The Combined Approach to Query Answering in *DL-Lite*

R. Kontchakov,[1] C. Lutz,[2] D. Toman,[3] F. Wolter[4] and M. Zakharyaschev[1]

[1] Department of CS and Information Systems
Birkbeck College London, UK
{roman,michael}@dcs.bbk.ac.uk

[2] Fachbereich Mathematik und Informatik
Universität Bremen, Germany
clu@informatik.uni-bremen.de

[3] D.R. Cheriton School of CS
University of Waterloo, Canada
david@cs.uwaterloo.ca

[4] Department of Computer Science
University of Liverpool, UK
frank@csc.liv.ac.uk

Abstract

Databases and related information systems can benefit from the use of ontologies to enrich the data with general background knowledge. The *DL-Lite* family of ontology languages was specifically tailored towards such ontology-based data access, enabling an implementation in a relational database management system (RDBMS) based on a query rewriting approach. In this paper, we propose an alternative approach to implementing ontology-based data access in *DL-Lite*. The distinguishing feature of our approach is to allow rewriting of both the query and the data. We show that, in contrast to the existing approaches, no exponential blowup is produced by the rewritings. Based on experiments with a number of real-world ontologies, we demonstrate that query execution in the proposed approach is often more efficient than in existing approaches, especially for large ontologies. We also show how to seamlessly integrate the data rewriting step of our approach into an RDBMS using views (which solves the update problem) and make an interesting observation regarding the succinctness of queries in the original query rewriting approach.

Introduction

Description logics (DLs), as well as DL-based dialects of the Web ontology languages OWL and OWL 2, have been tailored as knowledge representation formalisms supporting the 'classical' reasoning tasks such as satisfiability and subsumption, which are used at the stage of ontology design. Modern DL reasoners are indeed able to classify large and complex real-world ontologies, the OWL version of the medical ontology Galen being the latest fallen stronghold (Kazakov 2009). Along with the growing popularity and availability of ontologies, novel ways of their use, which go far beyond classical reasoning, have started to emerge. In particular, it is generally believed in the KR community that ontology languages can play a key role in the next generation of information systems. The core idea is *ontology-based data access*, where ontologies enrich the data with additional background knowledge, thus facilitating the use and integration of incomplete and semistructured data from heterogeneous sources (Dolby et al. 2008; Heymans et al. 2008; Poggi et al. 2008). In this context, the main reasoning task is to answer queries posed to the data while taking account of the knowledge provided by the ontology. It has turned out, however, that this task does not scale well in traditional DLs and is dramatically less efficient than querying in standard relational database management systems (RDBMSs).

An investigation of DLs for which ontology-based data access can be reduced to query answering in RDBMSs—thus taking advantage of the decades of research invested to make RDBMSs scalable—was launched in the series of papers (Calvanese et al. 2005; 2006; 2008). The ultimate aim was to identify DLs for which every conjunctive query q over a data instance $\mathcal{D}$, given an ontology $\mathcal{T}$, can be rewritten—independently of $\mathcal{D}$—into a first-order query $q^{\mathcal{T}}$ over $\mathcal{D}$ alone and then executed by an RDBMS. This effort gave birth to a new family of DLs, called the *DL-Lite* family, and subsequently to the OWL 2 QL profile of OWL 2. The *rewriting approach* to ontology-based data access has been implemented in various systems such as QuOnto (Acciarri et al. 2005), Owlgres (Stocker and Smith 2008) and REQUIEM (Pérez-Urbina, Motik, and Horrocks 2008). Unfortunately, experiments have revealed that these systems do not provide sufficient scalability even for medium-size ontologies (with a few hundred axioms). In a nutshell, the reason is that the rewritten queries are of size $(|\mathcal{T}| \cdot |q|)^{|q|}$ in all known rewriting techniques, which can be prohibitive for efficient execution by an RDBMS when $|\mathcal{T}|$ is large (even if $|q|$ is relatively small).

A different *combined approach* to ontology-based data access using RDBMSs, with the main goal of overcoming the inherent limitation of the rewriting approach being applicable only to DLs where conjunctive query answering is in AC^0 for data complexity, has been proposed in (Lutz, Toman, and Wolter 2009). This combined approach separates query answering into two steps: first, the data $\mathcal{D}$ is extended—independently of possible queries—by taking account of the ontology $\mathcal{T}$, and then any given query over $\mathcal{T}$ and $\mathcal{D}$ is rewritten—independently of $\mathcal{D}$—to a relational query over the extended data. The new technique was applied to (extensions of) the DL $\mathcal{EL}$ (underlying the OWL 2 EL profile), which is PTIME-complete for data complexity.

In this paper, we investigate the application of the combined approach to conjunctive query answering for *DL-Lite* ontologies. In particular, we present *polynomial* rewriting techniques for both the data and the query in the case when ontologies are formulated in *DL-Lite*$^{\mathcal{N}}_{horn}$, which properly contains *DL-Lite*$_{\sqcap,\mathcal{F}}$ (Calvanese et al. 2006) and is among the most commonly used *DL-Lite* dialects without role inclusions. We also consider *DL-Lite*$^{(\mathcal{HN})}_{horn}$, the extension of *DL-Lite*$^{\mathcal{N}}_{horn}$ with role inclusions, which covers the vast majority of *DL-Lite* ontologies used in practice. In this case, we could not avoid an exponential blowup of the rewritten queries (only in the number of roles), while keeping the expanded data polynomial. On the positive side, the exponential rewriting allows us to answer a wider class of positive existential queries, which properly includes the conjunctive queries. To evaluate the new techniques, we have conducted experiments with real-world *DL-Lite* ontologies, which demonstrate that the combined approach outperforms pure query rewriting. It is to be noted, however, that these amenities come at a price: in general, the combined approach is applicable only if the information system is allowed to manipulate the source data, which may not be the case in some information integration scenarios.

To explain our approach in more detail, suppose that we want to answer conjunctive queries over a data instance $\mathcal{D}$ given an ontology $\mathcal{T}$. As a first step, we expand $\mathcal{D}$ by 'applying' the axioms of $\mathcal{T}$, which gives a new data instance $\mathcal{D}'$ of size $\mathcal{O}(|\mathcal{D}| \cdot |\mathcal{T}| + |\mathcal{T}|^2)$ in the worst case. Then, given a conjunctive query q to be executed over $\mathcal{D}$ and $\mathcal{T}$, we rewrite q into a first-order query $q^\dagger$ over $\mathcal{D}'$ (independently of $\mathcal{D}$ and 'almost' independently of $\mathcal{T}$) whose size is $\mathcal{O}(|q|)$ for *DL-Lite*$^{\mathcal{N}}_{horn}$ ontologies. The rewriting $q^\dagger$ is fundamentally different from the rewriting of (Kontchakov et al. 2009), which involves an exponential blowup in the worst case. For a *DL-Lite*$^{(\mathcal{HN})}_{horn}$ ontology $\mathcal{T}$, we can reduce conjunctive query answering to the case without role inclusions, but the resulting query may have to be a union of $r^{|q|}$ queries of the form $q^\dagger$, where r is the maximum number of subroles of a role occurring in q. We also show that the expansion of data can be implemented using views in the RDBMS; this has a convenient side-effect that the expanded database can be automatically and transparently adjusted when the underlying data is updated. In addition, the view-based construction yields a novel way for pure query rewriting for *DL-Lite*$^{\mathcal{N}}_{horn}$. When applied to *DL-Lite*$_{\mathcal{F}}$ (Calvanese et al. 2006), which disallows conjunction and all number restrictions except (unqualified) existential quantifiers and functionality constraints, this new technique blows up the rewritten query only polynomially. As views are no longer involved in this case, we obtain the first polynomial pure query rewriting technique for a *DL-Lite* logic.

Preliminaries

We briefly introduce *DL-Lite*$^{(\mathcal{HN})}_{horn}$, the most expressive dialect of *DL-Lite* for which positive existential query answering is in AC^0 for data complexity under the unique name assumption, along with its fragments that are considered in this paper, such as *DL-Lite*$^{\mathcal{N}}_{horn}$. For more details and the relation to other *DL-Lite* logics, we refer the interested reader to (Artale et al. 2009). Let $\mathsf{N_I}$, $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite sets of *individual names*, *concept names* and *role names*. *Roles* R and *concepts* C are built according to the following syntax rules, where $P \in \mathsf{N_R}$, $A \in \mathsf{N_C}$ and $m > 0$:

$$R \ ::= \ P \ \mid \ P^-, \qquad C \ ::= \ \bot \ \mid \ A \ \mid \ \geq m\, R.$$

As usual, we write $\exists R$ for $\geq 1\, R$ and identify $(P^-)^-$ with P. We use $\mathsf{N_R^-}$ to denote the set of all roles. In DL, ontologies are represented as TBoxes. A *DL-Lite*$^{(\mathcal{HN})}_{horn}$ *TBox* is a finite set $\mathcal{T}$ of *concept* and *role inclusions* (*CIs* and *RIs*), which take the form $C_1 \sqcap \cdots \sqcap C_n \ \sqsubseteq \ C$ and $R_1 \sqsubseteq R_2$, respectively. Denote by $\sqsubseteq^*_{\mathcal{T}}$ the transitive-reflexive closure of the RIs in $\mathcal{T}$. It is required that if $S \sqsubseteq^*_{\mathcal{T}} R$ with $S \neq R$, then $\mathcal{T}$ does not contain a CI with $C_i = \ \geq m\, R$, for $m \geq 2$, on its left-hand side. Without this restriction, conjunctive query answering becomes coNP-hard for data complexity (Artale et al. 2009), which means that the combined approach is no longer possible without an exponential blowup of the data (Lutz, Toman, and Wolter 2009). *DL-Lite*$^{\mathcal{N}}_{horn}$ is the fragment of *DL-Lite*$^{(\mathcal{HN})}_{horn}$ in which RIs are disallowed.

An *ABox* is used to store instance data. Formally, it is a finite set of *concept assertions* $A(a)$ and *role assertions* $P(a, b)$, where $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. We denote by $\mathsf{Ind}(\mathcal{A})$ the set of individual names occurring in $\mathcal{A}$, and often write $P^-(a, b) \in \mathcal{A}$ instead of $P(b, a) \in \mathcal{A}$. A *knowledge base* (*KB*) is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox.

The semantics of *DL-Lite*$^{(\mathcal{HN})}_{horn}$ is defined in the standard way based on interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ an *interpretation function* that maps each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each $P \in \mathsf{N_R}$ to a relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Throughout the paper, we mostly adopt the *unique name assumption* (UNA), i.e., require that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for distinct $a, b \in \mathsf{N_I}$. In the context of OWL, the UNA is not adopted. The combined approach to the case without the UNA is discussed later on in the paper. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts and roles by setting

$$(P^-)^{\mathcal{I}} = \{(e, d) \mid (d, e) \in P^{\mathcal{I}}\},$$

$$\bot^{\mathcal{I}} = \emptyset, \qquad (\geq m\, R)^{\mathcal{I}} = \{d \mid \#\{e \mid (d, e) \in R^{\mathcal{I}}\} \geq m\},$$

where $\#X$ is the cardinality of X. Given an interpretation $\mathcal{I}$, we write $\mathcal{I} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ if $\bigcap_{i=1}^n C_i^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, $\mathcal{I} \models R_1 \sqsubseteq R_2$ if $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$, $\mathcal{I} \models A(a)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and $\mathcal{I} \models P(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$. The interpretation $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{T} \cup \mathcal{A}$. $\mathcal{K}$ is *consistent* if it has a model. We write $\mathcal{K} \models \alpha$ whenever $\mathcal{I} \models \alpha$ for all models $\mathcal{I}$ of $\mathcal{K}$.

