Adding *DL-Lite* TBoxes to Proper Knowledge Bases

Giuseppe De Giacomo[1] and Hector Levesque[2]

[1] Sapienza Università di Roma
Rome, Italy
degiacomo@dis.uniroma1.it
[2] University of Toronto
Toronto, Canada
hector@cs.toronto.edu

DRAFT

Abstract. Levesque's proper knowledge bases (proper KBs) correspond to infinite sets of ground positive and negative facts, with the notable property that for FOL formulas in a certain normal form, which includes conjunctive queries and positive queries possibly extended with a controlled form of negation, entailment reduces to formula evaluation. However proper KBs represent extensional knowledge only. In description logic terms, they correspond to ABoxes. In this paper, we augment them with *DL-Lite* TBoxes, expressing intensional knowledge (i.e., the ontology of the domain). *DL-Lite* has the notable property that conjunctive query answering over TBoxes and standard description logic ABoxes is reducible to formula evaluation over the ABox only. Here, we investigate whether such a property extends to ABoxes consisting of proper KBs. Specifically, we consider two *DL-Lite* variants: *DL-Lite*$_{rdfs}$, roughly corresponding to RDFS, and *DL-Lite*$_{core}$, roughly corresponding to OWL 2 QL. We show that when a *DL-Lite*$_{rdfs}$ TBox is coupled with a proper KB, the TBox can be compiled away, reducing query answering to evaluation on the proper KB alone. But this reduction is no longer possible when we associate proper KBs with *DL-Lite*$_{core}$ TBoxes. Indeed, we show that in the latter case, query answering even for conjunctive queries becomes coNP-hard in data complexity.

1 Introduction

Many applications involving knowledge representation require an *open-world* setting, with *incomplete information* on their domain of interest [2, 7, 15, 16]. In such conditions, querying a knowledge base is typically based on *logical inference*, which is generally computationally infeasible. Indeed, the most successful applications of logics in Computer Science, namely relational databases [1] and model checking [5] assume complete information, and are based on the *evaluation* of logical formulas over a finite model. In particular, evaluating an FOL formula against a database requires only a simple recursive procedure and is indeed sub-polynomial (AC^0) in data complexity (i.e., in the computational complexity measured over the size of the database only). A natural question is whether there are interesting cases in which logical inference, required to deal with incomplete information, can be compiled into formula evaluation and hence retain the deductive efficiency of database retrieval without requiring complete knowledge, as with databases.

Based on this idea, Levesque [17] proposes the notion of a *proper knowledge base* (proper KB), where incomplete knowledge amounts to a possibly infinite set of positive or negative ground facts (without disjunctions or existentials). For this kind of KB he devises a reasoning procedure based on formula *evaluation* that essentially has the efficiency of first-order logic evaluation over a finite model (AC^0 in data complexity). He showed that this evaluation procedure is logically sound, and also complete when the formula is in a special normal form, called $\mathcal{NF}$. This class of formula notably includes conjunctive queries and positive queries possibly extended with a controlled form of negation, His ideas have been further developed in, e.g, [19, 18, 12].

Compiling logical inference into evaluation is also at the base of one of the most fruitful developments in description logics (DLs) [3] in the last decade, the introduction of so called ontology-based query answering systems and the *DL-Lite* family [9, 10]. These logics are designed for retaining the data complexity of FOL evaluation, while being able to capture most constructs used in UML Class Diagrams or Entity Relationship Diagrams [6]. They generalize W3C RDF Schema (RDFS) [8, 14], and are at the base of the OWL 2 QL profile of the W3C standard OWL 2 [20].

DLs consider knowledge divided into intensional knowledge and extensional knowledge. Intensional knowledge is expressed as a *TBox*, i.e., a finite set of universal logical assertions describing the domain of interest in terms of classes (called *concepts*), which are unary predicates, and relationships between classes (called *roles*), which are binary predicates. Extensional knowledge is expressed as an *ABox*, which consist of a finite set of positive facts involving concepts and roles of the TBox. (Open-world semantics is assumed.) Often the TBox is used to capture the ontology of the domain, while the ABox is used to capture contingent knowledge on individuals belonging to the domain. The main reasoning task of interest for the logics in the *DL-Lite* family is *query answering*, that is, computing substitutions for the open variables in the query for which the resulting formulas are logically entailed by the TBox and the ABox. The queries typically considered are conjunctive queries and the union of conjunctive queries. The first are FOL formulas where only conjunction and existential quantification is allowed, while the second include also disjunction (but, no forms of negation, nor universal quantification). The key feature of the DLs belonging to the *DL-Lite* family is the so-called *first-order* rewritability: query answering for a query Q can be performed in a sound and complete way by compiling away the TBox into a new FOL query $Q_{\mathcal{T}}$ that can be evaluated over the ABox, considered as a database. As the result, query answering in *DL-Lite* is AC^0 in data complexity like formula evaluation in a relational DB.

In this paper, we consider knowledge bases constituted by a TBox expressed in variants of *DL-Lite* and an ABox consisting of a Levesque's proper KB. In particular we consider two members of the *DL-Lite* family: *DL-Lite*$_{rdfs}$, which roughly correspond to RDFS [14], and *DL-Lite*$_{core}$, which roughly correspond to OWL 2 QL [9, 10]. The latter is actually the simplest *DL-Lite* that includes assertions of the form $A \sqsubseteq \exists R$.

We show that in the case of proper KBs extended with *DL-Lite*$_{rdfs}$ TBoxes, we can compile away the TBox retaining soundness and completeness of reasoning, so that when the resulting query is in $\mathcal{NF}$, the proper KB evaluation procedure will be both sound and complete. (In particular, for conjunctive queries and union of conjunctive queries this will always be the case.) This theoretical result has an immediate practical impact: it is possible to build efficient ontology-based query aswering system, where:

(i) RDFS is used to express the ontology of the domain (considering that *DL-Lite*$_{rdfs}$ captures the description logic fragment of RDFS, i.e., the fragment obtained by dropping RDFS meta-modeling features); *(ii)* proper KBs are used to express extensional knowledge in a very rich way, and *(iii)* SPARQL is used as a concrete query language for expressing ($\mathcal{NF}$) first-order queries [21].

Then we turn to *DL-Lite*$_{core}$ and show that, unfortunately, it is not possible to reduce query answering over TBoxes and ABoxes consisting of proper KBs to FOL query evaluation. We do so by proving that even for conjunctive queries, any sound and complete procedure must be *coNP*-hard, and hence, the proper KB evaluation procedure remains sound but must necessarily be incomplete. This has the practical impact of ruling out the possibility of building sound, complete and efficient ontology-based query answering systems that adopt OWL 2 QL as the ontology language.[3]

The rest of the paper is organized as follows. In Sections 2 and 3, we review proper KBs and *DL-Lite*. In Section 4, we show soundness and completeness results for TBoxes in *DL-Lite*$_{rdfs}$. In Section 5, we show that moving to TBoxes in *DL-Lite*$_{core}$, we lose the required efficiency. Finally in Section 6, we draw some conclusions and discuss future work. An appendix with the detailed proof of the result in Section 4 completes the paper.

2 Proper Knowledge Bases

Standard names. We use an ordinary first-order logical language $\mathcal{L}$ with an infinite supply of predicate symbols (including $=$), an infinite supply of constants, called *standard names* (which we write as $^{\#}1, ^{\#}2, ^{\#}3, \ldots$), and no other function or constant symbols. We denote the set of standard names by $\mathcal{N}$. We use the notation α^x_n to mean the result of replacing every free occurrence of variable x in formula α by standard name n. We adopt the usual Tarski semantics for $\mathcal{L}$, with $\models$ understood as normal logical entailment. However, we make the *unique name assumption* for the standard names. This means that we implicitly assume a theory of equality $\mathcal{E}$ formed by the usual axioms of equality (reflexitivity, symmetry, transitivity, and substitution of equals for equals) together with $\{\, n \neq n' \mid n$ and n' are distinct standard names $\}$. A *knowledge base* (KB) $\mathcal{K}$ consists of a finite set of sentences (closed formulas) belonging to $\mathcal{L}$. To $\mathcal{K}$ we will always implicitly add the equality theory $\mathcal{E}$. The (implicit) adoption of $\mathcal{E}$ implies that $\mathcal{K}$ has a model iff it has a *standard* model, that is, one where $=$ is interpreted as identity and the domain is isomorphic to the set of standard names Hence, w.l.o.g., we can make *domain closure assumption*: we can assume that the only objects in the domain of interpretation are the standard names. A key property of adopting standard names is the following.

