a is an instance of an atomic concept B in every model of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff the knowledge base

$$\big(\mathcal{T} \cup \{A_{\neg B} \sqsubseteq \neg B\},\ \mathcal{A} \cup \{A_{\neg B}(a)\}\big)$$

is not satisfiable, where $A_{\neg B}$ is a fresh concept name.

The following deterministic algorithm checks whether a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfiable:

- for every subset Σ_0 of $\Sigma(\mathcal{T})$, we do the following:
  - **(c)** compute $core_{\Sigma_0}(\mathcal{T})$ and check whether it is satisfiable;
  - **(p)** for every object name $a \in ob(\mathcal{A})$,
    - ∗ compute the $q_{R,a}$, for $R \in role^{\pm}(\mathcal{T})$,
    - ∗ compute $proj_{\Sigma_0}(\mathcal{K}, a)$ and check whether it is satisfiable.

The above deterministic algorithm requires space bounded by a logarithmic function in the size $|\mathcal{A}|$ of ABox. Indeed, in order to enumerate all subsets Σ_0 of $\Sigma(\mathcal{T})$ one needs $|role^{\pm}(\mathcal{T})|$ cells of the work tape—this does not depend on $|\mathcal{A}|$. At step **(c)**, the size of $core_{\Sigma_0}(\mathcal{T})$ 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 $ob(\mathcal{A})$, and this requires $\log |\mathcal{A}|$ cells on the working tape. Next, the $q_{R,a}$, for $R \in role^{\pm}(\mathcal{T})$, can be computed using

$$q_{\mathcal{T}} \cdot \log |\mathcal{A}|$$

of extra space: for every role $R \in role^{\pm}(\mathcal{T})$ and every q, $1 \leq q \leq q_{\mathcal{T}}$, one enumerates all q-tuples $(a_{i_1}, \dots, a_{i_q})$ of distinct objects in $ob(\mathcal{A})$ 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. (17)). Finally, for each $a \in ob(\mathcal{A})$, the size of $proj_{\Sigma_0}(\mathcal{K}, a)$ does not depend on $|\mathcal{A}|$ and its satisfiability can be checked deterministically.

The above calculations show that the algorithm needs $const \cdot \log |\mathcal{A}|$ cells on the working tape, where $const$ does not depend on $|\mathcal{A}|$. □

In fact, the algorithm provided in the proof above shows that the satisfiability and instance checking problems for $DL\text{-}Lite_{bool}$ knowledge bases belong to the parallel complexity class AC_0 (see, e.g., [19]).

5 Query answering

By a *positive existential query* $q(x_1, \dots, x_n)$ we mean any first-order formula

$$\varphi(x_1, \dots, x_n) \tag{20}$$

constructed by means of conjunction, disjunction and existential quantification form atoms of the from $A(t)$ and $P(t_1, t_2)$, where A is a concept name, P a role name, and t, t_1, t_2 are *terms* taken from the list of variables $y_0, y_1, \dots$ and the list of object names $a_0, a_1, \dots$. More precisely,

$$\begin{array}{rcl} t & ::= & y_i \quad | \quad a_i, \\ \varphi & ::= & A_i(t) \quad | \quad P_i(t_1, t_2) \quad | \quad \varphi_1 \wedge \varphi_2 \quad | \quad \varphi_1 \vee \varphi_2 \quad | \quad \exists y_i\, \varphi. \end{array}$$

The free variables of φ are called *distinguished variables* of q and the bound ones are *non-distinguished variables* of q. We write $q(x_1, \dots, x_n)$ for a query with distinguished variables $x_1, \dots, x_n$.

A *conjunctive query* is a positive existential query which contains no disjunction (that is, it is constructed from atoms by means of conjunction and existential quantification).

Given a query $q(\vec{x}) = \varphi(\vec{x})$ with $\vec{x} = x_1, \dots, x_n$ 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 $\varphi(\vec{x})$ with the ith member of $\vec{a}$. Queries containing no distinguished variables will be called *ground* (sentences).