Let $\mathsf{N_V}$ be a countably infinite set of *variables*. Taken together, the sets $\mathsf{N_V}$ and $\mathsf{N_I}$ form the set $\mathsf{N_T}$ of *terms*. A *first-order* (*FO*) *query* is a first-order formula $q = \varphi(\vec{v})$ in the signature $\mathsf{N_C} \cup \mathsf{N_R}$ with terms from $\mathsf{N_T}$, where the concept and role names are treated as unary and binary predicates, respectively, and the sequence $\vec{v} = v_1, \ldots, v_k$ of variables from $\mathsf{N_V}$ contains all the free variables of φ. The variables $\vec{v}$ are called the *answer variables* of q, and q is *k-ary* if $\vec{v}$ comprises k variables. A *positive existential query* is a first-order query of the form $q = \exists \vec{u}\, \psi(\vec{u}, \vec{v})$, where ψ is constructed using conjunction and disjunction from *concept atoms* $A(t)$ and *role atoms* $P(t, t')$ with $t, t' \in \mathsf{N_T}$. As in the case of ABox assertions, we can write $P^-(t, t')$ instead of $P(t', t)$. A *conjunctive query* (*CQ*) is a positive existential query containing no disjunction. The variables in $\vec{u}$ are called the *quantified variables* of q. We denote by $\mathsf{qvar}(q)$ the set of quantified variables $\vec{u}$, by $\mathsf{avar}(q)$ the set of answer variables $\vec{v}$, and by $\mathsf{term}(q)$ the set of terms in q.

Let $q = \varphi(\vec{v})$ be a k-ary FO query and $\mathcal{I}$ an interpretation. A map $\pi\colon \mathsf{term}(q) \to \Delta^{\mathcal{I}}$ with $\pi(a) = a^{\mathcal{I}}$, for $a \in \mathsf{term}(q) \cap \mathsf{N_I}$, is called a *match* for q in $\mathcal{I}$ if $\mathcal{I}$ satisfies q under the variable assignment that maps each $v \in \mathsf{avar}(q)$ to $\pi(v)$; in this case we write $\mathcal{I} \models^{\pi} q$. For a k-tuple of individual names $\vec{a} = a_1, \dots, a_k$, a match π for q in $\mathcal{I}$ is called an *$\vec{a}$-match* if $\pi(v_i) = a_i^{\mathcal{I}}$, $i \le k$. We say that $\vec{a}$ is an *answer* to q in an interpretation $\mathcal{I}$ if there is an $\vec{a}$-match for q in $\mathcal{I}$ and use $\mathsf{ans}(q, \mathcal{I})$ to denote the set of all answers to q in $\mathcal{I}$. We say that $\vec{a}$ is a *certain answer* to q over a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $\vec{a} \subseteq \mathsf{Ind}(\mathcal{A})$ and $\mathcal{I} \models q[\vec{a}]$ for all models $\mathcal{I}$ of $\mathcal{K}$. The set of all certain answers to q over $\mathcal{K}$ is denoted by $\mathsf{cert}(q, \mathcal{K})$.

Throughout the paper, we use $|\mathcal{K}|$ to denote the *size* of a KB $\mathcal{K}$, that is, the number of symbols required to write $\mathcal{K}$. $|\mathcal{T}|$, $|\mathcal{A}|$ and $|q|$ are defined analogously. Unless otherwise stated, we assume numbers to be encoded in binary, and thus $|\ge m\, R| = \mathcal{O}(\log m)$.

ABox Extension

First, we describe the ABox extension part of the combined approach to CQ answering in *DL-Lite*$^{\mathcal{N}}_{horn}$. Semantically, the ABox extension means expanding the ABox $\mathcal{A}$ to a *canonical interpretation* $\mathcal{I}_{\mathcal{K}}$ for the given KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. More specifically, $\mathcal{I}_{\mathcal{K}}$ is constructed by (i) expanding the set $\mathsf{Ind}(\mathcal{A})$ of individual names in $\mathcal{A}$ with additional individuals to witness existential and number restrictions, and (ii) expanding the extensions of concept and role names as required by the CIs in $\mathcal{T}$. The individual names in (i) are taken from the set $\mathsf{N}_\mathsf{I}^{\mathcal{T}} = \{c_P, c_{P^-} \mid P \text{ a role name in } \mathcal{T}\}$, which is assumed to be disjoint from $\mathsf{Ind}(\mathcal{A})$. The domain of $\mathcal{I}_{\mathcal{K}}$ will contain those witnesses c_R that are really needed in any model of $\mathcal{K}$. To identify such witnesses, we require the following definition. A role R is called *generating in $\mathcal{K}$* if there exist $a \in \mathsf{Ind}(\mathcal{A})$ and $R_1, \dots, R_n = R$ such that the following conditions hold:

(agen) $\mathcal{K} \models \exists R_1(a)$ but $R_1(a, b) \notin \mathcal{A}$ for all $b \in \mathsf{Ind}(\mathcal{A})$ (written $a \rightsquigarrow c_{R_1}$),

(rgen) for $i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$ (written $c_{R_i} \rightsquigarrow c_{R_{i+1}}$).

It can be seen that R is generating in $\mathcal{K}$ if, and only if, every model $\mathcal{I}$ of $\mathcal{K}$ contains some point $x \in \Delta^{\mathcal{I}}$ with an incoming R-arrow, but there is a model $\mathcal{I}$ where no such x is identified by an individual name in the ABox.

The *canonical interpretation $\mathcal{I}_{\mathcal{K}}$ for $\mathcal{K}$* is defined as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}_{\mathcal{K}}} &= \mathsf{Ind}(\mathcal{A}) \cup \{c_R \mid R \in \mathsf{N}_\mathsf{R}^-,\ \ R \text{ is generating in } \mathcal{K}\},\\
a^{\mathcal{I}_{\mathcal{K}}} &= a, \text{ for all } a \in \mathsf{Ind}(\mathcal{A}),\\
A^{\mathcal{I}_{\mathcal{K}}} &= \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \\
&\quad\ \{c_R \in \Delta^{\mathcal{I}_{\mathcal{K}}} \mid \mathcal{T} \models \exists R^- \sqsubseteq A\},\\
P^{\mathcal{I}_{\mathcal{K}}} &= \{(a, b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A}\} \cup \\
&\quad\ \{(d, c_P) \in \Delta^{\mathcal{I}_{\mathcal{K}}} \times \mathsf{N}_\mathsf{I}^{\mathcal{T}} \mid d \rightsquigarrow c_P\} \cup \\
&\quad\ \{(c_{P^-}, d) \in \mathsf{N}_\mathsf{I}^{\mathcal{T}} \times \Delta^{\mathcal{I}_{\mathcal{K}}} \mid d \rightsquigarrow c_{P^-}\}.
\end{aligned}$$

Clearly, $\mathcal{I}_{\mathcal{K}}$ is an extension of the ABox $\mathcal{A}$ if we represent the concept and role memberships in the form of ABox assertions. The number of domain elements in $\mathcal{I}_{\mathcal{K}}$ does not exceed $|\mathcal{K}|$.

The canonical interpretation $\mathcal{I}_{\mathcal{K}}$ is *not* in general a model of $\mathcal{K}$. Indeed, this cannot be the case because *DL-Lite*$^{\mathcal{N}}_{horn}$ does not enjoy the finite model property (Calvanese et al. 2005) whereas $\mathcal{I}_{\mathcal{K}}$ is always finite.

Example 1 Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where

$$\mathcal{T} = \{A \sqsubseteq \exists P,\ \ge 2\, P^- \sqsubseteq \bot\}, \quad \mathcal{A} = \{A(a),\ A(b)\}.$$

Then $\Delta^{\mathcal{I}_{\mathcal{K}}} = \{a, b, c_P\}$, $P^{\mathcal{I}_{\mathcal{K}}} = \{(a, c_P), (b, c_P)\}$, and so $c_P \in (\ge 2\, P^-)^{\mathcal{I}_{\mathcal{K}}}$ and $\mathcal{I}_{\mathcal{K}} \not\models \mathcal{K}$.

As far as query answering is concerned, this is not a problem. More important is that $\mathcal{I}_{\mathcal{K}}$ does not always give the correct answers to CQs, i.e., it is *not* the case that $\mathcal{I}_{\mathcal{K}} \models q[\vec{a}]$ iff $\mathcal{K} \models q[\vec{a}]$ for all k-ary CQs q and k-tuples $\vec{a} \subseteq \mathsf{Ind}(\mathcal{A})$. We illustrate this with two examples.

Example 2 Consider the 'cyclic' query $q = \exists v\, P(v, v)$ over $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where

$$\mathcal{T} = \{A \sqsubseteq \exists P,\ \exists P^- \sqsubseteq \exists P\}, \quad \mathcal{A} = \{A(a)\}.$$

Then $\Delta^{\mathcal{I}_{\mathcal{K}}} = \{a, c_P\}$, $P^{\mathcal{I}_{\mathcal{K}}} = \{(a, c_P), (c_P, c_P)\}$ and $A^{\mathcal{I}_{\mathcal{K}}} = \{a\}$. The assignment π defined by $\pi(v) = c_P$ shows that $\mathcal{I}_{\mathcal{K}} \models q$. On the other hand, the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a, 1, 2, \dots\}$, $P^{\mathcal{I}} = \{(a, 1)\} \cup \{(n, n+1) \mid n \ge 1\}$ and $A^{\mathcal{I}} = \{a\}$ is a model of $\mathcal{K}$, but $\mathcal{I} \not\models q$. Thus $\mathcal{K} \not\models q$.

Example 3 Consider next the 'fork-shaped' query

$$q = \exists v_2\, (P(v_1, v_2) \land P(v_3, v_2))$$

over $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where

$$\mathcal{T} = \{A \sqsubseteq \exists P\}, \quad \mathcal{A} = \{A(a), A(b)\}.$$

Then $\Delta^{\mathcal{I}_{\mathcal{K}}} = \{a, b, c_P\}$, $P^{\mathcal{I}_{\mathcal{K}}} = \{(a, c_P), (b, c_P)\}$ and $A^{\mathcal{I}_{\mathcal{K}}} = \{a, b\}$. Clearly, $\mathcal{I}_{\mathcal{K}} \models q[a, b]$. On the other hand, the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a, b, c_1, c_2\}$, $A^{\mathcal{I}} = \{a, b\}$, and $P^{\mathcal{I}} = \{(a, c_1), (b, c_2)\}$ is a model of $\mathcal{K}$ such that $\mathcal{I} \not\models q[a, b]$. Thus $\mathcal{K} \not\models q[a, b]$.

We overcome these problems in two steps. First, we show that the unravelling of $\mathcal{I}_{\mathcal{K}}$ into a forest-shaped interpretation $\mathcal{U}_{\mathcal{K}}$ *does* give the right answers to queries. However, $\mathcal{U}_{\mathcal{K}}$ may be infinite, and so we cannot store it as a database instance. But, as shown in the next section, any given CQ q can be rewritten into an FO query $q^{\dagger}$ in such a way that the answers to q over $\mathcal{U}_{\mathcal{K}}$ are identical to the answers to $q^{\dagger}$ over $\mathcal{I}_{\mathcal{K}}$. This enables us to use the *finite* interpretation $\mathcal{I}_{\mathcal{K}}$ as a relational instance and still obtain correct answers to queries.