Theorem 1. *[17] Suppose that $\mathcal{K}$ is a KB (including $\mathcal{E}$) and α a possibly open formula in $\mathcal{L}$. Let H be the set formed by all the (finitely many) standard names that appear in $\mathcal{K}$ or α and at least one other not occurring in $\mathcal{K}$ and α. Then*

$$\mathcal{K} \models \forall x.\alpha \;\; \textit{iff} \;\; \mathcal{K} \models \alpha^x_n \textit{ for every } n \in H.$$

This means that we can determine whether $\forall x.\alpha$ is entailed by checking whether a finite set of instances of α are entailed.

[3] In fact, our infeasibility result applies also to $\mathcal{EL}$, and hence rules out also OWL 2 EL [4, 20].

Levesque's proper KBs. Following [17], a *proper knowledge base* $\mathcal{A}$ (we denote proper KB by $\mathcal{A}$ to stress its role of ABox that they will play in this paper) is a finite collection of sentences of $\mathcal{L}$ of the form

$$\forall \boldsymbol{x}.(e \supset \varrho),$$

where

- e is an *equality formula*, i.e. a quantifier-free formula whose only predicate is equality, and free variables are among $\boldsymbol{x}$,
- ϱ is $P(\boldsymbol{x})$ or $\neg P(\boldsymbol{x})$, for some predicate P of arity $|\boldsymbol{x}|$ in $\mathcal{L}$,
- "$\supset$" is the usual material implication connective.

Proper KBs $\mathcal{A}$ are required to be *consistent* (under the implicit equality theory for standard names $\mathcal{E}$).

Every proper KB is a finite representation for a possibly infinite consistent set of ground literals:

$$\{\, \varrho\theta \mid \mathcal{A} \models \varrho\theta \,\}.$$

where, θ is a substitution of free variables by standard names, and $\varrho\theta$ denotes ϱ after the substitution.

Proper KBs have many interesting properties. For instance, any *finite* set of ground literals can be reformulated as a proper KB, by simply rewriting any ground literal $\varrho\theta$ in the set as $\forall \boldsymbol{x}.\, (\boldsymbol{x} = \boldsymbol{x}\theta \supset \varrho)$. Also (some) *infinite* sets of literals can be represented, *e.g.*, $\forall x, y, z.(x \neq z \wedge y = {}^{\#}3 \supset R(x, y, z))$. Also, similarly to databases, we can make the closed-world assumption for *some* of the predicates. For example, the following proper KB

$$\begin{aligned}\mathcal{A} = \{&\forall x.\, (x = {}^{\#}2 \vee x = {}^{\#}3 \vee x = {}^{\#}5 \vee x = {}^{\#}9 \supset P(x)),\\ &\forall x.\, (x \neq {}^{\#}2 \wedge x \neq {}^{\#}3 \wedge x \neq {}^{\#}5 \wedge x \neq {}^{\#}9 \supset \neg P(x))\}\end{aligned}$$

makes the closed-world assumption on P. But, unlike databases, we can leave the interpretation of a predicate *open* on a finite number of objects: consider the proper KB

$$\mathcal{A} = \{\, \forall x.(x \neq {}^{\#}1 \supset \neg P(x)) \,\}$$

then $\mathcal{A} \models \neg P({}^{\#}2)$, $\mathcal{A} \models \neg P({}^{\#}3)$, $A \models \neg P({}^{\#}4) \ldots$, but $\mathcal{A} \not\models P({}^{\#}1)$ and $\mathcal{A} \not\models \neg P({}^{\#}1)$.

Reasoning with proper KBs. The reasoning task of interest for proper KBs is query answering. In particular, as in [17], here we focus implicitly on boolean queries only. A (boolean) query Q is a sentence, i.e., a closed formula, in $\mathcal{L}$. Answering Q over a proper KB $\mathcal{A}$ consists in checking the entailment

$$\mathcal{A} \models Q.$$

It is in general *undecidable* to determine whether or not $\mathcal{A} \models Q$. (Consider the case $\mathcal{A} = \emptyset$, where we still need to determine whether an arbitrary first order formula is valid.) As an alternative, Levesque [17] proposes a *limited reasoning procedure* V analogous to the *evaluation* function used for databases under the closed-world assumption, which, however, may return 1 (known to be true), 0 (known to be false), or $\frac{1}{2}$ (unknown). Given a proper KB $\mathcal{A}$ and a query Q, the evaluation procedure $V[\mathcal{A}, Q]$ is defined as follows:

1. (Ground atomic fact) if $Q = \varrho\theta$ then

$$V[\mathcal{A}, \varrho\theta] = \begin{cases} 1 \text{ if there is a } \forall \boldsymbol{x}.(e \supset \varrho) \in \mathcal{A} \text{ s.t. } \mathcal{E} \models e\theta \\ 0 \text{ if there is a } \forall \boldsymbol{x}.(e \supset \overline{\varrho}) \in \mathcal{A} \text{ s.t. } \mathcal{E} \models e\theta \\ \frac{1}{2} \text{ otherwise} \end{cases}$$

   where $\overline{\varrho}$ denotes the result of adding or removing negation from ϱ.
2. (Ground equality atom) if $Q = (n = n')$ then

$$V[\mathcal{A}, (n = n')] = \begin{cases} 1 \text{ if } n \text{ and } n' \text{ are the same standard name} \\ 0 \text{ otherwise} \end{cases}$$

3. (Negation) if $Q = \neg\alpha$ then

$$V[\mathcal{A}, \neg\alpha] \;=\; 1 - V[\mathcal{A}, \alpha]$$

4. (Disjunction) if $Q = \alpha \vee \beta$ then

$$V[\mathcal{A}, (\alpha \vee \beta)] \;=\; \max\{V[\mathcal{A}, \alpha],\, V[\mathcal{A}, \beta]\}$$

5. (Conjunction) if $Q = \alpha \wedge \beta$ then

$$V[\mathcal{A}, (\alpha \wedge \beta)] \;=\; \min\{V[\mathcal{A}, \alpha],\, V[\mathcal{A}, \beta]\}$$

6. (Existential quantification) if $Q = \exists x.\alpha$ then

$$V[\mathcal{A}, \exists x.\alpha] \;=\; \max_{n \in H} V[\mathcal{A}, \alpha^x_n]$$

   where H is the set of standard names appearing in $\mathcal{A}$ or α plus a new one.
7. (Universal quantification) if $Q = \forall x.\alpha$ then

$$V[\mathcal{A}, \forall x.\alpha] \;=\; \min_{n \in H} V[\mathcal{A}, \alpha^x_n]$$

   where again H is the set of standard names appearing in $\mathcal{A}$ or α plus a new one.

Notice that as expected we have that

$$V[\mathcal{A}, \alpha \wedge \beta] = V[A, \neg(\neg\alpha \vee \neg\beta)], \qquad V[\mathcal{A}, \forall x.\alpha] = V[A, \neg\exists x.\neg\alpha].$$

The evaluation procedure V *is tractable* in a very strong sense. Analogously to database evaluation, is is easy to see that $V[\mathcal{A}, Q]$ can be computed in AC^0 in data complexity, i.e., in the number of standard names mentioned in $\mathcal{A}$ and Q. From a more practical point of view we have:

1. If e is a ground equality formula, then $\mathcal{E} \models e$ iff $V[\emptyset, e] = 1$ and can be determined in time linear in $|e|$.
2. $V[\mathcal{A}, \varrho\theta]$ can be determined in time linear in $|\mathcal{A}|$: scan $\mathcal{A}$ for $\forall \boldsymbol{x}.(e \supset \varrho)$ or $\forall \boldsymbol{x}.(e \supset \overline{\varrho})$ and check if $\mathcal{E} \models e\theta$.
3. Overall, computing $V[\mathcal{A}, Q]$ can be made as efficient as database retrieval [19].