Let $\mathcal{I}$ be a *DL-Lite$_{bool}$* interpretation of the form (1). An *assignment* $\mathfrak{a}$ in Δ (for the query $q(\vec{x})$) is a function associating with every non-distinguished 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 $y^{\mathcal{I},\mathfrak{a}} = \mathfrak{a}(y)$. Next, we define the *satisfaction relation* for positive existential formulas with respect to a given assignment $\mathfrak{a}$:

$$\begin{array}{rcl} \mathcal{I} \models^{\mathfrak{a}} A(t) & \text{iff} & t^{\mathcal{I},\mathfrak{a}} \in A^{\mathcal{I}}, \\ \mathcal{I} \models^{\mathfrak{a}} P(t_1, t_2) & \text{iff} & (t_1^{\mathcal{I},\mathfrak{a}}, t_2^{\mathcal{I},\mathfrak{a}}) \in P^{\mathcal{I}}, \\ \mathcal{I} \models^{\mathfrak{a}} \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{I} \models^{\mathfrak{a}} \varphi_1 \text{ and } \mathcal{I} \models^{\mathfrak{a}} \varphi_2, \\ \mathcal{I} \models^{\mathfrak{a}} \varphi_1 \vee \varphi_2 & \text{iff} & \mathcal{I} \models^{\mathfrak{a}} \varphi_1 \text{ or } \mathcal{I} \models^{\mathfrak{a}} \varphi_2, \\ \mathcal{I} \models^{\mathfrak{a}} \exists y_i\, \varphi_1 & \text{iff} & \mathcal{I} \models^{\mathfrak{b}} \varphi_1, \text{ for some assignment } \mathfrak{b} \text{ in } \Delta \text{ that may differ from } \mathfrak{a} \text{ on } y_i. \end{array}$$

For a ground query $q(\vec{a}) = \varphi(\vec{a})$, we write $\mathcal{K} \models q(\vec{a})$ if $\mathcal{I} \models^{\mathfrak{a}} \varphi(\vec{a})$, for every model $\mathcal{I}$ of $\mathcal{K}$ (with domain Δ) and every (some) assignment $\mathfrak{a}$ in Δ.

Given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that a tuple $\vec{a}$ of object names from $ob(\mathcal{A})$ 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 *answer* to the query $q(\vec{x})$ with respect to $\mathcal{K}$ is the set

$$\{\vec{a} = (a_{i_1}, \dots, a_{i_n}) \in (ob(\mathcal{A}))^n \mid \mathcal{K} \models q(\vec{a})\}.$$

The *query answering problem* we analyse in this section is formulated as follows: given a *DL-Lite$_{bool}$* knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a query $q(\vec{x})$, and a tuple $\vec{a}$ of object names from $ob(\mathcal{A})$, decide whether $\mathcal{K} \models q(\vec{a})$. Another variant of the query answering problem 'list all the answers $\vec{a}$ to $q(\vec{x})$ with respect to $\mathcal{K}$' is known to be logspace equivalent to previous one; see, e.g., [1, Exercise 16.13]. We distinguish between two cases of the query answering problem: the general case of positive existential queries and the case of conjunctive queries. Here we are interested in the *data complexity* of the query answering problem, that is, the TBox $\mathcal{T}$ and the query $q(\vec{x})$ are assumed to be fixed.

Let us begin with the following well-known result:

Theorem 9 ([12])**.** *The conjunctive query answering problem for Krom knowledge bases is data-hard for* coNP.

Proof. We reduce the complement of the NP-complete satisfiability problem for 2+2 CNFs to the query answering in Krom knowledge bases (see [21]). Let $x_1, \dots, x_k$ be Boolean variables and f be a 2+2 CNF, that is, a formula of the form $\bigwedge_{j=1}^{n} D_j$, where $D_j = x_{i_j^1} \vee x_{i_j^2} \vee \neg x_{i_j^3} \vee \neg x_{i_j^4}$ $(1 \leq i_j^1, i_j^2, i_j^3, i_j^4 \leq k)$, for all $1 \leq j \leq n$. Denote the following ABox by $\mathcal{A}_f$:

$$\big\{S_1(d_j, x_{i_j^1}), S_2(d_j, x_{i_j^2}), S_3(d_j, x_{i_j^3}), S_4(d_j, x_{i_j^4}) \mid 1 \leq j \leq n\big\}.$$

Let $\mathcal{T} = \{T \sqsubseteq \neg F, \ \ \neg F \sqsubseteq T\}$. Then f is unsatisfiable iff the query

$$\exists y \Big(\exists y_1 \, (S_1(y, y_1) \wedge F(y_1)) \ \wedge \ \exists y_2 \, (S_2(y, y_2) \wedge F(y_2)) \ \wedge \\ \exists y_3 \, (S_3(y, y_3) \wedge T(y_3)) \ \wedge \ \exists y_4 \, (S_1(y, y_4) \wedge T(y_4))\Big)$$

has an answer with respect to $(\mathcal{T}, \mathcal{A}_f)$. □

The matching upper bound follows from Theorem 14 below.