The unravelling $\mathcal{U}_{\mathcal{K}}$ is defined as follows. A *path* in $\mathcal{I}_{\mathcal{K}}$ is a finite sequence $ac_{R_1} \cdots c_{R_n}$, $n \ge 0$, such that $a \in \mathsf{Ind}(\mathcal{A})$

and $R_1, \ldots, R_n$ satisfy **(agen)** and **(rgen)** (that is, $a \rightsquigarrow c_{R_1}$ and $c_{R_i} \rightsquigarrow c_{R_{i+1}}$, for $1 \leq i < n$). We denote by $\mathsf{tail}(\sigma)$ the last element in a path σ. The interpretation $\mathcal{U}_\mathcal{K}$ is then defined by taking:

$$\begin{aligned}
\Delta^{\mathcal{U}_\mathcal{K}} &= \{a \cdot c_{R_1} \cdots c_{R_n} \mid a \in \mathsf{Ind}(\mathcal{A}), n \geq 0, \\
&\qquad\qquad a \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n}\}, \\
a^{\mathcal{U}_\mathcal{K}} &= a, \text{ for all } a \in \mathsf{Ind}(\mathcal{A}), \\
A^{\mathcal{U}_\mathcal{K}} &= \{\sigma \in \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathsf{tail}(\sigma) \in A^{\mathcal{I}_\mathcal{K}}\}, \\
P^{\mathcal{U}_\mathcal{K}} &= \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \\
&\quad \cup \{(\sigma, \sigma \cdot c_P) \in \Delta^{\mathcal{U}_\mathcal{K}} \times \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathsf{tail}(\sigma) \rightsquigarrow c_P\} \\
&\quad \cup \{(\sigma \cdot c_{P^-}, \sigma) \in \Delta^{\mathcal{U}_\mathcal{K}} \times \Delta^{\mathcal{U}_\mathcal{K}} \mid \mathsf{tail}(\sigma) \rightsquigarrow c_{P^-}\},
\end{aligned}$$

where '$\cdot$' denotes concatenation. Notice that the interpretations $\mathcal{I}$ constructed in Examples 2 and 3 are isomorphic to the respective unravellings $\mathcal{U}_\mathcal{K}$. The interpretation $\mathcal{U}_\mathcal{K}$ is forest-shaped in the sense that the graph $G = (V, E)$ with $V = \Delta^{\mathcal{U}_\mathcal{K}}$ and $E = \{(\sigma, \sigma \cdot c_R) \mid \mathsf{tail}(\sigma) \rightsquigarrow c_R\}$ is a forest. The map $\tau\colon \Delta^{\mathcal{U}_\mathcal{K}} \to \Delta^{\mathcal{I}_\mathcal{K}}$ defined by taking $\tau(\sigma) = \mathsf{tail}(\sigma)$ is a homomorphism from $\mathcal{U}_\mathcal{K}$ onto $\mathcal{I}_\mathcal{K}$, and so $\mathcal{U}_\mathcal{K} \models q$ implies $\mathcal{I}_\mathcal{K} \models q$ for all CQs q (but, as shown in Examples 2 and 3, not necessarily vice versa). Just like $\mathcal{I}_\mathcal{K}$, the interpretation $\mathcal{U}_\mathcal{K}$ is *not* in general a model of $\mathcal{K}$. A simple example is given by $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, $\mathcal{T} = \{A \sqsubseteq\ \geq 2\, P\}$ and $\mathcal{A} = \{A(a)\}$, where we have $\mathcal{U}_\mathcal{K} \not\models A \sqsubseteq\ \geq 2P$ because a is P-related only to $a \cdot c_P$ in $\mathcal{U}_\mathcal{K}$. Nevertheless, $\mathcal{U}_\mathcal{K}$ gives the right answers to all CQs, as shown by the following:

Theorem 4 *For every consistent DL-Lite$^\mathcal{N}_{horn}$ KB $\mathcal{K}$ and every CQ q, we have* $\mathsf{cert}(q, \mathcal{K}) = \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.

The proof of this theorem, as well as all other omitted proofs, can be found in Appendix: Proof of Theorem 4.

Note that Theorem 4 requires consistency of $\mathcal{K}$. We shall see below that there is an FO query $q^\mathcal{T}_\bot$ such that $\mathsf{ans}(q^\mathcal{T}_\bot, \mathcal{A}) = \emptyset$ iff $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent. This will allow us to check consistency using an RDBMS before building the canonical interpretation.

Query Rewriting for *DL-Lite$^\mathcal{N}_{horn}$*

We now present a polytime algorithm that rewrites every CQ q into an FO query $q^\dagger$ such that

- $\mathsf{ans}(q, \mathcal{U}_\mathcal{K}) = \mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K})$,
- the length of $q^\dagger$ is $\mathcal{O}(|q| \cdot |\mathcal{T}|)$ (and can be made $\mathcal{O}(|q|)$ by adding an auxiliary database relation).

By Theorem 4, we can compute the answers to q over $\mathcal{K}$ using an RDBMS to execute $q^\dagger$ over $\mathcal{I}_\mathcal{K}$ stored as a relational instance.

To simplify notation, we often identify a CQ q with the set of its atoms and use $P^-(v, u) \in q$ as a synonym of $P(u, v) \in q$. The rewriting $q^\dagger$ of the CQ $q = \exists \vec{u}\, \varphi$ is defined as a formula of the form

$$q^\dagger = \exists \vec{u}\, (\varphi \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3),$$

where φ_1, φ_2 and φ_3 are Boolean combinations of equalities $t_1 = t_2$ and each of the t_i is either a term in q or a constant $c_R \in \mathsf{N}^\mathcal{T}_\mathsf{I}$.

The purpose of φ_1 is to select only those matches where all answer variables receive values from $\mathsf{Ind}(\mathcal{A})$, as required by the definition of certain answers:

$$\varphi_1 = \bigwedge_{v \in \mathsf{avar}(q)} \; \bigwedge_{c_R \in \mathsf{N}^\mathcal{T}_\mathsf{I}} (v \neq c_R).$$

The intuition behind φ_2 and φ_3 is that the rewriting $q^\dagger$ of q has to select exactly those matches of q in $\mathcal{I}_\mathcal{K}$ that can be 'reproduced' as matches in $\mathcal{U}_\mathcal{K}$. Due to the forest structure of $\mathcal{U}_\mathcal{K}$, this requirement imposes strong constraints on the way in which variables can be matched to the non-ABox elements of $\mathcal{I}_\mathcal{K}$. Essentially, the part of q that is mapped to the non-ABox elements of $\mathcal{I}_\mathcal{K}$ must be homomorphically embeddable into a forest. This intuition is captured by the following definition. Let $(\mathsf{N}^-_\mathsf{R})^*$ be the set of all finite words over N^-_R (including the empty word ε).

Definition 5 Let $q = \exists \vec{u}\, \varphi$ be a CQ with $R(t, t') \in q$. A partial function $f_{R,t}\colon \mathsf{term}(q) \to (\mathsf{N}^-_\mathsf{R})^*$ is a *tree witness for R and t in q* if its domain is minimal (with respect to set-theoretic inclusion) such that the following conditions hold, where $w \in (\mathsf{N}^-_\mathsf{R})^*$:

- $f_{R,t}(t) = \varepsilon$;
- if $f_{R,t}(s) = \varepsilon$ and $R(s, s') \in q$ then $f_{R,t}(s') = R$;
- if $f_{R,t}(s) = w \cdot S$ and $S'(s, s') \in q$ with $S' \neq S^-$ then $f_{R,t}(s') = w \cdot S \cdot S'$;
- if $f_{R,t}(s) = w \cdot S$ and $S^-(s, s') \in q$ then $f_{R,t}(s') = w$.

Note that, for $R(t, t') \in q$, a tree witness $f_{R,t}$ does not necessarily exist. However, if it does exist then it is clearly unique. The following lemma shows that tree witnesses can be efficiently computed.

Lemma 6 *Given a CQ q and $R(t, t') \in q$, it can be decided in time $\mathcal{O}(|q|)$ whether a tree witness $f_{R,t}$ exists. And if it does exist, $f_{R,t}$ can be computed in time $\mathcal{O}(|q|)$.*

Proof. Follows from Lemma 15 (to be proved in Appendix: Proof of Theorem 10), which provides a step-by-step procedure for constructing a tree witness, with the steps mimicking the four rules above. This procedure may fail in the construction process, in which case no tree witness exists. ❑

Intuitively, the tree witness $f_{R,t}$ deals with matches π in $\mathcal{I}_\mathcal{K}$, where for some $R(t, t') \in q$, we have $\pi(t') = c_R$. The reproduction π' of this match in $\mathcal{U}_\mathcal{K}$ has to satisfy $\pi'(t') = \sigma \cdot c_R$ for some σ. Due to the condition $R^-_i \neq R_{i+1}$ in **(rgen)**, we have $\sigma \cdot c_R \cdot c_{R^-} \notin \Delta^{\mathcal{U}_\mathcal{K}}$. By the definition of $\mathcal{U}_\mathcal{K}$, $(\pi'(t), \sigma \cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$ thus implies $\pi'(t) = \sigma$. Now, $f_{R,t}$ serves two purposes. First, it identifies, via its domain $\mathfrak{D} \subseteq \mathsf{term}(q)$, those terms in q that *must* be mapped by π' to the subtree of $\mathcal{U}_\mathcal{K}$ with root $\pi'(t)$. Second, it describes a homomorphic embedding of $q|_\mathfrak{D}$ (i.e., q restricted to $\mathfrak{D}$) into that subtree: $f_{R,t}(s) = R_1 \cdots R_k$ means that $\pi'(s) = \pi(t) \cdot c_{R_1} \cdots c_{R_k}$. Due to the structure of $\mathcal{U}_\mathcal{K}$, it can actually be seen that the described homomorphic embedding is the *only* possible embedding of this kind. Therefore, if the tree witness for $R(t, t')$ does not exist, then no homomorphic embedding is possible, which means that t' cannot be mapped to $\sigma \cdot c_R$ in $\mathcal{U}_\mathcal{K}$, and so not to c_R in $\mathcal{I}_\mathcal{K}$ either. This is precisely what φ_2 is for:

$$\varphi_2 = \bigwedge_{\substack{R(t,t')\in q \\ f_{R,t} \text{ does not exist}}} (t' \neq c_R).$$

Example 7 We illustrate φ_2 using the query $q = \exists v\, P(v, v)$ and KB $\mathcal{K}$ from Example 2. As we saw, $\mathcal{K} \not\models q$ but $\mathcal{I}_\mathcal{K} \models q$ since $(c_P, c_P) \in P^{\mathcal{I}_\mathcal{K}}$. Now observe that there exists no tree witness $f_{P,v}$ because otherwise we would have $f_{P,v}(v) = \varepsilon$ and, by $P(v, v) \in q$, also $f_{P,v}(v) = P$, contrary to $f_{P,v}$ being a function. Thus, $v \neq c_P$ is a conjunct of φ_2, and so $\mathcal{I}_\mathcal{K} \not\models q^\dagger$. In general, the variable v can only be matched to an ABox individual no matter what $\mathcal{K}$ is: both $v \neq c_P$ and $v \neq c_{P^-}$ are conjuncts of φ_2 and, by the definition of $\mathcal{I}_\mathcal{K}$, we have $(c_R, c_R) \notin P^{\mathcal{I}_\mathcal{K}}$ for all $\mathcal{K}$ and $R \notin \{P, P^-\}$.