The procedure V is always *logically sound*:

Theorem 2. *[17] For any proper KB $\mathcal{A}$ and any query Q in $\mathcal{L}$, we have:*

- *if* $V[\mathcal{A}, Q] = 1$ *then* $\mathcal{A} \models Q$*;*
- *if* $V[\mathcal{A}, Q] = 0$ *then* $\mathcal{A} \models \neg Q$*.*

However V *is not (and cannot be) logically complete* in general. For example: $\mathcal{E} \models (p \vee \neg p)$ but $V[\emptyset, (p \vee \neg p)] = \frac{1}{2}$.

In [17] completeness is shown for a semantically defined sublanguage of $\mathcal{L}$, called $\mathcal{NF}$. We say that a set of sentences S is *logically separable* iff for every consistent set of ground literals L, if $L \cup S$ has no standard model, then for some $\alpha \in S$, $L \cup \{ \alpha \}$ has no standard model. Then the normal form $\mathcal{NF} \subseteq \mathcal{L}$ is defined as the least set such that: *(i)* if α is a ground atom or equality formula, then $\alpha \in \mathcal{NF}$; *(ii)* if $\alpha \in \mathcal{NF}$, then $\neg\alpha \in \mathcal{NF}$; *(iii)* if $S \subseteq \mathcal{NF}$, S is logically separable, and S is finite, then $\bigwedge S \in \mathcal{NF}$; *(iv)* if $S \subseteq \mathcal{NF}$, S is logically separable, and $S = \{ \alpha_n^x \mid n \text{ is a standard name} \}$, then $\forall x.\alpha \in \mathcal{NF}$.

Theorem 3. *[17] For any proper KB $\mathcal{A}$ and any Q in $\mathcal{NF}$, we have:*

- *if* $\mathcal{A} \models Q$ *then* $V[\mathcal{A}, Q] = 1$*;*
- *if* $\mathcal{A} \models \neg Q$ *then* $V[\mathcal{A}, Q] = 0$*.*

Unfortunately $\mathcal{NF}$ is a semantical condition and checking if a formula is in $\mathcal{NF}$ is itself undecidable. However an interesting sufficient syntactic condition for belonging to $\mathcal{NF}$ is the following: we say two literals are *conflict-free* iff either they have the same polarity, or they use different predicates, or they use different standard names at some argument position.

Theorem 4. *[17] Let Q be a query in $\mathcal{L}$, if all pairs of literals in Q are conflict-free, then Q in $\mathcal{NF}$.*

Notably all *positive queries* (i.e., without $\neg$ and $\forall$), hence including *conjunctive queries* (i.e., using only $\wedge$ and $\exists$) and *union of conjunctive queries* (i.e., disjunctions of conjunctive queries), are conflict-free.

3 *DL-Lite*$_{rdfs}$ and *DL-Lite*$_{core}$

Description logics (DLs) [3] describe the domain of interest in terms of individuals denoting objects, concepts, denoting sets of objects, and roles, denoting binary relations between objects. In DLs, starting from *concepts names* (denoted by A) and *roles names* (denoted by R), we can construct complex *concepts* C, D and *roles* ρ, τ by inductively applying suitable constructors that depend on the DL in question.

A DL knowledge base $\mathcal{K}$ consists of a TBox $\mathcal{T}$, expressing intensional knowledge and an ABox $\mathcal{A}$, expressing extensional knowledge. *TBox* $\mathcal{T}$ is constituted by a finite set of *concept and role inclusions* of the form

$$C \sqsubseteq D, \qquad \rho \sqsubseteq \tau$$

where the form of concepts C, D and roles ρ, τ depend on the specific DL. We allow inclusions to be cyclic, which is required in virtually all ontology-based and conceptual modeling applications[4]. A standard DL *ABox* $\mathcal{A}$ consists of a finite set of positive ground literals involving concepts and roles of the TBox.

In this paper, we consider *DL-Lite*$_{core}$, the simplest language of the *DL-Lite* family [9, 10]. A *TBox* in *DL-Lite*$_{core}$ is a finite set of *inclusion assertions* of the form:

$$C \sqsubseteq D, \qquad C \sqsubseteq \neg D$$

where concepts C, D and roles ρ, τ are defined by the following syntax:

$$C, D ::= A \mid \exists\rho \qquad\qquad \rho ::= R \mid R^-$$

where $\exists\rho$ is the projection of binary role ρ on the first component and R^- is the *inverse* of role R. TBoxes expressed in *DL-Lite*$_{core}$ capture a core fragment of UML class diagrams: isa between classes ($A \sqsubseteq B$, A and B are concepts names), typing of roles ($\exists R \sqsubseteq A$, $\exists R^- \sqsubseteq B$), disjointness between classes ($A \sqsubseteq \neg B$), and mandatory participation of instances of a class to roles ($A \sqsubseteq \exists R$ or $B \sqsubseteq \exists R^-$), see [9]. *DL-Lite*$_{core}$ roughly corresponds to OWL 2 QL profile [20], where we disallow the use of inclusion assertion on roles $\rho \sqsubseteq \tau$.

Besides *DL-Lite*$_{core}$, we consider also *DL-Lite*$_{rdfs}$, which is obtained from *DL-Lite*$_{core}$ by dropping the possibility of using $\exists\rho$ on the right-hand side of inclusion assertions, but including *inclusion assertions on roles* of the form:

$$\rho \sqsubseteq \tau$$

with $\rho, \tau ::= R \mid R^-$. Hence, we loose the possibility of expressing mandatory participation, but we gain the possibility of expressing "subproperties" through isa's on roles, thus capturing RDFS [8] (without meta-level assertions), interpreted according to the *extensional semantics* [14].

We give the semantics of *DL-Lite*$_{core}$ and *DL-Lite*$_{rdfs}$ by exhibiting the FOL formula corresponding to each concept and role expression. In particular, if t and t' are terms, then $\rho[t, t']$ is the first-order formula defined by

$$\begin{aligned} R[t, t'] &= R(t, t') \\ R^-[t, t'] &= R(t', t). \end{aligned}$$

Similarly, $C[t]$ is the first-order formula defined by

$$\begin{aligned} A[t] &= A(t) \\ \exists\rho[t] &= \exists x.\rho[t, x] \\ \neg A[t] &= \neg A(t) \\ \neg\exists\rho[t] &= \forall x.\neg\rho[t, x]. \end{aligned}$$

Assertions of the form $C \sqsubseteq D$ and of the form $\rho \sqsubseteq \tau$ correspond, respectively, to

$$\forall x.(C[x] \supset D[x]) \qquad \forall x, y.(\rho[x, y] \supset \tau[x, y])$$

[4] When a TBox is acyclic, it can be treated as a set of abbreviations and eliminated w.l.o.g.

Typically in *DL-Lite*, we are interested in query answering, where queries are conjunctive queries or union of conjunctive queries. These are possibly open formulas expressed in terms of the concepts (unary predicates) and roles (binary predicate) of $\mathcal{T}$ and $\mathcal{A}$. When such formulas are closed we call such queries boolean. In particular, in this paper, we focus on *boolean queries* only. Given a TBox $\mathcal{T}$ and ABox $\mathcal{A}$, and a (boolean) query Q we are interested in checking whether

$$\mathcal{T} \cup \mathcal{A} \models Q$$

Notably the *DL-Lite* variants enjoy the *first-order rewritability* property, which in our setting says that for every conjunctive query or union of conjunctive queries Q:

$$\mathcal{T} \cup \mathcal{A} \models Q \text{ iff } \mathcal{A} \models Q_{\mathcal{T}}$$

where $Q_{\mathcal{T}}$ is a union of conjunctive query obtained by *rewriting* Q *using* $\mathcal{T}$, e.g., by the reformulation algorithm in [9], so as to "compile away" the TBox $\mathcal{T}$, and evaluate $Q_{\mathcal{T}}$ over the ABox $\mathcal{A}$ only, considered as a database (with complete information, i.e., closed world assumption). As a result, query evaluation in *DL-Lite* is AC^0 in data complexity, i.e., in the size of the ABox.