Theorem 10. *The data complexity of the positive existential query answering problem for Horn knowledge bases is in* LogSpace.

Proof. The plan of the proof is as follows:

- First, we show in Lemma 11 how to 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 knowledge base $\mathcal{T}$.
- Second, we show in Lemma 12 and Corollary 13 that, actually, 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, $\vec{y}$, variables in the given query as well as the size of $\mathcal{T}$).
- The LogSpace query answering algorithm will consider then all proper possible assignments of elements in that finite part of $\mathcal{I}_0$ to the variables $\vec{x}$, $\vec{y}$, compute the corresponding types (the concepts that contain these elements), and, finally, evaluate the query.

Suppose that we have a *consistent* Horn knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ (with all its concept and role names occurring in the TBox $\mathcal{T}$) and a positive existential query $q(\vec{x})$ of the form (20). Let $\mathfrak{M}_0$ be the *minimal Herbrand model* for $\mathcal{K}^\flat$. We remind the reader (for details consult, e.g., [2, 20]) that $\mathfrak{M}_0$ can be constructed by taking the intersection of all Herbrand models for $\mathcal{K}^\flat$, that is, of all models based on the domain Λ_0 which consists of all constant symbols from $\mathcal{K}^\flat$, i.e.,

$$\Lambda_0 \ = \ ob(\mathcal{A}) \ \cup \ \{dr \mid R \in role^{\pm}(\mathcal{T})\}.$$

Another way of constructing $\mathfrak{M}_0$ is to apply the following procedure to $\mathcal{K}^\flat$. Let $\beta(\mathcal{T})$ be the set of atomic concept names in $\mathcal{T}$ together with the $\geq q\, R$, for $q \in Q_{\mathcal{T}}^{R}$ and $R \in role^{\pm}(\mathcal{T})$ (cf. concepts of the form B on p. 2). Then the set of all predicates in $\mathcal{K}^\flat$ is $\{B^* \mid B \in \beta(\mathcal{T})\}$. Denote by $\Sigma_{\mathcal{K}}$ the set of all conjuncts of $\mathcal{K}^\flat$, that is, the set of all subformulas of $\mathcal{K}^\flat$ of the form $\forall x \, \big(B_1^*(x) \wedge \dots \wedge B_k^*(x) \to B^*(x)\big)$, $\forall x \, \big(B_1^*(x) \wedge \dots \wedge B_k^*(x) \to \bot\big)$, $\forall x \, \big(B_1^*(x) \to B^*(dr)\big)$ and $B^*(c)$, where $B_1, \dots, B_k, B \in \beta(\mathcal{T})$, $dr \in \Lambda_0$ for $R \in role^{\pm}(\mathcal{T})$ and $c \in \Lambda_0$.

Step 0. For every $c \in \Lambda_0$, set

$$\boldsymbol{t}^0(c) \;=\; \big\{B \in \beta(\mathcal{T}) \mid B^*(c) \in \Sigma_{\mathcal{K}}\big\}.$$

Step $n+1$. Suppose we have already defined $\boldsymbol{t}^n(c)$ for all $c \in \Lambda_0$ and $n \geq 0$. Then we set

$$\begin{aligned}\boldsymbol{t}^{n+1}(c) \;=\; & \boldsymbol{t}^n(c)\\ & \cup \big\{B \in \beta(\mathcal{T}) \mid \forall x\,(B_1^*(x) \wedge \cdots \wedge B_k^*(x) \to B^*(x)) \in \Sigma_{\mathcal{K}}\\ & \qquad\qquad\qquad\qquad\qquad\qquad \text{and } B_1, \ldots, B_k \in \boldsymbol{t}^n(c)\big\}\\ & \cup \big\{B \in \beta(\mathcal{T}) \mid \forall x\,(B_1^*(x) \to B^*(c)) \in \Sigma_{\mathcal{K}} \text{ and } B_1 \in \boldsymbol{t}^n(c') \text{ for some } c' \in \Lambda_0\big\}.\end{aligned}$$