If $f_{R,t}$ exists and $R(t, t') \in q$ then, in principle, t' can be mapped to a c_R in $\mathcal{I}_\mathcal{K}$. But then we still have to ensure that all terms in the domain $\mathfrak{D}$ of $f_{R,t}$ are matched in $\mathcal{I}_\mathcal{K}$ in a way that can be reproduced in the relevant subtree of $\mathcal{U}_\mathcal{K}$ by a homomorphic embedding. As already mentioned, the only possible such embedding is the one described by $f_{R,t}$, and thus it suffices to ensure that if t' is mapped to c_R, then each term s in the domain of $f_{R,t}$ is mapped to $f_{R,t}(s)$. To achieve this using a conjunct φ_3 of only *linear* size, we first define an appropriate equivalence relation on terms. For $s, t \in \mathsf{term}(q)$ and $R \in \mathsf{N}_\mathsf{R}^-$, we write $s \equiv_q^R t$ if there are atoms $R(s, s'), R(t, t') \in q$, $f_{R,s}$ exists and $f_{R,s}(t) = \varepsilon$.

Lemma 8 *For all* $R \in \mathsf{N}_\mathsf{R}^-$, $\equiv_q^R$ *is an equivalence relation.*

Proof. Reflexivity of $\equiv_q^R$ is trivial by the definition of tree witnesses, while both symmetry and transitivity are immediate consequence of the following observation.

Suppose that $f_{R,t}(s)$ is defined. Then there are atoms $R_i(t_{i-1}, t_i) \in q$, $0 < i \leq n$, with $R = R_1$, $t = t_0$, $s = t_n$ such that the following property holds: if $f_{R_1,t_0}(t_j) = \varepsilon$, for $0 < j \leq n$, then $R_j = R_1^-$ and, in the case when $j < n$, we also have $R_{j+1} = R_1$. Moreover, if $f_{R_1,t_0}(t_n) = \varepsilon$ then f_{R_1,t_n} is clearly defined (remember that in this case $R_n = R_1^-$) and $f_{R_1,t_n}(t_0) = \varepsilon$. To see this, it suffices to observe that to compute $f_{R_1,t_0}(t_n)$ we can take the word $R_1 \cdot R_2 \cdots R_n$ and then successively remove from it all the *leftmost* pairs of the form $R \cdot R^-$ (cf. reductions in free groups). If in the removal process we obtain a word of the form $R_i \cdot R_j \cdot w$ with $R_i = R_j^-$ then $f_{R_1,t_0}(t_j) = \varepsilon$. To compute $f_{R_1,t_n}(t_0)$, we can take the same word $R_1 \cdot R_2 \cdots R_n$, successively remove from it all the *rightmost* pairs of the form $R \cdot R^-$, and then take the inverse. ❑

For $R \in \mathsf{N}_\mathsf{R}^-$ and $t \in \mathsf{term}(q)$, denote by $[t]_q^R$ the equivalence class of $\equiv_q^R$ generated by t. We can now define φ_3 with the required properties by taking:

$$\varphi_3 = \bigwedge_{\substack{[t]_q^R \\ f_{R,t} \text{ exists}}} \Big(\bigvee_{\substack{R(s,s')\in q \\ s \equiv_q^R t}} (s' = c_R) \;\rightarrow\; \bigwedge_{s \equiv_q^R t} (s = t) \Big).$$

Example 9 We illustrate φ_3 using the 'fork-shaped' query $q = \exists v_2\, (P(v_1, v_2) \wedge P(v_3, v_2))$ from Example 3. For every KB $\mathcal{K}$, if $\mathcal{U}_\mathcal{K} \models q[a_1, a_3]$ and v_2 is not mapped to an ABox individual, then $a_1 = a_3$. This is not the case in $\mathcal{I}_\mathcal{K}$ as, depending on $\mathcal{K}$, we might be able to map v_2 to c_P. Thus, it is sufficient to add $(v_2 = c_P) \rightarrow (v_1 = v_3)$ as a conjunct to $P(v_1, v_2) \wedge P(v_3, v_2)$. This is achieved using φ_3 as follows: f_{P,v_1} exists and $[v_1]_q^P = \{v_1, v_3\}$; therefore, φ_3 contains the conjunct $(v_2 = c_P) \rightarrow (v_1 = v_3)$, as required.

It is easy to see that $q^\dagger$ can be computed in polynomial time in the size of the query q and the set $\mathsf{N}_\mathsf{I}^\mathcal{T}$. The length of $q^\dagger$ is $\mathcal{O}(|q| \cdot |\mathcal{T}|)$ since φ_1 is of length $\mathcal{O}(|q| \cdot |\mathcal{T}|)$ and φ_2, φ_3 are of length $\mathcal{O}(|q|)$. If we add a unary database table aux that identifies exactly the elements of $\mathsf{N}_\mathsf{I}^\mathcal{T}$, then we can replace φ_1 with

$$\varphi_1' = \bigwedge_{v \in \mathsf{avar}(q)} \neg \mathsf{aux}(v)$$

and obtain $q^\dagger$ of size $\mathcal{O}(|q|)$.

The main result of this paper is the following theorem:

Theorem 10 *For every DL-Lite*$_{horn}^{\mathcal{N}}$ *KB* $\mathcal{K}$ *and every CQ* q, $\mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K}) = \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.

The proof is given in Appendix: Proof of Theorem 10.

Canonical Interpretation by FO Queries

The aim of this section is to show that the canonical interpretation $\mathcal{I}_\mathcal{K}$ for a *DL-Lite*$_{horn}^{\mathcal{N}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ can be constructed by means of FO queries. This will allow us to implement the construction of $\mathcal{I}_\mathcal{K}$ in an RDBMS using (materialised) views. The benefit is that *updates* of the ABox $\mathcal{A}$, such as insertions and deletions, are automatically reflected in those views, which solves the problem of updating $\mathcal{I}_\mathcal{K}$ in a simple and elegant way.

Given a *DL-Lite*$_{horn}^{\mathcal{N}}$ TBox $\mathcal{T}$ and concept and role names A and P, we construct FO queries $q_A^\mathcal{T}(x)$ and $q_P^\mathcal{T}(x, y)$ such that the answers to $q_A^\mathcal{T}$ and $q_P^\mathcal{T}$ over $\mathcal{A}$ coincide with $A^{\mathcal{I}_\mathcal{K}}$ and, respectively, $P^{\mathcal{I}_\mathcal{K}}$ for all ABoxes $\mathcal{A}$ and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. It will be convenient for us to regard $\mathcal{A}$ as an interpretation $\mathcal{I}_\mathcal{A}$ that includes all elements of $\mathsf{N}_\mathsf{I}^\mathcal{T}$ as domain elements which are not involved in any concept or role memberships:

- $\Delta^{\mathcal{I}_\mathcal{A}} = \mathsf{Ind}(\mathcal{A}) \cup \mathsf{N}_\mathsf{I}^\mathcal{T}$;
- $A^{\mathcal{I}_\mathcal{A}} = \{a \mid A(a) \in \mathcal{A}\}$, for all $A \in \mathsf{N}_\mathsf{C}$;
- $P^{\mathcal{I}_\mathcal{A}} = \{(a, b) \mid P(a, b) \in \mathcal{A}\}$, for all $P \in \mathsf{N}_\mathsf{R}$.

Instead of constructing $\Delta^{\mathcal{I}_\mathcal{A}}$ explicitly for evaluating the subsequent queries in a relational system, we could rely on domain independence of relational queries (Abiteboul et al. 1995).

We now construct $q_A^\mathcal{T}(x)$ in three steps. First, for each concept C in $\mathcal{T}$, we inductively define a query $\mathsf{exp}_C^\mathcal{T}(x)$ whose answers on $\mathcal{I}_\mathcal{A}$ determine $C^{\mathcal{I}_\mathcal{K}}$ when restricted to $\Delta^{\mathcal{I}_\mathcal{K}}$ (this restriction will be ensured in the third step). To simplify presentation, we assume that $\mathcal{T}$ contains all CIs of the form $\geq m\, R \sqsubseteq\, \geq m'\, R$, for pairs of concepts $\geq m\, R$ and $\geq m'\, R$ in $\mathcal{T}$ such that $m' < m$, and no $\geq m''\, R$, for $m' < m'' < m$, occurs in $\mathcal{T}$. Clearly, the extended TBox is only linearly larger than the original one. Set

$$\mathsf{exp}_{\bot}^{\mathcal{T},0}(x) = \bot, \qquad \mathsf{exp}_{A}^{\mathcal{T},0}(x) = A(x),$$
$$\mathsf{exp}_{\exists R}^{\mathcal{T},0}(x) = (x = c_{R^-}) \vee \exists y\, R(x,y),$$
$$\mathsf{exp}_{\geq mR}^{\mathcal{T},0}(x) = \exists y_1,\dots,y_m \,\big(\bigwedge_{1\leq i\leq m} R(x,y_i) \wedge \bigwedge_{i\neq j}(y_i \neq y_j)\big),$$

where $m \geq 2$, and for $j \geq 1$, set

$$\mathsf{exp}_{C}^{\mathcal{T},j}(x) = \mathsf{exp}_{C}^{\mathcal{T},j-1}(x) \;\vee \bigvee_{C_1\sqcap\dots\sqcap C_n \sqsubseteq C} \bigwedge_{i=1}^{n} \mathsf{exp}_{C_i}^{\mathcal{T},j-1}(x).$$

Thus, $\mathsf{exp}_{C}^{\mathcal{T},j}(x)$ adds to $\mathsf{exp}_{C}^{\mathcal{T},j-1}(x)$ those elements of $\Delta^{\mathcal{I}_\mathcal{A}}$ that can be obtained by one inference step of SLD resolution (Kowalski and Kuehner 1971). As we do not need more than $|\mathcal{T}|$ inference steps, the formulas $\mathsf{exp}_{C}^{\mathcal{T},j}(x)$ are all equivalent for $j \geq |\mathcal{T}|$. We set $\mathsf{exp}_{C}^{\mathcal{T}}(x) = \mathsf{exp}_{C}^{\mathcal{T},|\mathcal{T}|}(x)$.