4 Proper KBs with *DL-Lite*$_{rdfs}$ TBoxes

The question that this paper addresses is whether something analogous to *DL-Lite* first-order rewritability holds also in the case of ABoxes consisting of proper KBs.

More precisely let's consider KBs formed by a *DL-Lite*$_{rdfs}$ TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ constituted by an proper KB over the unary and binary predicates forming the alphabet of the TBox. We restrict our attentions to proper KBs $\mathcal{A}$ that are *consistent* with the TBox $\mathcal{T}$, i.e., that $\mathcal{A} \models \neg\mathcal{T}$ where $\neg\mathcal{T}$ denotes the negation of the conjunction of all assertions in the TBox $\mathcal{T}$. On such KBs, we consider boolean queries Q which are any first-order sentence and we are interested in query answering i.e., in checking, whether:

$$\mathcal{T} \cup \mathcal{A} \models Q$$

In particular, we want to study whether there exists another FOL query $Q_{\mathcal{T}}$ such that

$$\mathcal{T} \cup \mathcal{A} \models Q \text{ iff } \mathcal{A} \models Q_{\mathcal{T}}$$

That is, we want to compile away the TBox and obtain another query $Q_{\mathcal{T}}$ to ask over the proper KB $\mathcal{A}$ alone. Notice that, differently from the case of standard *DL-Lite*, $\mathcal{A}$ cannot be seen as a database, since it still includes incomplete information and the close world assumption cannot be made. However if we can reduce $\mathcal{T} \cup \mathcal{A} \models Q$ to $\mathcal{A} \models Q_{\mathcal{T}}$, we can then use the evaluation procedure V to compute $V[\mathcal{A}, Q_{\mathcal{T}}]$, which is always sound and complete for queries in $\mathcal{NF}$. Indeed we are also interested in sufficient conditions for completeness. We show below that when $\mathcal{T}$ is a *DL-Lite*$_{rdfs}$ TBox, $Q_{\mathcal{T}}$ can always be obtained. Moreover that there are interesting class of queries Q (including conjunctive queries, and union of conjunctive queries) for which the evaluation procedure V applied to $Q_{\mathcal{T}}$ is indeed complete.

Without loss of generality we assume the query Q to be in negation normal form (NNF), i.e., with negation appearing only in literals. We use the following notation. The notation ρ^- means the role that results from adding or removing a superscript minus from ρ. The notation $\overline{C}$ means the concept that results from adding or removing a negation from C.

Next we define two crucial relations $\sqsubseteq_{\mathbf{R}}$ and $\sqsubseteq_{\mathbf{C}}$ denoting the chain of inclusions among concepts and roles respectively.

- The $\sqsubseteq_{\mathbf{R}}$ relation holding between pairs of roles is the reflexive transitive closure of the relation
$$\{\ (\rho, \tau) \mid\ \rho \sqsubseteq \tau \in \mathcal{T} \quad \text{or} \quad \rho^- \sqsubseteq \tau^- \in \mathcal{T}\ \}.$$
- The $\sqsubseteq_{\mathbf{C}}$ relation holding between pairs of concept is the reflexive transitive closure of the relation
$$\{\ (C, D) \mid\ C \sqsubseteq D \in \mathcal{T} \quad \text{or} \quad \overline{D} \sqsubseteq \overline{C} \in \mathcal{T} \quad \text{or} \quad C = \exists\rho,\ D = \exists\tau,\ \rho \sqsubseteq_{\mathbf{R}} \tau\ \}.$$

Note that, if $C \sqsubseteq_{\mathbf{C}} D$ then $\mathcal{T} \models \forall x.(C[x] \supset D[x])$ and similarly, if $\rho \sqsubseteq_{\mathbf{R}} \tau$ then $\mathcal{T} \models \forall x, y.(\rho[x, y] \supset \tau[x, y])$.

With these two relations at hand, we can define the rewriting $Q_{\mathcal{T}}$ of a query Q wrt a *DL-Lite*$_{rdfs}$ TBox $\mathcal{T}$.

Definition 1 (Rewriting). *Let $\mathcal{T}$ be a DL-Lite$_{rdfs}$ TBox and query Q in NNF, we define the* rewriting $Q_{\mathcal{T}}$ *of Q wrt $\mathcal{T}$ to be Q with every positive $A(t)$ replaced by*

$$\bigvee_{C \sqsubseteq_{\mathbf{C}} A} C[t],$$

every $\neg A(t)$ replaced by

$$\bigvee_{A \sqsubseteq_{\mathbf{C}} D} \overline{D}[t],$$

every positive $R(t, t')$ replaced by

$$\bigvee_{\rho \sqsubseteq_{\mathbf{R}} R} \rho[t, t'],$$

and every $\neg R(t, t')$ replaced by

$$\bigvee_{R \sqsubseteq_{\mathbf{R}} \tau} \neg\tau[t, t'] \ \vee \bigvee_{\exists R \sqsubseteq_{\mathbf{C}} D} \overline{D}[t] \ \vee \bigvee_{\exists R^- \sqsubseteq_{\mathbf{C}} D} \overline{D}[t'].$$

As we show below, the resulting formula enjoys the desired property: it is the result of compiling away the TBox from Q.

Theorem 5. *Let $\mathcal{T}$ be a DL-Lite$_{rdfs}$ TBox, $\mathcal{A}$ be a proper KB consistent with $\mathcal{T}$, Q a boolean query in NNF, and $Q_{\mathcal{T}}$ its rewriting defined as above. Then*

$$\mathcal{T} \cup \mathcal{A} \models Q \text{ iff } \mathcal{A} \models Q_{\mathcal{T}}$$

Proof. The proof requires extra machinery and has been moved to the appendix. □

In general, Theorem 5 does not induce an analogue of "first-order rewritability", in the sense that $Q_{\mathcal{T}}$ cannot be "evaluated" over the ABox $\mathcal{A}$. However, if $Q_{\mathcal{T}}$ is $\mathcal{NF}$, then it does, since the evaluation procedure V becomes sound and complete and hence it becomes sufficient to check whether $V[\mathcal{A}, Q_{\mathcal{T}}]$ to know whether $\mathcal{A} \models Q_{\mathcal{T}}$. Unfortunately checking whether $Q_{\mathcal{T}}$ is in $\mathcal{NF}$ is in general undecidable. However we can polynomially check $Q_{\mathcal{T}}$ for conflict-freeness. We can exploit this for giving a nice sufficient condition for the completeness of V.

Definition 2 (conflict-free for a TBox). *Let $\mathcal{T}$ be a DL-Lite$_{rdfs}$ TBox, Q a boolean query in NNF and $Q_{\mathcal{T}}$ its rewriting defined as above. Q is* conflict-free for a TBox $\mathcal{T}$ *iff $Q_{\mathcal{T}}$ is conflict-free.*

Note that *positive queries* are always conflict free for *DL-Lite$_{rdfs}$* TBoxes, including *conjunctive queries* and *union of conjunctive queries*. For example, if Q is a conjunctive query then $Q_{\mathcal{T}}$ is equivalent to a union of conjunctive queries, and hence is conflict-free.

For conflict free queries, we can exploit Theorem 5 and the soundness and completenes results for V to get:

Theorem 6. *Let $\mathcal{T}$ be a DL-Lite$_{rdfs}$ TBox, $\mathcal{A}$ be a proper KB consistent with $\mathcal{T}$, Q a boolean query in NNF , and $Q_{\mathcal{T}}$ its rewriting defined as above. If Q is conflict-free for $\mathcal{T}$, we have:*

- $\mathcal{T} \cup \mathcal{A} \models Q$ *iff* $V[\mathcal{A}, Q_{\mathcal{T}}] = 1$;
- $\mathcal{T} \cup \mathcal{A} \models \neg Q$ *iff* $V[\mathcal{A}, Q_{\mathcal{T}}] = 0$.