Clearly, $\boldsymbol{t}^n(c) \subseteq \boldsymbol{t}^{n+1}(c)$, for each $c \in \Lambda_0$. Moreover, after finitely many, say m, steps we will have $\boldsymbol{t}^m(c) = \boldsymbol{t}^{m+1}(c)$. Denote this maximal $\boldsymbol{t}^m(c)$ by $\boldsymbol{t}^{\mathfrak{M}_0}(c)$. Finally, for each $B \in \beta(\mathcal{T})$, we set $(B^*)^{\mathfrak{M}_0} = \big\{c \in \Lambda_0 \mid B \in \boldsymbol{t}^{\mathfrak{M}_0}(c)\big\}$. Recall that we have assumed $\mathcal{K}$ (and so $\mathcal{K}^\flat$) to be consistent. This means that all conjuncts of $\mathcal{K}^\flat$ of the form $\forall x\,(B_1^*(x) \wedge \cdots \wedge B_k^*(x) \to \bot)$ are automatically satisfied in $\mathfrak{M}_0$. Therefore, $\mathfrak{M}_0 \models \mathcal{K}^\flat$ for the resulting model $\mathfrak{M}_0$, which is obviously the minimal Herbrand model for $\mathcal{K}^\flat$.

Now we convert $\mathfrak{M}_0$ into the model for $\mathcal{K}^\dagger$ as in the proof of Corollary 2, and then apply to it the unravelling procedure described in the proof of Theorem 1. Let $\mathcal{I}_0$ be the resulting model of $\mathcal{K}$ with the domain Δ_0. Given a model $\mathcal{J}$ and a point w in its domain, we let

$$\boldsymbol{t}^{\mathcal{J}}(w) \;=\; \{B \in \beta(\mathcal{T}) \mid w \in B^{\mathcal{J}}\}.$$

The properties of the model $\mathcal{I}_0$ we need in this proof are as follows: for every model $\mathcal{J}$ of $\mathcal{K}$ with domain Γ, we have:

(ext$_a$) for every $a \in ob(\mathcal{A})$, $\boldsymbol{t}^{\mathcal{I}_0}(a^{\mathcal{I}_0}) \subseteq \boldsymbol{t}^{\mathcal{J}}(a^{\mathcal{J}})$;

(ext$_{dr}$) for every $w \in \Delta_0 \setminus \{a^{\mathcal{I}_0} \mid a \in ob(\mathcal{A})\}$,

- if $w \in (\geq 1\, R)^{\mathcal{I}_0}$, for some role $R \in \mathit{role}^{\pm}(\mathcal{T})$, then $\boldsymbol{t}^{\mathcal{I}_0}(w) \subseteq \boldsymbol{t}^{\mathcal{J}}(w')$, for every $w' \in (\geq 1\, R)^{\mathcal{J}}$;
- otherwise, $\boldsymbol{t}^{\mathcal{I}_0}(w) \subseteq \boldsymbol{t}^{\mathcal{J}}(w')$ for every $w' \in \Gamma$;

(ext-e$_{dr}$) for each role $R \in \mathit{role}^{\pm}(\mathcal{T})$, if $(\geq 1\, R)^{\mathcal{I}_0} \neq \emptyset$ then $(\geq 1\, R)^{\mathcal{J}} \neq \emptyset$.

These properties follow immediately from the fact that $\mathfrak{M}_0$ is the minimal Herbrand model for $\mathcal{K}^\flat$.

Lemma 11. *For every positive existential sentence ψ, $\mathcal{K} \models \psi$ iff $\mathcal{I}_0 \models \psi$.*

Proof. The implication ($\Rightarrow$) is trivial. To show ($\Leftarrow$), consider an arbitrary model $\mathcal{J}$ of $\mathcal{K}$ based on some domain Γ. We have to prove that

$$\mathcal{J} \models \psi \quad \text{whenever} \quad \mathcal{I}_0 \models \psi.$$

As ψ is a positive existential sentence, it is enough to construct a homomorphism $f : \mathcal{I}_0 \to \mathcal{J}$. We do this by induction on the construction of $\Delta_0 = \bigcup_{m=0}^{\infty} W_m$ (see the proof of Theorem 1). More precisely, f is defined as the union of f_m, $m \geq 0$, where each f_m has the following properties:

(a) for every $w \in W_m$, if $w \in B^{\mathcal{I}_0}$ then $f_m(w) \in B^{\mathcal{J}}$, for each $B \in \beta(\mathcal{T})$,

(b) for all $w, u \in W_m$, if $(w, u) \in R^{\mathcal{I}_0}$ then $(f_m(w), f_m(u)) \in R^{\mathcal{J}}$, for each $R \in \mathit{role}^{\pm}(\mathcal{T})$.