Lemma 11 *For a DL-Lite$^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ and a concept C in $\mathcal{T}$, $\mathsf{ans}(\mathsf{exp}_{C}^{\mathcal{T}}, \mathcal{I}_\mathcal{A}) \cap \Delta^{\mathcal{I}_\mathcal{K}} = C^{\mathcal{I}_\mathcal{K}}$ for all KBs $\mathcal{K} = (\mathcal{T},\mathcal{A})$.*

Second, we construct queries $q_P^{\mathcal{T}}(x,y)$ that determine $P^{\mathcal{I}_\mathcal{K}}$ (directly, without selecting a subset of the answers). Let $\mathsf{rgen}_{R_n}^{\mathcal{T}}$ be the set of pairs (R_1, R_{n-1}) of roles in $\mathcal{T}$ such that there is a sequence $R_1,\dots,R_n$ satisfying **(rgen)**. Clearly, $\mathsf{rgen}_{R_n}^{\mathcal{T}}$ depends only on $\mathcal{T}$ and can be computed in time polynomial in $|\mathcal{T}|$. For a role name P, set

$$q_P^{\mathcal{T}}(x,y) = P(x,y) \vee \big(\mathsf{gen}_P^{\mathcal{T}}(x) \wedge (y = c_P)\big) \;\vee \big(\mathsf{gen}_{P^-}^{\mathcal{T}}(y) \wedge (x = c_{P^-})\big),$$
$$\mathsf{gen}_R^{\mathcal{T}}(x) = \mathsf{agen}_R^{\mathcal{T}}(x) \;\vee \bigvee_{(R_1,S)\in \mathsf{rgen}_R^{\mathcal{T}}} \big(\exists z\, \mathsf{agen}_{R_1}^{\mathcal{T}}(z) \wedge (x = c_S)\big),$$
$$\mathsf{agen}_R^{\mathcal{T}}(x) = \mathsf{exp}_{\exists R}^{\mathcal{T}}(x) \wedge \neg\exists y\, R(x,y) \wedge \bigwedge\nolimits_{c_S \in \mathsf{N}_\mathsf{I}^{\mathcal{T}}} (x \neq c_S).$$

Lemma 12 *For a DL-Lite$^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ and a role name P, $\mathsf{ans}(q_P^{\mathcal{T}}, \mathcal{I}_\mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$ for all KBs $\mathcal{K} = (\mathcal{T},\mathcal{A})$.*

To define queries $q_A^{\mathcal{T}}(x)$ computing $A^{\mathcal{I}_\mathcal{K}}$, it is enough, by Lemma 11, to restrict $\mathsf{exp}_A^{\mathcal{T}}(x)$ to the domain of $\mathcal{I}_\mathcal{K}$. So as the third step we set $q_A^{\mathcal{T}}(x) = \mathsf{exp}_A^{\mathcal{T}}(x) \wedge D(x)$, where

$$D(x) = \bigwedge\nolimits_{c_R \in \mathsf{N}_\mathsf{I}^{\mathcal{T}}} \Big((x = c_R) \rightarrow \exists z\, \mathsf{gen}_R^{\mathcal{T}}(z)\Big).$$

Lemma 13 *For a DL-Lite$^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}$ and a concept name A, $\mathsf{ans}(q_A^{\mathcal{T}}, \mathcal{I}_\mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$ for all KBs $\mathcal{K} = (\mathcal{T},\mathcal{A})$.*

The constructed queries $q_A^{\mathcal{T}}$ and $q_P^{\mathcal{T}}$ allow us to define $\mathcal{I}_\mathcal{K}$ based on $\mathcal{I}_\mathcal{A}$ using views. They also provide us with the previously announced query $q_\bot^{\mathcal{T}} = \exists x\,(\mathsf{exp}_\bot^{\mathcal{T}}(x) \wedge D(x))$ which can be used to check consistency of $\mathcal{K}$ (see the remark after Theorem 4): a KB $\mathcal{K} = (\mathcal{T},\mathcal{A})$ is consistent if, and only if, $\mathsf{ans}(q_\bot^{\mathcal{T}}, \mathcal{I}_\mathcal{A}) = \emptyset$.

Polynomial Rewriting for *DL-Lite*$^{\mathcal{F}}_{core}$

A very interesting observation is that we can combine the rewritten query $q^\dagger$ with the queries constructing $\mathcal{I}_\mathcal{K}$. The resulting FO query $q^{\mathcal{T},\dagger}$ can be executed directly over $\mathcal{I}_\mathcal{A}$ rather than $\mathcal{I}_\mathcal{K}$. Thus we obtain a novel technique for the pure query rewriting approach. It involves only a *polynomial* blowup for *DL-Lite*$^{\mathcal{F}}_{core}$, called *DL-Lite*$_\mathcal{F}$ in (Calvanese et al. 2006), which is the fragment of *DL-Lite*$^{\mathcal{N}}_{horn}$ with CIs of the form $C_1 \sqsubseteq C_2$, $\geq 2\,R \sqsubseteq \bot$ or $C_1 \sqcap C_2 \sqsubseteq \bot$ and the C_i of the form A or $\exists R$. More precisely, the length of $q^{\mathcal{T},\dagger}$ for a *DL-Lite*$^{\mathcal{F}}_{core}$ TBox $\mathcal{T}$ is

$$\mathcal{O}(|q| \cdot |\mathcal{T}| \cdot \max_{R \text{ a role in } \mathcal{T}} |\mathsf{rgen}_R^{\mathcal{T}}|),$$

which is linear in $|q|$ and at most cubic in $|\mathcal{T}|$. All the previously known query rewriting techniques for *DL-Lite*$_\mathcal{F}$ produce exponential results. Our technique can also be applied to *DL-Lite*$^{\mathcal{N}}_{core}$, which extends *DL-Lite*$^{\mathcal{F}}_{core}$ with arbitrary number restrictions $\geq m\,R$. In this case, however, we have to take account of the number encoding because the subformulas $\mathsf{exp}_{\geq mR}^{\mathcal{T},0}(x)$ are of length $\mathcal{O}(m^2)$. If the numbers are represented in unary then $|q^{\mathcal{T},\dagger}| = \mathcal{O}(|q| \cdot |\mathcal{T}|^5)$. If the numbers are coded in binary and we are allowed to use aggregation functions (e.g., `COUNT` in SQL), then $|q^{\mathcal{T},\dagger}|$ is $\mathcal{O}(|q| \cdot |\mathcal{T}|^4)$. Furthermore, if the subqueries $\mathsf{exp}_C^{\mathcal{T},j}(x)$ are defined as views and thus contribute with size 1 to the length of $\mathsf{exp}_C^{\mathcal{T},j+1}(x)$ (a form of structure sharing), then similar considerations also apply to TBoxes in the full language of *DL-Lite*$^{\mathcal{N}}_{horn}$. Without views or aggregation, the queries used for constructing $\mathcal{I}_\mathcal{K}$ are of exponential size.

Query Answering in *DL-Lite*$^{(\mathcal{HN})}_{horn}$

We now show how our (combined) approach to CQ answering over *DL-Lite*$^{\mathcal{N}}_{horn}$ KBs can be extended to answering *positive existential queries* over *DL-Lite*$^{(\mathcal{HN})}_{horn}$ KBs, which can contain role inclusions, subject to the constraint formulated in the preliminaries. In fact, we show that this more general case can be reduced (at a price of an exponential blowup) to CQ answering over *DL-Lite*$^{\mathcal{N}}_{horn}$ KBs.

Given a *DL-Lite*$^{(\mathcal{HN})}_{horn}$ KB $\mathcal{K} = (\mathcal{T},\mathcal{A})$, we first transform (in polynomial time) the TBox $\mathcal{T}$ into a *DL-Lite*$^{\mathcal{N}}_{horn}$ TBox $\mathcal{T}_h$ by removing all RIs and adding the CIs $\exists R \sqsubseteq \exists S$ whenever $R \sqsubseteq^*_\mathcal{T} S$. Since $\mathcal{K}_h = (\mathcal{T}_h, \mathcal{A})$ is a *DL-Lite*$^{\mathcal{N}}_{horn}$ KB, it has a canonical interpretation $\mathcal{I}_{\mathcal{K}_h}$, which can be stored as a relational instance in the RDBMS. We then show that every positive existential query q can be rewritten into a union of CQs (UCQ) q_h such that the answers to q over $\mathcal{K}$ coincide with the answers to q_h over $\mathcal{I}_{\mathcal{K}_h}$. We rely on Theorem 10 to answer the UCQ q_h in a component-wise fashion.

We construct q_h in two steps. First, we compensate the removal of RIs from $\mathcal{T}$ and transform q into another positive existential query q' by replacing each atom $R(t,t')$ in q with the disjunction $\bigvee_{S \sqsubseteq^*_\mathcal{T} R} S(t,t')$. Clearly, q' can be constructed in polynomial time. Second, we convert q' into disjunctive normal form q_h, i.e., into a UCQ. Of course, this results in an exponential blowup. More precisely, we obtain a union of at most $r^{|q|}$ conjunctive queries, where r is the maximum over $|\{S \mid S \sqsubseteq^*_\mathcal{T} R\}|$, for role atoms $R(t,t')$ in q, which is an improvement over the known $(|\mathcal{T}| \cdot |q|)^{|q|}$ bound for the pure query rewriting approach.

Theorem 14 *For all consistent DL-Lite$^{(\mathcal{HN})}_{horn}$ KBs $\mathcal{K}$ and positive existential queries q, $\mathsf{cert}(q,\mathcal{K}) = \mathsf{ans}(q_h^\dagger, \mathcal{I}_{\mathcal{K}_h})$.*

Proof. By Theorems 4 and 10, it suffices to show that $\mathsf{cert}(q,\mathcal{K}) = \mathsf{cert}(q_h,\mathcal{K}_h)$. Let q (and so q_h) be k-ary.

($\subseteq$) Let $\vec{a}$ be a k-tuple of individual names from $\mathcal{A}$ such that $\mathcal{K}_h \not\models q_h[\vec{a}]$. Then there is a model $\mathcal{I}_h$ of $\mathcal{K}_h$ such that $\mathcal{I}_h \not\models q_h[\vec{a}]$. Construct a new interpretation $\mathcal{I}$ by setting $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}_h}$, $A^{\mathcal{I}} = A^{\mathcal{I}_h}$ for all $A \in \mathsf{N_C}$, and

$$P^{\mathcal{I}} = \{(d, e) \mid (d, e) \in S^{\mathcal{I}_h} \text{ for } S \in \mathsf{N}_{\mathsf{R}}^-,\ S \sqsubseteq^*_{\mathcal{T}} P\}$$

for all $P \in \mathsf{N_R}$. To show that $\mathcal{K} = (\mathcal{T}, \mathcal{A}) \not\models q[\vec{a}]$, it suffices to prove that (i) $\mathcal{I} \models \mathcal{T}$ and (ii) $\mathcal{I} \not\models q[\vec{a}]$.

(i) By definition, $\mathcal{I} \models R \sqsubseteq S$ for all RIs $R \sqsubseteq S$ in $\mathcal{T}$. To prove that we have $\mathcal{I} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ for all CIs $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ in $\mathcal{T}$, assume $d \in (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}}$ and show that $d \in C^{\mathcal{I}}$. This follows if $d \in (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_h}$ because $\mathcal{I}^h \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ and $C^{\mathcal{I}_h} \subseteq C^{\mathcal{I}}$ by the definition of C and $\mathcal{I}$. Assume to the contrary of what has to be shown that $d \notin (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_h}$. Then there exists C_i such that $d \notin C_i^{\mathcal{I}_h}$. Since $d \in C_i^{\mathcal{I}}$, this can only be the case if C_i is of the form $\geq m\, R$. Then there exist e and S such that $(d, e) \in S^{\mathcal{I}_h}$, $S \sqsubseteq^*_{\mathcal{T}} R$ and $S \neq R$. By the definition of *DL-Lite*$_{horn}^{(\mathcal{HN})}$, this means that $m = 1$. But then $d \in (\exists S)^{\mathcal{I}_h}$ and $\exists S \sqsubseteq \exists R \in \mathcal{T}_h$ imply $d \in (\exists R)^{\mathcal{I}_h}$, which is a contradiction as $d \notin C_i^{\mathcal{I}_h}$.