Hence, for queries that are conflict-free for the TBox, query answering reduces to evaluation and is indeed AC^0 in data complexity (i.e., in the number of standard names occurring in the ABox and in the query).

5 Proper KBs with *DL-Lite$_{core}$* TBoxes

Next we investigate KBs formed by a *DL-Lite$_{core}$* TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ formed as a proper KB. Unfortunately in this case we have a negative result: query answering by evaluation is in general unachievable even for queries consisting of boolean conjunctive queries. Indeed, if query aswering by evaluation were possible the data complexity of query answering would be AC^0. However we show that, even with a TBox consisting of a single assertion of the form $A \sqsubseteq \exists R$, conjunctive query answering in proper KBs requires reasoning by cases on the data, and is indeed *coNP*-hard in data complexity.

Theorem 7. *Conjunctive query answering in proper KBs with TBoxes including assertions of the form*

$$A \sqsubseteq \exists R$$

is coNP-hard with respect to data complexity.

Proof. The proof is based on a reduction from $2+2$-CNF unsatisfiability, which is shown to be *coNP*-complete in [13]. A $2+2$-CNF formula is a CNF formula in which each clause has exactly four literals: two positive ones and two negative ones.

Given a $2+2$-CNF formula $F = c_1 \wedge \cdots \wedge c_n$, where $c_i = \ell^i_{1+} \vee \ell^i_{2+} \vee \neg\ell^i_{1-} \vee \neg\ell^i_{2-}$, we associate with it the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. The alphabet of $\mathcal{K}$ includes one concept A and five roles P_1, P_2, N_1, N_2 and R with the following intuitive meaning:

- concept $A(x)$ denotes that x is an atomic proposition;
- role $P_1(x, y)$ (resp. $P_2(x, y)$) denotes that the atomic proposition y is in first (resp. second) positive position of the clause x;
- role $N_1(x, y)$ (resp. $N_2(x, y)$) denotes that the atomic proposition y is in first (resp. second) negative position of the clause x;
- role $R(x, y)$ denotes that the truth value y is assigned to the atomic proposition x.

The TBox $\mathcal{T}$ is simply:

$$A \sqsubseteq \exists R$$

The ABox $\mathcal{A}$ is formed by the proper KR equivalent to the following atomic assertions (see examples in Section 2 for hints on how to represent these finitely using equality):

$$
\begin{array}{l}
A(\ell^1_{1+}), A(\ell^1_{2+}), A(\ell^1_{1-}), A(\ell^1_{2-}), \\
\cdots \\
A(\ell^n_{1+}), A(\ell^n_{2+}), A(\ell^n_{1-}), A(\ell^n_{2-}), \\
\\
P_1(c_1, \ell^1_{1+}), P_2(c_1, \ell^1_{2+}), N_1(c_1, \ell^1_{1-}), N_2(c_1, \ell^1_{2-}), \\
\cdots \\
P_1(c_n, \ell^n_{1+}), P_2(c_n, \ell^n_{2+}), N_1(c_n, \ell^n_{1-}), N_2(c_n, \ell^n_{2-}) \\
\\
\neg R(\ell^1_{1+}, {}^\#2), \neg R(\ell^1_{1+}, {}^\#3), \neg R(\ell^1_{1+}, {}^\#4), \cdots \\
\neg R(\ell^1_{2+}, {}^\#2), \neg R(\ell^1_{2+}, {}^\#3), \neg R(\ell^1_{2+}, {}^\#4), \cdots \\
\neg R(\ell^1_{1-}, {}^\#2), \neg R(\ell^1_{1-}, {}^\#3), \neg R(\ell^1_{1-}, {}^\#4), \cdots \\
\neg R(\ell^1_{2-}, {}^\#2), \neg R(\ell^1_{2-}, {}^\#3), \neg R(\ell^1_{2-}, {}^\#4), \cdots \\
\cdots \\
\neg R(\ell^n_{1+}, {}^\#2), \neg R(\ell^n_{1+}, {}^\#3), \neg R(\ell^n_{1+}, {}^\#4), \cdots \\
\neg R(\ell^n_{2+}, {}^\#2), \neg R(\ell^n_{2+}, {}^\#3), \neg R(\ell^n_{2+}, {}^\#4), \cdots \\
\neg R(\ell^n_{1-}, {}^\#2), \neg R(\ell^n_{1-}, {}^\#3), \neg R(\ell^n_{1-}, {}^\#4), \cdots \\
\neg R(\ell^n_{2-}, {}^\#2), \neg R(\ell^n_{2-}, {}^\#3), \neg R(\ell^n_{2-}, {}^\#4), \cdots
\end{array}
$$

where, $c_1, \ldots, c_n$ and $\ell^1_{1+}, \ell^1_{2+}, \ell^1_{1-}, \ell^1_{2-}, \ldots, \ell^n_{1+}, \ell^n_{2+}, \ell^n_{1-}, \ell^n_{2-}$ are standard names chosen to be different from each other. The standard names ${}^\#0$ and ${}^\#1$ are used to represent the truth values $true$ and $false$ respectively. Intuitively the binary predicates P_1 P_2, N_1, N_2 associate to clauses c_i their four atomic propositions $\ell^i_{1+}, \ell^i_{2+}, \ell^i_{1-}, \ell^i_{2-}$ in their respective first/second, positive/negative position. The binary predicate R associates truth values to atomic propositions, which given the infinite set of assertions of the from $\neg R(\ell, k)$ can only be either ${}^\#0$ or ${}^\#1$ for the atomic propositions mentioned in the clauses.

Finally, we consider the following boolean conjunctive query:

$$
\begin{array}{rl}
Q \;=\; & \exists x, y_{1+}, y_{2+}, y_{1-}, y_{2-}. \\
& (P_1(x, y_{1+}) \wedge R(y_{1+}, {}^\#0) \wedge P_2(x, y_{2+}) \wedge R(y_{2+}, {}^\#0) \wedge \\
& \; N_1(x, y_{1-}) \wedge R(y_{1-}, {}^\#1) \wedge N_2(x, y_{2-}) \wedge R(y_{2-}, {}^\#1))
\end{array}
$$

Intuitively query Q checks if it is possible to assign the "wrong" truth value to all proposition y_{1+}, y_{2+}, y_{1-}, y_{2-} of some clause x. More precisely, checking whether $\mathcal{T} \cup \mathcal{A} \models Q$ (i.e., whether the query is certainly true in $\mathcal{T} \cup \mathcal{A}$) corresponds to checking whether in every truth assignment for the formula F there exists a clause whose positive atomic propositions are interpreted as false and whose negative atomic propositions are interpreted as true, i.e., a clause that is not satisfied. Next we show that the formula F is unsatisfiable if and only if $\mathcal{T} \cup \mathcal{A} \models Q$.

"$\Rightarrow$" Towards contradiction, suppose that the formula F is unsatisfiable but $\mathcal{T} \cup \mathcal{A} \not\models Q$. Then there exists a model $\mathcal{M}$ such that $\mathcal{M} \models \mathcal{T} \cup \mathcal{A}$, but $\mathcal{M} \not\models Q$. Notice that the given the assertions in $\mathcal{A}$ the only way not to satisfy Q is that for each $i = 1, \ldots, n$, we have that either $\neg R(\ell^i_{1+}, {}^\#0)$ or $\neg R(\ell^i_{2+}, {}^\#0)$ or $\neg R(\ell^i_{1-}, {}^\#1)$ or $\neg R(\ell^i_{2-}, {}^\#1)$. On the other hand for each such ℓ^i, by the TBox assertion $A \sqsubseteq \exists R$, there must exists some v such that $R(\ell^i, v)$, and because of the infinite assertions on $\neg R(\ell^i, {}^\#2)$, $\neg R(\ell^i, {}^\#3)$, $\neg R(\ell^i, {}^\#4), \ldots$ it must be the case that $v = {}^\#0$ or $v = {}^\#1$. So we have that for each clause c_i we must have that $R(\ell^i_{1+}, {}^\#1)$ or $R(\ell^i_{2+}, {}^\#1)$ or $R(\ell^i_{1-}, {}^\#0)$ or $R(\ell^i_{2-}, {}^\#0)$. But this would imply that the set of clauses F is indeed satisfiable, contradicting the hypotesis.