For the basis of induction, recall that $W_0 = \Lambda_0$ consists of the interpretations of all constants in $\mathcal{K}^{\dagger}$. So we have two cases:

- If $w = a^{\mathcal{I}_0}$, for $a \in \mathit{ob}(\mathcal{A})$, then set $f_0(w) = a^{\mathcal{J}}$.
- Otherwise, $w \in W_0 \setminus \{a^{\mathcal{I}_0} \mid a \in \mathit{ob}(\mathcal{A})\}$ and
  - if $w \in (\geq 1\, R)^{\mathcal{I}_0}$, for some $R \in \mathit{role}^{\pm}(\mathcal{T})$, then we can select, by **(ext-e$_{dr}$)**, some $w' \in (\geq 1\, R)^{\mathcal{J}}$ and set $f_0(w) = w'$;
  - otherwise, we take an arbitrary $w' \in \Gamma$ and set $f_0(w) = w'$.

Then (a) follows immediately from **(ext$_a$)** and **(ext$_{dr}$)**. In order to show (b) note that, for each role $R \in \mathit{role}^{\pm}(\mathcal{T})$ and all $w, u \in W_0$, if $(w, u) \in R^{\mathcal{I}_0}$ then $(a_i^{\mathcal{I}_0}, a_j^{\mathcal{I}_0}) \in R^{\mathcal{I}_0}$ for some $a_i, a_j \in \mathit{ob}(\mathcal{A})$ such that $w = a_i^{\mathcal{I}_0}$ and $u = a_j^{\mathcal{I}_0}$. Therefore, $R(a_i, a_j) \in \mathcal{A}$ or $\mathit{inv}(R)(a_j, a_i) \in \mathcal{A}$. In either case we have $(a_i^{\mathcal{J}}, a_j^{\mathcal{J}}) \in R^{\mathcal{J}}$, and so $(f_0(w), f_0(u)) \in R^{\mathcal{J}}$.

For the induction step, suppose that f_m has already been defined for W_m, $m \geq 0$. Set $f_{m+1}(w) = f_m(w)$ for all $w \in W_m$. Now consider an arbitrary $u \in W_{m+1} \setminus W_m$. According to the unravelling construction, there is some $w_0 \in W_m$ such that either $(w_0, u) \in P_k^{m+1}$ or $(u, w_0) \in P_k^{m+1}$ (a defect of w_0 is cured at step $m+1$). Let $R = P_k$ in the former case and $R = P_k^-$ in the latter. Then we have $w_0 \in (\geq 1\, R)^{\mathcal{I}_0}$, and so, by (a), $f_m(w_0) \in (\geq 1\, R)^{\mathcal{J}}$. Therefore, there exists a point $u' \in \Gamma$ such that $(f_m(w_0), u') \in R^{\mathcal{J}}$. Set $f_{m+1}(u) = u'$. By definition, we have (b). Recall also that $u \in (\geq 1\, \mathit{inv}(R))^{\mathcal{I}_0}$ and $u' \in (\geq 1\, \mathit{inv}(R))^{\mathcal{J}}$, and so we obtain (a) by **(ext$_{\mathit{inv}(dr)}$)**.

As f is a homomorphism, $\mathcal{I}_0 \models \psi$, and ψ is a positive existential sentence, we must have $\mathcal{J} \models \psi$ as well. ☐

Given a (possibly empty) set $D \subseteq \mathit{ob}(\mathcal{A})$ and some $k \geq 0$, we define the *k-neighbourhood* $\delta_k(D)$ of D as the minimal subset of Δ_0 satisfying the following conditions:

- $\{a^{\mathcal{I}_0} \mid a \in D\} \subseteq \delta_k(D)$;
- $W_0 \setminus \{a^{\mathcal{I}_0} \mid a \in \mathit{ob}(\mathcal{A})\} \subseteq \delta_k(D)$;
- if $u \in \delta_k(D)$, $w \in W_k$ and $(u, w) \in R^{\mathcal{I}_0}$ for some $R \in \mathit{role}^{\pm}(\mathcal{T})$, then $w \in \delta_k(D)$.

Note that $|\delta_k(D)| \leq |D| \cdot (|\mathit{role}^{\pm}(\mathcal{T})| \cdot q_{\mathcal{T}})^{k+1}$; more importantly, the size of $\delta_k(D)$ does not depend on the size of $\mathcal{A}$.