(ii) Assuming $\mathcal{I} \models^\pi q$ for an $\vec{a}$-match π for q in $\mathcal{I}$, we show that $\mathcal{I}_h \models^\pi q'$, which is equivalent to $\mathcal{I}_h \models^\pi q_h$. As q' is constructed from subformulas of the form $A(t)$ and $\bigvee_{S \sqsubseteq^*_{\mathcal{T}} R} S(t, t')$, for $R(t, t') \in q$, we consider two cases. For $A(t)$, we have $\mathcal{I}_h \models^\pi A(t)$ whenever $\mathcal{I} \models^\pi A(t)$, because $A^{\mathcal{I}} = A^{\mathcal{I}_h}$. For $\bigvee_{S \sqsubseteq^*_{\mathcal{T}} R} S(t, t')$ with $R(t, t') \in q$, if $\mathcal{I} \models^\pi R(t, t')$ then, by the construction of $\mathcal{I}$, there exists S with $S \sqsubseteq^*_{\mathcal{T}} R$ and $(\pi(t), \pi(t')) \in S^{\mathcal{I}_h}$, from which $\mathcal{I}_h \models^\pi \bigvee_{S \sqsubseteq^*_{\mathcal{T}} R} S(t, t')$. As q' is built from these subformulas using only conjunction and disjunction, we have $\mathcal{I}_h \models^\pi q'$ whenever $\mathcal{I} \models^\pi q$. Thus, $\mathcal{I}_h \models q_h[\vec{a}]$.

($\supseteq$) If $\mathcal{K}_h = (\mathcal{T}_h, \mathcal{A}) \models q_h[\vec{a}]$ then $(\mathcal{T}_h \cup \mathcal{T}, \mathcal{A}) \models q[\vec{a}]$. But then $\mathcal{T} \models \mathcal{T}_h$ implies $\mathcal{K} = (\mathcal{T}, \mathcal{A}) \models q[\vec{a}]$. ❑

Query Answering in *DL-Lite*$_{horn}^{(\mathcal{HF})}$ without UNA

The Web ontology language OWL does not adopt the unique name assumption (UNA), but allows instead *equality* and *inequality constraints* of the form $a \approx b$ and $a \not\approx b$ for individual names a, b. Without UNA, CQ answering in *DL-Lite*$_{core}^{\mathcal{N}}$ (with or without $\approx$ and $\not\approx$) becomes CONP-hard for data complexity (Artale et al. 2009). If, however, *DL-Lite*$_{horn}^{(\mathcal{HN})}$ TBoxes contain concepts $\geq m\, R$ with $m \geq 2$ only in the form of functionality constraints $\geq 2\, R \sqsubseteq \bot$ (this fragment is called *DL-Lite*$_{horn}^{(\mathcal{HF})}$) then query answering is 'only' PTIME-complete for data complexity. If concepts $\geq m\, R$ with $m \geq 2$ are disallowed altogether, then the complexity further drops to LOGSPACE. In the combined approach, both equality and functionality constraints can be eliminated at the stage of constructing the canonical interpretation. Indeed, given a *DL-Lite*$_{horn}^{\mathcal{F}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, denote by $\mathsf{eq}_{\mathcal{K}}$ the minimal equivalence relation such that

– $(a, b) \in \mathsf{eq}_{\mathcal{K}}$, for each $a \approx b$ in $\mathcal{K}$,
– $\geq 2\, R \sqsubseteq \bot \in \mathcal{T}$, $R(a, b), R(a', b') \in \mathcal{A}$, $(a, a') \in \mathsf{eq}_{\mathcal{K}}$ implies $(b, b') \in \mathsf{eq}_{\mathcal{K}}$.

To construct $\mathcal{I}_{\mathcal{K}}$, we first compute the relation $\mathsf{eq}_{\mathcal{K}}$ and then take it into account in the definition of $A^{\mathcal{I}_{\mathcal{K}}}$ and $P^{\mathcal{I}_{\mathcal{K}}}$ above by stating in the $\mathsf{exp}_C^{\mathcal{I}, j}(x)$ that they can also be obtained from $\mathsf{exp}_C^{\mathcal{I}, j-1}(y) \wedge \mathsf{eq}_{\mathcal{K}}(x, y)$; the queries $q_P^{\mathcal{I}}(x, y)$ are modified accordingly. The RIs in *DL-Lite*$_{horn}^{(\mathcal{HF})}$ can be treated at the query rewriting stage similarly to *DL-Lite*$_{horn}^{(\mathcal{HN})}$ under UNA.

Note also that *DL-Lite*$_{horn}^{(\mathcal{HF})}$ TBoxes (and OWL 2 QL ontologies) may contain role disjointness, (a)symmetry and (ir)reflexivity constraints. The presented approach can be extended to handle these features.

Experiments

We evaluate the performance of the combined rewriting technique by comparing it with the pure query rewriting approach introduced in (Calvanese et al. 2005; 2006; 2007; 2008) and implemented in the QuOnto system (Acciarri et al. 2005; Poggi, Rodriguez, and Ruzzi 2008).

The experiments use several *DL-Lite* ontologies formulated in *DL-Lite*$_{core}$, the common fragment of *DL-Lite*$_{horn}^{\mathcal{N}}$ and the logic underlying QuOnto. Among the ontologies considered are the *DL-Lite*$_{core}$ approximation *Galen-Lite* of the medical ontology *Galen* (consisting of the *DL-Lite*$_{core}$ CIs implied by Galen), the *Core* ontology (a representation of a fragment of a supply-chain management system used by the bookstore chain Waterstone's), the *StockExchange* ontology (an EU financial institution's ontology), and the *University* ontology (a *DL-Lite*$_{core}$ version of the LUBM ontology developed at Lehigh University to describe the university organisational structure). The sample ontologies cover a wide spectrum of *DL-Lite* ontologies, ranging from complex concept hierarchies (as in Galen-Lite) to ontologies with rich role interactions (such as Core). The data was stored and the test queries were executed using DB2-Express version 9.5 running on Intel Core 2 Duo 2.5GHz CPU, 4GB memory and 500GB storage under Linux 2.6.28.

Figure 1 summarises the running times for several test queries and randomly generated ABoxes of various size. For each ABox, we report the number of individuals (Ind, in thousands), the numbers of concept assertions (CAs, in millions) and role assertions (RAs, in millions) in the original ABox and in the canonical interpretation. For each query, we then show the execution times in the columns UN (the unmodified query over the original ABox, which does not give correct answers and serves as an 'ultimate lower bound'), RW (the rewritten query executed over the canonical interpretation), and QO (the query produced by QuOnto executed over the original ABox). The queries reported in the table are sample CQs with 3–6 atoms of various topologies (the exact shapes of the queries can be found at `http://www.dcs.bbk.ac.uk/~roman/query-rewriting/queries.txt`; the queries to StockExchange and University were taken from (Pérez-Urbina, Motik, and Horrocks 2009)); the size of the queries is limited by the feasibility of creating a QuOnto rewriting; the technique proposed here scales to considerably larger conjunctive queries. In the case of the University ontology, *role inclusions* were incorporated into the query rewriting as out-

	Ind	ABox size (in M)				query											
		original		canonical		*Q1*			*Q2*			*Q3*			*Q4*		
	(in K)	CA	RA	CA	RA	UN	RW	QO	UN	RW	QO	UN	RW	QO	UN	RW	QO
Galen-Lite	20	2.0	2.0	9.9	3.7	0.02	0.04	13.69	0.02	0.08	1.65	0.02	0.11	1m 28	0.12	0.22	16m 11
2733 concepts,	50	5.0	5.0	24.8	9.3	0.04	0.55	14.39	0.05	0.19	2.21	0.03	0.28	51.39	0.11	0.43	13m 26
207 roles, and	70	10.0	10.0	43.0	15.4	0.03	0.76	17.56	0.11	0.55	3.01	0.06	0.73	1m 11	0.15	0.63	13m 00
4888 axioms	100	20.0	20.0	75.0	25.8	0.05	0.87	23.86	0.14	0.76	6.55	0.12	0.95	1m 31	0.18	1.52	16m 23
Core	50	2.0	2.0	5.5	2.8	0.22	0.37	17m 41	0.30	0.41	38m 16	0.13	0.29	1m 7	0.19	0.46	6m 26
81 concepts,	100	5.0	5.0	11.8	5.7	0.46	3.97	25m 32	0.53	5.97	97m 47	0.50	1.10	2m 15	0.20	1.00	12m 02
58 roles, and	200	10.0	10.0	23.7	11.4	0.80	5.73	38m 33	0.86	6.65	67m 13	0.81	1.78	3m 28	0.78	2.57	13m 38
381 axioms	300	20.0	20.0	54.5	27.8	1.28	7.32	23m 04	1.34	8.03	71m 49	1.87	3.12	5m 31	1.70	3.86	14m 55
University	100	2.0	2.0	5.5	2.7	0.06	2.61	26m 08	0.10	0.15	49m 36	0.02	0.05	29m 10	0.45	0.67	9m 49
31 concepts,	300	5.0	5.0	13.9	7.2	0.05	5.22	36m 22	0.21	0.13	33m 54	0.02	0.03	29m 29	0.94	1.84	9m 52
25 roles, and	500	10.0	10.0	25.2	13.7	0.06	5.18	22m 17	0.13	0.32	31m 33	0.02	0.02	24m 43	1.48	2.40	10m 02
103 axioms	800	20.0	20.0	46.5	25.3	0.06	0.10	27m 48	0.11	0.30	56m 44	0.02	0.02	27m 42	3.34	3.66	9m 51
StockExchange	200	2.0	2.0	7.0	4.0	1.01	4.08	4m 19	0.89	2.88	17m 56	1.02	2.98	67m 34	1.01	4.06	> 2 h
17 concepts,	500	5.0	5.0	17.6	10.0	2.34	8.64	5m 01	2.08	6.01	12m 11	2.43	6.57	35m 33	1.71	9.26	> 2 h
12 roles, and	1000	10.0	10.0	35.2	20.1	4.47	11.33	6m 19	4.34	13.36	15m 56	4.47	14.37	45m 42	4.45	20.84	> 2 h
62 axioms	1500	20.0	20.0	57.3	35.5	9.31	18.30	7m 12	16.32	22.05	11m 09	9.15	39.37	38m 24	10.31	56.74	> 2 h

Figure 1: Query processing times (in seconds).

lined above, without a significant impact on the overall results.

The results can be summarised as follows: (1) query answering in our approach is competitive in performance with executing the original queries over the data (indeed, the query rewriting simply introduces additional *selection conditions* on top of the original CQ that are executed in a pipelined fashion by the RDBMS); (2) query answering using the QuOnto approach is often prohibitively expensive even for relatively small ontologies; and (3) the construction of canonical interpretations via materialised views can be performed off-line within 2 hours even for the largest data sets (where loading the data into the RDBMS alone takes tens of minutes.) Incremental updates of the ABox can be supported by relying on techniques developed for efficient materialised view maintenance (Colby et al. 1996) albeit not all of these techniques have been implemented in commercial database systems such as DB2 as of today.