"$\Leftarrow$" Towards contradiction, suppose that $\mathcal{T} \cup \mathcal{A} \models Q$ but the formula F is satisfied by some truth assignment ϱ to its atomic propositions. Then, let $\mathcal{M}_\varrho$ be the interpretation for $\mathcal{T} \cup \mathcal{A}$ defined as follows:

$$
\begin{aligned}
A^{\mathcal{M}_\varrho} &= \{\ell \mid \ell \text{ is an atomic proposition in } F\} \\
P_1^{\mathcal{M}_\varrho} &= \{(c_i, \ell^i_{1+}) \mid \text{in } F, \ell^i_{1+} \text{ is the first positive atomic proposition of } c_i\} \\
P_2^{\mathcal{M}_\varrho} &= \{(c_i, \ell^i_{2+}) \mid \text{in } F, \ell^i_{2+} \text{ is the second positive atomic proposition of } c_i\} \\
N_1^{\mathcal{M}_\varrho} &= \{(c_i, \ell^i_{1-}) \mid \text{in } F, \ell^i_{1-} \text{ is the first negative atomic proposition of } c_i\} \\
N_2^{\mathcal{M}_\varrho} &= \{(c_i, \ell^i_{2-}) \mid \text{in } F, \ell^i_{2-} \text{ is the second negative atomic proposition of } c_i\} \\
R^{\mathcal{M}_\varrho} &= \{(\ell, v) \mid \varrho(\ell) = v\}
\end{aligned}
$$

It is easy to see that $\mathcal{M}_\varrho$ is a model of $\mathcal{T} \cup \mathcal{A}$. On the other hand, since F is satisfiable, for every clause in F there exists a positive atomic proposition interpreted as true or a negative atomic proposition interpreted as false. It follows that for every (standard name corresponding to) a clause c_i, either P_1 or P_2 relates c_i to a atomic proposition ℓ such that $(\ell, {}^\#1) \in R$ and $(\ell, {}^\#0) \notin R$, or either N_1 or N_2 relates c_i to a to a atomic proposition ℓ such that $(\ell, {}^\#0) \in R$ and $(\ell, {}^\#1) \notin R$. Hence Q evaluates to false in $\mathcal{M}_\varrho$, and therefore $\mathcal{T} \cup \mathcal{A} \not\models Q$, contradicting the hypothesis. □

This theorem rules out *DL-Lite*$_{core}$ and virtually all variants of *DL-Lite*, which allow for expressing $A \sqsubseteq \exists R$, including the two most prominent ones: *DL-Lite*$_R$, directly corresponding to OWL 2 QL [20], and *DL-Lite*$_\mathcal{A}$, often used in ontology-based data access applications [11]. For the same reason, it also rules out the whole $\mathcal{EL}$ family [4].

6 Conclusion

In this paper we have shown that is it feasible to extend Levesque's proper KBs with TBoxes expressed in *DL-Lite*$_{rdfs}$ while retaining the ability to reason by evaluating formulas for first-order queries of certain forms and hence solve query answering in

AC^0 in data complexity as for standard database query evaluation. This result is of practical interest considering that *DL-Lite*$_{rdfs}$ captures the description logic fragment of RDFS (i.e., dropping meta-modeling features) and that SPARQL can be used as a concrete query language for expressing first-order queries over RDFS [21].

We also showed that this result cannot be generalized to TBoxes expressed in OWL 2 QL or any *DL-Lite* variant that allows for assertions of the form $A \sqsubseteq \exists R$ [9, 10], including *DL-Lite*$_{core}$, since when combined with the power of proper KBs, reasoning by cases become necessary (query answering becomes *coNP*-hard). In fact, the result applies also to the DL $\mathcal{EL}$ [4] and hence to OWL 2 EL [20] as well.

Our result on proper KBs with *DL-Lite*$_{rdfs}$ TBoxes could be slightly generalized. In particular, it would be interesting to extend the TBox language, e.g., to deal with assertions of the form $\rho \sqsubseteq \neg\tau$ to express disjoint extension of roles, and getting closer to OWL 2 QL [9, 20], or to by considering n-ary roles [10]. Also, the language of proper KBs themselves can be extended, e.g., to deal with unknown individuals, i.e., *nulls*, as in [12]. We leave these extensions for future studies.

A Appendix

In this appendix we prove Theorem 5. As ABox $\mathcal{A}$ we consider any set (possibly infinite) of assertions of the form $A(n)$, $R(n, m)$ and $\neg A(n)$, $\neg R(n, m)$, where n and m are standard names. Notice that these ABoxes are more general then proper KBs (which indeed correspond to *certain* ABoxes of this form). The TBox $\mathcal{T}$ is a standard *DL-Lite*$_{rdfs}$ TBox. We assume $\mathcal{A}$ to be consistent with $\mathcal{T}$ (i.e., $\mathcal{A} \not\models \neg\mathcal{T}$.) We use the following notation. If $\mathcal{M}$ is a logical interpretation, then the extension of a concept C and a role ρ are respectively:

$$C^{\mathcal{M}} = \{n \mid \mathcal{M} \models C[n]\}, \qquad \rho^{\mathcal{M}} = \{(n, n') \mid \mathcal{M} \models \rho[n, n']\}.$$

Note that for any C, $\overline{C}^{\mathcal{M}} = \overline{C^{\mathcal{M}}}$. For any concept C and role ρ, we define

$$\mathit{MIN}(C) = \{n \mid \mathcal{A} \models C[n]\}, \qquad \mathit{MIN}(\rho) = \{(n, m) \mid \mathcal{A} \models \rho[n, m]\}.$$

Note that for any $\mathcal{M}$ such that $\mathcal{M} \models \mathcal{A}$, $\mathit{MIN}(C) \subseteq C^{\mathcal{M}}$ and $\mathit{MIN}(\rho) \subseteq \rho^{\mathcal{M}}$. For any $\mathcal{M}$ and C, we define

$$\mathcal{F}(C) = \bigcup_{D \sqsubseteq_{\mathbf{c}} C} \mathcal{E}(D)$$

where

$$\mathcal{E}(C) = \mathit{MIN}(C) \cup \bigcap_{C \sqsubseteq_{\mathbf{c}} D} D^{\mathcal{M}}.$$

Note that for any C, $\mathit{MIN}(C) \subseteq \mathcal{E}(C) \subseteq \mathcal{F}(C)$.

Lemma 1. *If $\mathcal{M} \models \mathcal{A}$ then $\mathcal{F}(C) \cap \mathcal{F}(\overline{C}) = \emptyset$.*

Proof. Suppose not. Then there is $n \in \mathcal{F}(C)$ and $n \in \mathcal{F}(\overline{C})$. Then for some $D \sqsubseteq_{\mathbf{c}} C$, $n \in \mathcal{E}(D)$ and for some $D' \sqsubseteq_{\mathbf{c}} \overline{C}$, $n \in \mathcal{E}(D')$. Since we have $D \sqsubseteq_{\mathbf{c}} C$ and $C \sqsubseteq_{\mathbf{c}} \overline{D'}$ (which is contrapositive of $D' \sqsubseteq_{\mathbf{c}} \overline{C}$), we also have $D \sqsubseteq_{\mathbf{c}} \overline{D'}$.