Lemma 12. *Let $\exists \vec{y}\, \psi(\vec{y})$ be a positive existential sentence in prenex form, $\vec{y} = y_1, \ldots, y_k$ and let D be the set of all constants occurring in ψ. Then $\mathcal{I}_0 \models \exists \vec{y}\, \psi(\vec{y})$ iff there is an assignment $\mathfrak{a}_0$ in Δ_0 such that $\mathcal{I}_0 \models^{\mathfrak{a}_0} \psi(\vec{y})$ and $\mathfrak{a}_0(y_i) \in \delta_k(D)$, $1 \leq i \leq k$.*

Proof. The implication ($\Leftarrow$) is trivial. To show ($\Rightarrow$) consider an assignment $\mathfrak{a}$ in Δ_0 such that $\mathcal{I}_0 \models^{\mathfrak{a}} \psi$. Suppose that there is i_0, $1 \leq i_0 \leq k$, with $\mathfrak{a}(y_{i_0}) \notin \delta_k(D)$. (If such an i_0 does not exist, we are done.) Set $Y^0 = \{y_{i_0}\}$. We iteratively extend this set to include all those variables that are (directly and indirectly) connected to y_{i_0}. More precisely, suppose that Y^n, $n \geq 0$, has been constructed. Let Y^{n+1} be the union of Y^n with the set of all those

y_i, $1 \le i \le k$, such that either $P(y_i, y_j)$ or $P(y_j, y_i)$ is a subformula of ψ, for some $y_j \in Y^n$. Denote the maximum of the Y^n by Y.

Let m the *minimum* number such that $\mathfrak{a}(y_i) \in W_m$, for $y_i \in Y$. Clearly, $m > 0$ as the cardinality of Y does not exceed k and the $P^{\mathcal{I}_0}$ connect only adjacent layers (i.e., V_n and V_{n+1}). For the same reason ψ has no subformulas of the form $P(a, y)$ or $P(y, a)$, for $a \in ob(\mathcal{A})$ and $y \in Y$. Now, for every $y_i \in Y$, we have $\mathfrak{a}(y_i) = w_i \in V_{m_i}$ for some $m \le m_i \le m_i + k$. It follows from the procedure of unravelling $\mathfrak{M}_0$ to $\mathcal{I}_0$ that, for each $y_i \in Y$, one can find $u_i \in V_{m_i - m}$ such that $cp(w_i) = cp(u_i)$; moreover, for every pair $y_i, y_j \in Y$, we have $(u_i, u_j) \in R^{\mathcal{I}_0}$ iff $(w_i, w_j) \in R^{\mathcal{I}_0}$, for each $R \in role^{\pm}(\mathcal{T})$. We define a new assignment $\mathfrak{a}_Y$ by taking

$$\mathfrak{a}_Y(y_i) \;=\; \begin{cases} \mathfrak{a}(y), & y \notin Y, \\ u_i, & y_i \in Y. \end{cases}$$

It follows that (i) $\mathcal{I}_0 \models^{\mathfrak{a}} \psi(\vec{y})$ iff $\mathcal{I}_0 \models^{\mathfrak{a}_Y} \psi(\vec{y})$ and (ii) $\mathfrak{a}_Y(y_i) \in \delta_k(D)$, for each $y_i \in Y$.

The above process can be now repeated to cure possible defects of $\mathfrak{a}_Y$. After sufficiently many repetitions we obtain an assignment $\mathfrak{a}_0$ as required by the lemma. $\square$

As an immediate consequence of the above two lemmas we obtain the following:

Corollary 13. *Let $q(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ be a positive existential query in prenex form with $\vec{y} = y_1, \dots, y_k$. Then, for every $\vec{a} = a_1, \dots, a_n$, we have $\mathcal{K} \models q(\vec{a})$ iff $\mathcal{I}_0 \models^{\mathfrak{a}} \varphi(\vec{a}, \vec{y})$, for some assignment $\mathfrak{a}$ in Δ_0 such that $\mathfrak{a}(y_i) \in \delta_k(D)$, for each $1 \le i \le k$, where*

$$D = \big\{a_1, \dots, a_n\big\} \;\cup\; \big\{a \in ob(\mathcal{A}) \mid \mathfrak{a}(y_i) = a^{\mathcal{I}_0},\ 1 \le i \le k\big\}.$$

We are now in a position to formulate our query answering algorithm for the given Horn knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and the positive existential query $q(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ in prenex form with $\vec{x} = x_1, \dots, x_n$ and $\vec{y} = y_1, \dots, y_k$.