Conclusion

We presented a combined approach to CQ answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ and some of its variants and demonstrated that this approach often allows more efficient query execution than pure query rewriting. There are several open issues for future work. In particular, we do not know whether the combined approach can be implemented for *DL-Lite*$^{(\mathcal{HN})}_{horn}$ without an exponential blowup in the rewritten queries. A closely related open problem is whether the combined approach for *DL-Lite*$^{\mathcal{N}}_{horn}$ can be extended to positive existential queries without such a blowup. Finally, our polynomial rewriting for *DL-Lite*$^{\mathcal{F}}_{core}$ in the pure query rewriting approach raises the question whether the exponential blowup can also be avoided in other variants of *DL-Lite*.

Appendix: Proof of Theorem 4

Theorem 4 *For every satisfiable DL-Lite*$^{\mathcal{N}}_{horn}$ *KB* $\mathcal{K}$ *and every CQ* q, $\mathsf{cert}(q,\mathcal{K}) = \mathsf{ans}(q,\mathcal{U}_\mathcal{K})$.

Proof. Suppose $\mathcal{K} = (\mathcal{T},\mathcal{A})$. As shown in (Artale et al. 2009), $\mathcal{K} \models q[\vec{a}]$ if, and only if, $\mathcal{J}_\mathcal{K} \models q[\vec{a}]$, where $\mathcal{J}_\mathcal{K}$ is the (canonical or minimal) model of $\mathcal{K}$ constructed inductively as follows.

Step 0. Set $W_0 = \mathsf{Ind}(\mathcal{A})$ and, for all concept and role names A and P, set $A_0 = \{a \in W_0 \mid \mathcal{K} \models A(a)\}$ and $P_0 = \{(a,b) \mid P(a,b) \in \mathcal{A}\}$; P_0^- is the inverse of P_0. In parallel with the construction of $\mathcal{J}_\mathcal{K}$ we also define a map $h\colon \Delta^{\mathcal{J}_\mathcal{K}} \to \Delta^{\mathcal{U}_\mathcal{K}}$. At step 0, we set $h_0(a) = a$ for $a \in W_0$.

The domain $\Delta^{\mathcal{J}_\mathcal{K}}$ of $\mathcal{J}_\mathcal{K}$ will consist of $\mathsf{Ind}(\mathcal{A})$ and multiple copies of certain 'virtual' points y_R for some roles R, which are supposed to serve as witnesses for incoming R-arrows. If w is a copy of y_R then we write $\mathit{cp}(w) = y_R$.

Step $n+1$. For a role R and a point $w \in W_n$, let $r_n(R,w)$ be the number of distinct R-successors of w in W_n, that is, $r_n(R,w) = \big|\{u \in W_n \mid (w,u) \in R_n\}\big|$. Let $r(R,a)$, for $a \in \mathsf{Ind}(\mathcal{A})$, be the maximum m for which $\mathcal{K} \models\, \geq m\, R(a)$ and, for $\mathit{cp}(w) = y_S$, let $r(R,w)$ be the maximum number m for which $\mathcal{K} \models \exists S^- \sqsubseteq\, \geq m\, R$. If such an m does not exist then we set $r(R,a) = 0$ or, respectively, $r(R,w) = 0$.

For each $w \in W_n$ with $r(R,w) - r_n(R,w) = l > 0$, we add l new points $u_1,\dots,u_l$ to W_n, set $\mathit{cp}(u_i) = y_R$, add the u_i to A_n if $\mathcal{K} \models \exists R^- \sqsubseteq A$, and add the pairs (w,u_i) to R_n. This defines W_{n+1}, A_{n+1} and P_{n+1}, for all concept and role names A and P. Let us now define h_{n+1}. Suppose that $h_n(w) = a \in \mathsf{Ind}(\mathcal{A})$. If $a \rightsquigarrow c_R$ then $a \cdot c_R \in \Delta^{\mathcal{U}_\mathcal{K}}$, $(a, a\cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$, and we set $h_{n+1}(u_i) = a \cdot c_R$, for $i \leq l$. If $a \not\rightsquigarrow c_R$ then, by **(agen)**, there is $b \in \mathsf{Ind}(\mathcal{A})$ such that $R(a,b) \in \mathcal{A}$, i.e., $(a,b) \in R^{\mathcal{U}_\mathcal{K}}$. Set $h_{n+1}(u_i) = b$. Assume now that $h_n(w) = \sigma \cdot c_S$ for some S. By IH, we have $\mathit{cp}(w) = y_S$. If $c_S \rightsquigarrow c_R$ then $\sigma \cdot c_S \cdot c_R \in \Delta^{\mathcal{U}_\mathcal{K}}$ and $(\sigma \cdot c_S, \sigma \cdot c_S \cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$. We set $h_{n+1}(u_i) = \sigma \cdot c_S \cdot c_R$, for $i \leq l$. Otherwise, by **(rgen)**, we must have $S^- = R$ and $(\sigma \cdot c_S, \sigma) \in R^{\mathcal{U}_\mathcal{K}}$. We then set $h_{n+1}(u_i) = \sigma$, for $i \leq l$.

Step ω. Finally, set $\Delta^{\mathcal{J}_\mathcal{K}} = \bigcup_{i<\omega} W_i$, $A^{\mathcal{J}_\mathcal{K}} = \bigcup_{i<\omega} A_i$ and $P^{\mathcal{J}_\mathcal{K}} = \bigcup_{i<\omega} P_i$, for all role and concept names A and P in $\mathcal{K}$, and $a^{\mathcal{J}_\mathcal{K}} = a$ for all individual names a. (Note that $\mathcal{J}_\mathcal{K} \models \mathcal{K}$.) And let $h = \bigcup_{i<\omega} h_i$.

It follows immediately from the definition that $\mathcal{U}_\mathcal{K}$ is a substructure of $\mathcal{J}_\mathcal{K}$. On the other hand, the map h is clearly a homomorphism from $\mathcal{J}_\mathcal{K}$ onto $\mathcal{U}_\mathcal{K}$. Therefore, $\mathcal{J}_\mathcal{K} \models q[\vec{a}]$ if, and only if, $\mathcal{U}_\mathcal{K} \models q[\vec{a}]$. ❑

Appendix: Proof of Theorem 10

To prove Theorem 10, we first give a more 'imperative' definition of the tree witnesses:

Lemma 15 *Given a CQ q and $R(t,t') \in q$, define a relation $X_{R,t} = \bigcup_{i\geq 0} X^i_{R,t} \subseteq \mathsf{term}(q) \times (\mathsf{N}^-_\mathsf{R})^*$, where*

$$
\begin{aligned}
X^0_{R,t} &= \{(t,\varepsilon)\},\\
X^{i+1}_{R,t} &= X^i_{R,t} \cup \{(s',R) \mid (s,\varepsilon) \in X^i_{R,t}, R(s,s') \in q\} \cup\\
&\quad \{(s', w\cdot S\cdot S') \mid (s, w\cdot S) \in X^i_{R,t},\\
&\qquad\qquad S'(s,s') \in q \text{ and } S' \neq S^-\} \cup\\
&\quad \{(s',w) \mid (s,w\cdot S) \in X^i_{R,t},\ \ S^-(s,s') \in q\},
\end{aligned}
$$

for $i \geq 0$. A tree witness $f_{R,t}$ exists and $f_{R,t} = X_{R,t}$ if, and only if, $X_{R,t}$ is a partial function.

Proof. It should be clear from the definitions that if a tree witness exists then the $X^i_{R,t}$ construct it in a step-by-step fashion. On the other hand, $X_{R,t}$ always exists and if it is not a partial function then there can be no $f_{R,t}$ satisfying the definition of the tree witnesses. ❑

As an immediate consequence, we obtain the following *composition property*:

Lemma 16 *If $f_{R,t}$ and $f_{S,s}$ exist, $f_{R,t}(s) = w\cdot Q$, $Q \neq S^-$ and $f_{S,s}(s')$ is defined, then $f_{R,t}(s') = f_{R,t}(s)\cdot f_{S,s}(s')$.*

Proof. We have $(s', w\cdot w) \in X^{i+j}_{R,t}$ if $(s,w) \in X^i_{R,t}$ and $(s',w') \in X^j_{S,s}$. ❑

Theorem 10 *For every DL-Lite$^\mathcal{N}_{horn}$ KB $\mathcal{K}$ and every CQ q, we have $\mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K}) = \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.*

Proof. ($\supseteq$) Let τ be an $\vec{a}$-match for $\mathcal{U}_\mathcal{K}$ and the CQ q. Define a map $\pi\colon \mathsf{term}(q) \to \Delta^{\mathcal{I}_\mathcal{K}}$ by taking $\pi(t) = \mathsf{tail}(\tau(t))$, for all $t \in \mathsf{term}(q)$. By the definitions of π and $\mathcal{U}_\mathcal{K}$, we have $\mathcal{I}_\mathcal{K} \models^\pi \varphi$, and so π is an $\vec{a}$-match for $\mathcal{I}_\mathcal{K}$ and q. Thus, it remains to show that $\mathcal{I}_\mathcal{K} \models^\pi \varphi_1 \wedge \varphi_2 \wedge \varphi_3$. To do this, we require the following lemma:

Lemma 17 *Suppose that $R(t,t') \in q$ and $\pi(t') = c_R$. Then $(s, S_1\cdots S_k) \in X_{R,t}$ implies $\tau(s) = \tau(t)\cdot c_{S_1}\cdots c_{S_k}$. It follows that $X_{R,t}$ is a partial function and $f_{R,t} = X_{R,t}$.*

Proof. By Lemma 15, it suffices to show that, for every $i \geq 0$, if $(s, S_1\cdots S_k) \in X^i_{R,t}$ then $\tau(s) = \tau(t)\cdot c_{S_1}\cdots c_{S_k}$. We proceed by induction on i. For the basis of induction, we have $X^0_{R,t} = \{(t,\varepsilon)\}$, and so there is nothing to show. For the induction step, we consider three cases in accordance with the definition of $X^{i+1}_{R,t}$.

(i) Let $(s',R) \in X^{i+1}_{R,t}$, $(s,\varepsilon) \in X^i_{R,t}$ and $R(s,s') \in q$. By IH, $\tau(s) = \tau(t)$. First assume $\tau(s) \notin \mathsf{Ind}(\mathcal{A})$. Then by **(rgen)**, $\pi(t') = c_R$ and $R(t,t') \in q$ entail $\mathsf{tail}(\tau(s)) = \mathsf{tail}(\tau(t)) \neq c_{R^-}$. Once more by **(rgen)**, $\mathcal{U}_\mathcal{K} \models^\tau R(s,s')$ implies $\tau(s') = \tau(s)\cdot c_R = \tau(t)\cdot c_R$ as required. Now assume $\tau(s) \notin \mathsf{Ind}(\mathcal{A})$. Then **(agen)**, $\pi(t') = c_R$ and $R(t,t') \in q$ yield that $R(\tau(t),b) = R(\tau(s),b) \notin \mathcal{A}$ for any $b \in \mathsf{Ind}(\mathcal{A})$. Thus $\mathcal{U}_\mathcal{K} \models^\tau R(s,s')$ implies $\tau(s') = \tau(t)\cdot c_R$.