Now consider the cases for $n \in \mathcal{E}(D)$. If $n \in \mathit{MIN}(D)$, then $\mathcal{A} \models D[n]$. Since $\mathcal{A} \not\models \neg\mathcal{T}$, $\mathcal{A} \not\models D'[n]$ and so $n \notin \mathit{MIN}(D')$. Moreover, since $\mathcal{M} \models \mathcal{A}$ it follows $n \in D^{\mathcal{M}}$, and hence $n \notin D'^{\mathcal{M}}$. Since by definition $D' \sqsubseteq_{\mathbf{c}} D'$, we get $n \notin \bigcap_{D' \sqsubseteq_{\mathbf{c}} E} E^{\mathcal{M}}$. This contradicts $n \in \mathcal{E}(D')$.

On the other hand, if $n \in \bigcap_{D \sqsubseteq_{\mathbf{c}} E} E^{\mathcal{M}}$, then $n \in \overline{D'}^{\mathcal{M}}$, so $n \notin D'^{\mathcal{M}}$. Since $\mathcal{M} \models \mathcal{A}$, $n \notin \mathit{MIN}(D')$. Moreover $n \notin \bigcap_{D' \sqsubseteq_{\mathbf{c}} E} E^{\mathcal{M}}$, since $D' \sqsubseteq_{\mathbf{c}} D'$. This again contradicts $n \in \mathcal{E}(D')$. □

For any $\mathcal{M}$ and ρ, we define

$$\mathcal{G}(\rho) = \bigcup_{\tau \sqsubseteq_{\mathbf{R}} \rho} \mathcal{H}(\tau)$$

where

$$\mathcal{H}(\rho) = \mathit{MIN}(\rho) \cup [(\mathcal{E}(\exists\rho) \times \mathcal{E}(\exists\rho^-)) \cap \bigcap_{\rho \sqsubseteq_{\mathbf{R}} \tau} \tau^{\mathcal{M}}].$$

Lemma 2. $\mathcal{G}(\rho) \subseteq \mathcal{F}(\exists\rho) \times \mathcal{F}(\exists\rho^-)$

Proof. We prove it for $\rho = R$. (The case of R^- is analogous.) If $(n, m) \in \mathcal{G}(R)$ then for some $\rho \sqsubseteq_{\mathbf{R}} R$, $(n, m) \in \mathcal{H}(\rho)$. There are two cases: if $(n, m) \in \mathit{MIN}(\rho)$, then $n \in \mathit{MIN}(\exists\rho)$ and then $m \in \mathit{MIN}(\exists\rho^-)$, in which case, $n \in \mathcal{E}(\exists\rho)$ and $m \in \mathcal{E}(\exists\rho^-)$; if $(n, m) \in (\mathcal{E}(\exists\rho \times \mathcal{E}(\exists\rho^-))$, then again $n \in \mathcal{E}(\exists\rho)$ and $m \in \mathcal{E}(\exists\rho^-)$. Since $\rho \sqsubseteq_{\mathbf{R}} R$, $\exists\rho \sqsubseteq_{\mathbf{c}} \exists R$ and $\exists\rho^- \sqsubseteq_{\mathbf{c}} \exists R^-$. It follows that $n \in \mathcal{F}(\exists R)$ and $m \in \mathcal{F}(\exists R^-)$. □

Given a logical interpretation $\mathcal{M}$, we define a related one $\mathcal{M}^*$ by $A^{\mathcal{M}^*} = \mathcal{F}(A)$ and $R^{\mathcal{M}^*} = \mathcal{G}(R)$.

Lemma 3. *If $\mathcal{M} \models \mathcal{A}$ then $\mathcal{M}^* \models \mathcal{T}$.*

Proof. First suppose $C \sqsubseteq D \in \mathcal{T}$. Note that if $C \sqsubseteq_{\mathbf{c}} D$, then $\{E \mid E \sqsubseteq_{\mathbf{c}} C\} \subseteq \{E \mid E \sqsubseteq_{\mathbf{c}} D\}$, and so $\mathcal{F}(C) \subseteq \mathcal{F}(D)$. Because of the restriction on the TBOX language, $C = A$ or $C = \exists\rho$. In the case of A, we have $A^{\mathcal{M}^*} = \mathcal{F}(A)$; in the case of $\exists\rho$, we have $\exists\rho^{\mathcal{M}^*} \subseteq \mathcal{F}(\exists\rho)$ by Lemma 2. In both cases, $C^{\mathcal{M}^*} \subseteq \mathcal{F}(C)$. Similarly, because of the language restriction, $D = A$ or $D = \neg A$, and so either way $\mathcal{F}(D) \subseteq D^{\mathcal{M}^*}$ (since $\mathcal{F}(\neg A) \subseteq \overline{\mathcal{F}(A)}$ by Lemma 1). It then follows that $C^{\mathcal{M}^*} \subseteq \mathcal{F}(C) \subseteq \mathcal{F}(D) \subseteq D^{\mathcal{M}^*}$.

Now suppose $\rho \sqsubseteq \tau \in \mathcal{T}$. As above, we have that if $\rho \sqsubseteq_{\mathbf{R}} \tau$, then $\{\rho' \mid \rho' \sqsubseteq_{\mathbf{R}} \rho\} \subseteq \{\tau' \mid \tau' \sqsubseteq_{\mathbf{R}} \tau\}$, and so $\mathcal{G}(\rho) \subseteq \mathcal{G}(\tau)$. It follows that $\rho^{\mathcal{M}^*} = \mathcal{G}(\rho) \subseteq \mathcal{G}(\tau) = \tau^{\mathcal{M}^*}$. □

Lemma 4. *If $\mathcal{M} \models \mathcal{A}$ then $\mathcal{M}^* \models \mathcal{A}$.*

Proof. We consider the four cases of assertions in $\mathcal{A}$.

Suppose $A(n) \in \mathcal{A}$. Then $n \in \mathit{MIN}(A)$, so $n \in \mathcal{E}(A) \subseteq \mathcal{F}(A)$. Therefore, $\mathcal{M}^* \models A(n)$.

Suppose $\neg A(n) \in \mathcal{A}$. Then $n \in \mathit{MIN}(\neg A)$ and $n \notin A^{\mathcal{M}}$. Now suppose that $D \sqsubseteq_{\mathbf{c}} A$ for some D. Then $n \notin \mathit{MIN}(D)$ since otherwise $\mathcal{A} \models \neg\mathcal{T}$. Since $n \notin A^{\mathcal{M}}$, $n \notin \mathcal{E}(D)$. Since this holds for any $D \sqsubseteq_{\mathbf{c}} A$, $n \notin \mathcal{F}(A)$ and hence $\mathcal{M}^* \models \neg A(n)$.

Suppose $R(n,m) \in \mathcal{A}$. Then $(n,m) \in \mathit{MIN}(R)$, so $(n,m) \in \mathcal{H}(R) \subseteq \mathcal{G}(R)$. Therefore, $\mathcal{M}^* \models R(n,m)$.

Finally, suppose $\neg R(n,m) \in \mathcal{A}$. Then $(n,m) \notin R^{\mathcal{M}}$. Now suppose that $\tau \sqsubseteq_{\mathbf{R}} R$ for some τ. Then $(n,m) \notin \mathit{MIN}(\tau)$ since otherwise $\mathcal{A} \models \neg\mathcal{T}$. Since $(n,m) \notin R^{\mathcal{M}}$, $(n,m) \notin \mathcal{H}(\tau)$. Since this holds for any $\tau \sqsubseteq_{\mathbf{R}} R$, $(n,m) \notin \mathcal{G}(R)$ and hence $\mathcal{M}^* \models \neg R(n,m)$. □

Lemma 5. $\mathcal{T} \models (Q_{\mathcal{T}} \supset Q)$.

Proof. Assume that $\mathcal{M} \models \mathcal{T}$ and prove by induction on $|Q|$ that if $\mathcal{M} \models Q_{\mathcal{T}}$ then $\mathcal{M} \models Q$. Here are the base cases only.

Suppose $Q = A(n)$ and $\mathcal{M} \models Q_{\mathcal{T}}$. So for some $C \sqsubseteq_{\mathbf{C}} A$, $\mathcal{M} \models C[n]$. Since $\mathcal{M} \models \mathcal{T}$, $\mathcal{M} \models A(n)$.