The query answering algorithm:

1. First we check whether $\mathcal{K}$ is consistent. If it is inconsistent, then every $\vec{a} = a_1, \dots, a_n$ with $a_i \in ob(\mathcal{A})$ is an answer. Otherwise we go to step 2.

2. For every $(n + k)$-tuple $(a_1, \dots, a_n, c_1, \dots, c_k)$ with $a_i \in ob(\mathcal{A})$ and $c_i \in ob(\mathcal{A}) \cup \{\lambda\}$, where λ is a special marker meaning 'unnamed object' (i.e., some object that cannot be fixed at this stage since it may not belong to $ob(\mathcal{A})$; this partial assignment (i.e., λ stands for unassigned yet variable) will be extended to a full assignment at the very last step by enumerating all elements of $\delta_k(D)$), do:

   - set $D = \{a_i \mid 1 \le i \le n\} \;\cup\; \{c_i \mid c_i \ne \lambda, 1 \le i \le k\}$;
   - for each $R \in role^{\pm}(\mathcal{T})$, set $\boldsymbol{t}^{\bullet}(dr) = \emptyset$;
   - for each $a \in ob(\mathcal{A})$, compute its 'minimal type' as follows:
     - set $\boldsymbol{t}^0(a) = \big\{B \mid B^*(a) \in \Sigma_{\mathcal{K}}\big\}$
     - repeat
       - $\boldsymbol{t}^{n+1}(a) = \boldsymbol{t}^n(a) \cup \big\{B \mid \forall x\, (B_1^* \wedge \dots \wedge B_k^* \to B^*) \in \Sigma_{\mathcal{K}} \text{ and } B_1, \dots, B_k \in \boldsymbol{t}^n(a)\big\}$
     - until $\boldsymbol{t}^n(a) = \boldsymbol{t}^{n+1}(a)$;

  - for each $R \in \mathit{role}^{\pm}(\mathcal{T})$
    * if $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}^n(a)$ then set $\boldsymbol{t}^{\bullet}(dr) := \boldsymbol{t}^{\bullet}(dr) \cup \{\geq 1\, R\}$;
  - save $\boldsymbol{t}^n(a)$ as $\boldsymbol{t}(a)$ whenever $a \in D$;
- for all $R \in \mathit{role}^{\pm}(\mathcal{T})$, compute the minimal types of the dr (simultaneously):
  - for each $R \in \mathit{role}^{\pm}(\mathcal{T})$, set $\boldsymbol{t}^0(dr) = \boldsymbol{t}^{\bullet}(dr) \cup \big\{B \mid B^*(dr) \in \Sigma_{\mathcal{K}}\big\}$
  - repeat
    * for each $R \in \mathit{role}^{\pm}(\mathcal{T})$, set $\boldsymbol{t}^{n+1}(dr) = \boldsymbol{t}^n(dr)$
      $\cup \big\{B \mid \forall x\,(B_1^* \wedge \cdots \wedge B_k^* \rightarrow B^*) \in \Sigma_{\mathcal{K}} \text{ and } B_1, \ldots, B_k \in \boldsymbol{t}^n(dr)\big\}$;
    * for all $R, R' \in \mathit{role}^{\pm}(\mathcal{T})$
      · if $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}^n(dr')$ then set $\boldsymbol{t}^{n+1}(dr) := \boldsymbol{t}^{n+1}(dr) \cup \big\{\geq 1\, R\big\}$;
  - until $\boldsymbol{t}^n(dr) = \boldsymbol{t}^{n+1}(dr)$, for each $R \in \mathit{role}^{\pm}(\mathcal{T})$;
  - for each $R \in \mathit{role}^{\pm}(\mathcal{T})$, save $\boldsymbol{t}^n(dr)$ as $\boldsymbol{t}(dr)$;
- construct the part $\mathcal{I}_D$ of the model $\mathcal{I}_0$ that is based on $\delta_k(D)$ as its domain (note that all the required types are among the $\boldsymbol{t}(a)$, for $a \in D$, and the $\boldsymbol{t}(dr)$, for $R \in \mathit{role}^{\pm}(\mathcal{T})$);
- for each k-tuple $(c_1', \ldots, c_k')$ such that $c_i' \in \delta_k(D)$ if $c_i = \lambda$ and $c_i' = c_i$ otherwise, i.e., if $c_i \in D$, $1 \leq i \leq k$, do:
  - construct an assignment $\mathfrak{a}$ by taking $\mathfrak{a}(y_i) = c_i'$, $1 \leq i \leq k$;
  - evaluate $\mathcal{I}_D \models^{\mathfrak{a}} \varphi(\vec{a}, \vec{y})$, where $\vec{a} = a_1, \ldots, a_n$;
  - output $\vec{a}$ if the above relation holds.