Suppose $Q = \neg A(n)$ and $\mathcal{M} \models Q_{\mathcal{T}}$. So for some $A \sqsubseteq_{\mathbf{C}} D$, $\mathcal{M} \models \neg D[n]$. Since $\mathcal{M} \models \mathcal{T}$, $\mathcal{M} \models \neg A(n)$.

Suppose $Q = R(n,m)$ and $\mathcal{M} \models Q_{\mathcal{T}}$. So for some $\rho \sqsubseteq_{\mathbf{R}} R$, $\mathcal{M} \models \rho[n,m]$. Since $\mathcal{M} \models \mathcal{T}$, $\mathcal{M} \models R(n,m)$.

Suppose $Q = \neg R(n,m)$ and $\mathcal{M} \models Q_{\mathcal{T}}$. So one of the following: for some $R \sqsubseteq_{\mathbf{R}} \tau$, $\mathcal{M} \models \neg\tau[n,m]$ or for some $\exists R \sqsubseteq_{\mathbf{C}} D$, $\mathcal{M} \models \neg D[n]$ or for some $\exists R^- \sqsubseteq_{\mathbf{C}} D$, $\mathcal{M} \models \neg D[m]$. In all cases, since $\mathcal{M} \models \mathcal{T}$, $\mathcal{M} \models \neg R(n,m)$. □

Lemma 6. *If $\mathcal{M} \models \mathcal{A}$ and $\mathcal{M}^* \models Q$, then $\mathcal{M} \models Q_{\mathcal{T}}$.*

Proof. The proof is by induction on $|Q|$. Here are the base cases only.

Suppose $\mathcal{M}^* \models A(n)$. So for some $C \sqsubseteq_{\mathbf{C}} A$, $n \in \mathcal{E}(C)$. There are two cases: $n \in \mathit{MIN}(C)$ or $n \in C^{\mathcal{M}}$. Either way, since $\mathcal{M} \models \mathcal{A}$, $\mathcal{M} \models C[n]$. So $\mathcal{M} \models Q_{\mathcal{T}}$.

Suppose $\mathcal{M}^* \models \neg A(n)$. So for every $C \sqsubseteq_{\mathbf{C}} A$, $n \notin \mathcal{E}(C)$ and so $n \notin \mathcal{E}(A)$. So for some $A \sqsubseteq_{\mathbf{C}} D$, $n \notin D^{\mathcal{M}}$, and thus $\mathcal{M} \models \overline{D}[n]$. Therefore $\mathcal{M} \models Q_{\mathcal{T}}$.

Suppose $\mathcal{M}^* \models R(n,m)$. So for some $\rho \sqsubseteq_{\mathbf{R}} R$, $(n,m) \in \mathcal{H}(\rho)$. There are two cases: $(n,m) \in \mathit{MIN}(\rho)$ or $(n,m) \in \rho^{\mathcal{M}}$. Either way, since $\mathcal{M} \models \mathcal{A}$, $\mathcal{M} \models \rho[n,m]$. So $\mathcal{M} \models Q_{\mathcal{T}}$.

Suppose $\mathcal{M}^* \models \neg R(n,m)$. So for every $\rho \sqsubseteq_{\mathbf{R}} R$, $(n,m) \notin \mathcal{H}(\rho)$ and so $(n,m) \notin \mathcal{H}(R)$. Then $(n,m) \notin R^{\mathcal{M}} \cap (\mathcal{E}(\exists R) \times \mathcal{E}(\exists R^-))$. There are three cases: $(n,m) \notin R^{\mathcal{M}}$, in which case for some $R \sqsubseteq_{\mathbf{R}} \tau$, $\mathcal{M} \models \neg\tau[n,m]$, namely $\tau = R$; or $n \notin \mathcal{E}(\exists R)$ in which case for some $\exists R \sqsubseteq_{\mathbf{C}} D$, $n \notin D^{\mathcal{M}}$, and so $\mathcal{M} \models \overline{D}[n]$; or $m \notin \mathcal{E}(\exists R^-)$ in which case for some $\exists R^- \sqsubseteq_{\mathbf{C}} D$, $m \notin D^{\mathcal{M}}$, and so $\mathcal{M} \models \overline{D}[m]$. In all cases, $\mathcal{M} \models Q_{\mathcal{T}}$. □

Finally we are ready to prove the main claim.

Main claim. *$\mathcal{T} \cup \mathcal{A} \models Q$ iff $\mathcal{A} \models Q_{\mathcal{T}}$.*

Proof. ($\Leftarrow$) Suppose $\mathcal{A} \models Q_{\mathcal{T}}$. Let $\mathcal{M}$ be any logical interpretation such that $\mathcal{M} \models \mathcal{T} \cup \mathcal{A}$. Since, $\mathcal{M} \models \mathcal{A}$ and $\mathcal{A} \models Q_{\mathcal{T}}$, $\mathcal{M} \models Q_{\mathcal{T}}$. Since $\mathcal{M} \models \mathcal{T}$, $\mathcal{M} \models Q$ by Lemma 5. Therefore, $\mathcal{T} \cup \mathcal{A} \models Q$.

($\Rightarrow$) Suppose $\mathcal{A} \not\models Q_{\mathcal{T}}$. Then, there is an $\mathcal{M}$ such that $\mathcal{M} \models \mathcal{A}$ and $\mathcal{M} \not\models Q_{\mathcal{T}}$. By Lemma 3, $\mathcal{M}^* \models \mathcal{T}$. By Lemma 4, $\mathcal{M}^* \models \mathcal{A}$. By Lemma 6, $\mathcal{M}^* \not\models Q$. Therefore, $\mathcal{T} \cup \mathcal{A} \not\models Q$. □

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison Wesley (1995)
2. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann (2008)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
4. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. pp. 364–369 (2005)
5. Baier, C., Katoen, J.P.: Principles of model checking. MIT Press (2008)
6. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. Artif. Intell. 168(1–2), 70–118 (2005)
7. Brachman, R., Levesque, H.: Knowledge Representation and Reasoning. Morgan Kaufmann (2004)
8. Brickley, D., Guha, R., McBride, B.: RDF Schema 1.1. W3C Recommendation, World Wide Web Consortium (Feb 2014), http://www.w3.org/TR/rdf-schema/
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reasoning 39(3), 385–429 (2007)
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. Artif. Intell. 195, 335–360 (2013)
11. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M.: Using OWL in data integration. In: Semantic Web Information Management - A Model-Based Perspective, pp. 397–424. Springer (2009)
12. De Giacomo, G., Lespérance, Y., Levesque, H.J.: Efficient reasoning in proper knowledge bases with unknown individuals. In: Proc. of IJCAI'11. pp. 827–832 (2011)
13. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. J. of Logic and Computation 4(4), 423–452 (1994)
14. Franconi, E., Gutierrez, C., Mosca, A., Pirrò, G., Rosati, R.: The logic of extensional RDFS. In: Proc. of ISWC'13. pp. 101–116 (2013)
15. Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA. pp. 233–246 (2002)
16. Lenzerini, M.: Ontology-based data management. In: Proceedings of the 6th Alberto Mendelzon International Workshop on Foundations of Data Management, Ouro Preto, Brazil, June 27-30, 2012. pp. 12–15 (2012)
17. Levesque, H.J.: A completeness result for reasoning with incomplete first-order knowledge bases. In: KR. pp. 14–23 (1998)
18. Liu, Y.: Tractable Reasonng in Incomplete First-Order Knowledge Bases. Ph.D. thesis, Department of Computer Science, University of Toronto (2006)
19. Liu, Y., Levesque, H.J.: A tractability result for reasoning with incomplete first-order knowledge bases. In: IJCAI. pp. 83–88 (2003)
20. W3C: OWL 2 web ontology language primer (second edition). Tech. rep., W3C Recommendation 11 December 2012 (2012)
21. W3C: SPARQL 1.1 overview. Tech. rep., W3C Recommendation 21 March 2013 (2013)