This (deterministic) algorithm requires at most logarithmic space in the size of the ABox $\mathcal{A}$. Indeed, by Theorem 8, the consistency check at step 1 can be performed in logarithmic space in the size of $\mathcal{A}$. Then the space we need to enumerate the tuples $(a_1, \ldots, a_n, c_1, \ldots, c_k)$ and to store the current one is bounded by

$$(n + k) \cdot \big(\log |\mathcal{A}| + 1\big).$$

Next, we need extra space of size $2 \cdot |D| \cdot |\beta(\mathcal{T})|$ to compute and store the required types $\boldsymbol{t}^n(a)$ and $\boldsymbol{t}^{n+1}(a)$, for $a \in D$, plus $2 \cdot |\mathit{role}^{\pm}(\mathcal{T})| \cdot |\beta(\mathcal{T})|$ to compute and store the types $\boldsymbol{t}^n(dr)$ and $\boldsymbol{t}^{n+1}(dr)$, for $R \in \mathit{role}^{\pm}(\mathcal{T})$. At the next step the model $\mathcal{I}_D$ reuses those types and contains only $\delta_k(D)$ points; recall that

$$|\delta_k(D)| \;\leq\; |D| \cdot (|\mathit{role}^{\pm}(\mathcal{T})| \cdot q_{\mathcal{T}})^{k+1}$$

(and so the size of $\mathcal{I}_D$ does not depend on $|\mathcal{A}|$). Finally, the check at the last step involves the enumeration of all k-tuples of $\delta_k(D)$, which requires space of size $k \cdot \log |\delta_k(D)|$ (which again does not depend on the size of $\mathcal{A}$), and the actual evaluation of φ in $\mathcal{I}_D$ under the assignment $\mathfrak{a}$ does not depend on the size of $\mathcal{A}$ either. These calculations show that the overall space used by this deterministic algorithm is of the size logarithmic in the size of $\mathcal{A}$. □

It is not hard to see that the algorithm above belongs to the parallel complexity class AC_0.

If we deal with arbitrary, not necessarily Horn, knowledge bases $\mathcal{T}$, then Lemma 11 does not hold, and we have to consider basically all possible models for $\mathcal{T}$. It is not hard to prove, however, that if $\vec{a}$ is not an answer to the given positive existential query $q(\vec{x})$, then, similarly to Corollary 13, this fact can be established by means of some *finite* part of some (possibly infinite) model the size of which is linear in the size of the ABox. This observation provides us with a coNP query answering algorithm: to check that $\vec{a}$ is *not* an answer to $q(\vec{x})$ with respect to $\mathcal{T}$, we guess such a finite part and analyse all possible assignments to non-distinguished variables in it.

Theorem 14. *The data complexity of the positive existential query answering problem for DL-Lite$_{bool}$ knowledge bases bases is in* coNP.

Theorem 9 shows that this upper bound is optimal.

6 Conclusion

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. [8]. 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. Furthermore, as already done for *DL-Lite*, we are currently investigating the expressive power of the *DL-Lite$_{bool}$* family in the conceptual modeling (e.g., UML and EER) realm.

References

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

[2] K. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 493–574. 1990.

[3] F. Baader, C. Lutz, and B. Suntisrivaraporn. 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.

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

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

[6] E. Börger, E. Grädel, and Yu. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.

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

[8] 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.

[9] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tailoring OWL for data intensive ontologies. In *Proc. of the Workshop on OWL: Experiences and Directions (OWLED 2005)*, 2005.

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

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

[12] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[13] H. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York, 2nd edition, 2001.

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

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

[16] D. Kozen. *Theory of Computation*. Springer, 2006.

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

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

[19] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

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

[21] A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, 2:265–278, 1993.

[22] M. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC), 5–7 May 1982, San Francisco, California, USA*, pages 137–146. ACM, 1982.