(ii) Let $(s', w\cdot S\cdot S') \in X^{i+1}_{R,t}$, $(s, w\cdot S) \in X^i_{R,t}$, for some $w = S_1\cdots S_k$, $S' \neq S^-$ and $S'(s,s') \in q$. Then, by IH, $\tau(s) = \tau(t)\cdot c_{S_1}\cdots c_{S_k}\cdot c_S$. So, since $S' \neq S^-$ and $\mathcal{U}_\mathcal{K} \models^\tau S'(s,s')$, **(rgen)** gives us $\tau(s') = \tau(s)\cdot c_{S'}$, as required.

(iii) Suppose $(s',w) \in X^{i+1}_{R,t}$, $(s, w\cdot S) \in X^i_{R,t}$, for some $w = S_1\cdots S_k$, and $S^-(s,s') \in q$. By IH, we have $\tau(s) = \tau(t)\cdot c_{S_1}\cdots c_{S_k}\cdot c_S$. In view of **(rgen)** and $\mathcal{U}_\mathcal{K} \models^\tau S(s',s)$, we then obtain $\tau(s') = \tau(t)\cdot c_{S_1}\cdots c_{S_k}$, as required. ❑

We can now show that $\mathcal{I}_\mathcal{K} \models^\pi \varphi_1 \wedge \varphi_2 \wedge \varphi_3$.

φ_1**:** By the definition of matches, we have $\tau(v) \in \mathsf{Ind}(\mathcal{A})$ for any $v \in \mathsf{avar}(q)$ and $c_R \in \mathsf{N}^\mathcal{T}_\mathsf{I}$. And by the definition of π, $\pi(v) \neq c_R$ for any such v. Thus, $\mathcal{I}_\mathcal{K} \models^\pi \varphi_1$.

φ_2**:** Let $R(t,t') \in q$ and $\pi(t') = c_R$. To prove $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, it is enough to show that $f_{R,t}$ exists, which follows from Lemma 17. Thus, $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$.

φ_3**:** Take an equivalence class $[t]^R_q$ such that $f_{R,t}$ exists. Assume there is $R(s,s') \in q$ with $s \equiv^R_q t$ and $\pi(s') = c_R$, i.e., the premise in φ_3 is true. Then $f_{R,s}(t) = \varepsilon$ and Lemma 17 yields $\tau(s) = \tau(t)$. Take an $R(t,t') \in q$. Since $\tau(s) = \tau(t)$, $R(s,s') \in q$, and $\mathsf{tail}(\tau(s')) = c_R$, **(agen)** and **(rgen)** yield $\tau(s') = \tau(t')$. Thus $\pi(t') = c_R$. Since $f_{R,t}(s'') = \varepsilon$, another application of Lemma 17 yields $\tau(s'') = \tau(t)$ as required.

This shows that $\mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K}) \supseteq \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.

($\subseteq$) Assume now that π is an $\vec{a}$-match for $\mathcal{I}_\mathcal{K}$ and $q^\dagger$. Our aim is to show that there is an $\vec{a}$-match τ for $\mathcal{U}_\mathcal{K}$ and $q^\dagger$. Obviously, we can set $\tau(t) = \pi(t)$ whenever $\pi(t) \in \mathsf{Ind}(\mathcal{A})$. Defining $\tau(t)$ for other $t \in \mathsf{term}(q)$—that is, for the terms t that are mapped by π to points of the form c_R in $\mathcal{I}_\mathcal{K}$—is a bit more involved.

Lemma 18 *Suppose that $R(t,t') \in q$ and $\pi(t') = c_R$. If $f_{R,t}(s)$ is defined then*

$$
\pi(s) = \begin{cases} c_S, & \text{if } f_{R,t}(s) = w\cdot S,\\ \pi(t), & \text{if } f_{R,t}(s) = \varepsilon. \end{cases}
$$

Proof. The proof is by induction on i in the definition of $X_{R,t}$. The basis of induction is proved in the same way as in the proof of Lemma 17. For the induction step, we consider three cases.

(i) If $(s',R) \in X^{i+1}_{R,t}$, $(s,\varepsilon) \in X^i_{R,t}$ and $R(s,s') \in q$ then, by IH, $\pi(s) = \pi(t)$. Clearly, $\mathcal{I}_\mathcal{K} \models^\pi R(s,s')$ imply either (i.1) $\pi(s) \notin \mathsf{Ind}(\mathcal{A})$ and then, in view of $\pi(t') = c_R$ and **(rgen)**, $\pi(s) \neq c_{R^-}$, or (i.2) $\pi(s) \in \mathsf{Ind}(\mathcal{A})$ and then, by **(agen)**, $R(\pi(s),b) \notin \mathcal{A}$ for all $b \in \mathsf{Ind}(\mathcal{A})$. In either case, $\mathcal{I}_\mathcal{K} \models^\pi S(s,s')$ implies $\pi(s') = c_R$ as required.

(ii) If $(s', w\cdot S\cdot S') \in X^{i+1}_{R,t}$, $(s, w\cdot S) \in X^i_{R,t}$, $S'(s,s')$ in q, $S' \neq S^-$ then, by IH, $\pi(s) = c_S$. As $\mathcal{I}_\mathcal{K} \models^\pi S'(s,s')$ and $S' \neq S^-$, by **(rgen)**, $\pi(s') = c_{S'}$.

(iii) Suppose that $(s', w) \in X^{i+1}_{R,t}$, $(s, w \cdot S) \in X^i_{R,t}$ and $S^-(s, s') \in q$. If $w = \varepsilon$ then $S = R$ and, by $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, $\pi(s') = \pi(t)$. Otherwise, $w = w' \cdot Q$ and, by the definition of $X^i_{R,t}$, there is $(s'', w' \cdot Q) \in X^i_{R,t}$ with $(s', \varepsilon) \in X_{S,s''}$. By IH, $\pi(s'') = c_Q$ and, since $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, $X_{S,s''}$ is a partial function and $f_{S,s''}(s') = \varepsilon$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, we obtain $\pi(s'') = \pi(s') = c_Q$. ❑

Call a term $t \in \mathsf{term}(q)$ a *root* (*of* q *under* π) if

– either $\pi(t) \in \mathsf{Ind}(\mathcal{A})$

– or $\pi(t) = c_R$ and there is no atom $R(t', t) \in q$.

A root t is called *initial* if there is no $S(s, s') \in q$ such that s is a root, $\pi(s') = c_S$, $f_{S,s}(t)$ is defined and $f_{S,s}(t) \neq \varepsilon$.

Example 19 Consider the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with

$$\begin{aligned}\mathcal{T} &= \{A_1 \sqsubseteq A,\ A_2 \sqsubseteq A,\ \exists P^- \sqsubseteq \exists S \sqcap \exists R,\\ &\qquad\qquad \exists R^- \sqsubseteq \exists S,\ A \sqsubseteq \exists P\},\\ \mathcal{A} &= \{A_1(a),\ A_2(b)\}.\end{aligned}$$

Then the canonical interpretation $\mathcal{I}_\mathcal{K}$ is as follows:

$$\begin{aligned}&\Delta^{\mathcal{I}_\mathcal{K}} = \{a, b, c_P, c_S, c_R\},\\ &A^{\mathcal{I}_\mathcal{K}} = \{a, b\},\ \ A_1^{\mathcal{I}_\mathcal{K}} = \{a\},\ \ A_2^{\mathcal{I}_\mathcal{K}} = \{b\},\\ &P^{\mathcal{I}_\mathcal{K}} = \{(a, c_P), (b, c_P)\},\ \ R^{\mathcal{I}_\mathcal{K}} = \{(c_P, c_R)\},\\ &S^{\mathcal{I}_\mathcal{K}} = \{(c_P, c_S), (c_R, c_S)\}.\end{aligned}$$

Let $q = \exists t_1 t_2 t_3 t_4\, (R(t_1, t_2) \land S(t_2, t_3) \land S(t_4, t_3))$. Then

$$\begin{gathered} f_{R,t_1}(t_2) = R, \qquad f_{R,t_1}(t_1) = \varepsilon,\\ f_{R,t_1}(t_3) = R \cdot S, \qquad f_{R,t_1}(t_4) = R,\\ f_{S,t_4}(t_3) = S, \qquad f_{S,t_4}(t_4) = \varepsilon,\\ f_{S,t_4}(t_2) = \varepsilon \quad \text{and} \quad f_{S,t_4}(t_1) \text{ is not defined.}\end{gathered}$$

It is readily checked that the map π defined by taking

$$\pi(t_1) = c_P, \quad \pi(t_2) = \pi(t_4) = c_R, \quad \pi(t_3) = c_S$$

is an $\vec{a}$-match for $\mathcal{I}_\mathcal{K}$ and $q^\dagger$. Both t_1 and t_4 are roots, while t_2 and t_3 are not roots. Moreover, $f_{R,t_1}(t_4) = R$, while $f_{R,t_4}(t_1)$ is not defined.

Root t_1 is initial (although t_4 is a root with $S(t_4, t_3) \in q$, $\pi(c_3) = c_S$ and $f_{S,t_4}(t_1)$ not defined). On the contrary, root t_4 is not initial because $f_{R,t_1}(t_4) = R$.

Lemma 20 *If* $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ *then* t *is an initial root.*

Proof. By definition, if $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ then t is a root. To show that t is initial, assume to the contrary that there is some $S(s, s') \in q$ such that s is a root, $\pi(s') = c_S$, $f_{S,s}(t)$ is defined and $f_{S,s}(t) \neq \varepsilon$. By Lemma 18, $\pi(t) \in \mathsf{N}_\mathsf{I}^\mathcal{I}$, which is a contradiction. ❑

Lemma 21 *For each* $t \in \mathsf{term}(q)$*, either* t *is an initial root or there is an initial root* $s \in \mathsf{term}(q)$ *such that* $f_{R,s}(t)$ *is defined and nonempty.*

Proof. Let $t \in \mathsf{term}(q)$. We first find a root r and then an initial root s that are 'connected' to t.

If t is a root, we set $r = t$. Otherwise, there is some $R_0(t_1, t_0) \in q$ with $t_0 = t$ and $\pi(t_0) = c_{R_0}$. Further, either t_1 is a root or there is some $R_1(t_2, t_1) \in q$ with $\pi(t_1) = c_{R_1}$. We iterate this argument until we reach a root. To show that this indeed eventually happens, suppose otherwise. Then there is an infinite sequence $R_0(t_1, t_0), R_1(t_2, t_1), \dots$ of atoms in q such that $t = t_0$ and $\pi(t_i) = c_{R_i}$ for all $i \geq 0$. By **(rgen)**, we have $R_i \neq R^-_{i+1}$ for all $i \geq 0$. Since $\mathsf{term}(q)$ is finite, there are j, k with $j < k$ and $t_j = t_k$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, the tree witness $f_{R_k, t_{k+1}}$ exists. Since $R_i \neq R^-_{i+1}$ for all $i \geq 0$, it is easy to see that $f_{R_k, t_{k+1}}(t_i) = R_k R_{k-1} \cdots R_i$ for all $i \leq k$. We thus obtain $f_{R_k, t_{k+1}}(t_j) = R_k \cdots R_j$, contrary to $f_{R_k, t_{k+1}}(t_k) = R_k$ and $t_k = t_j$. So there is a sequence $R_0(t_1, t_0), \dots, R_{\ell-1}(t_\ell, t_{\ell-1}) \in q$ such that $t_0 = t$, $r = t_\ell$ is a root, $R_i \neq R^-_{i+1}$, for $i < \ell - 1$, and $\pi(t_i) = c_{R_i}$, for $i \leq \ell - 1$. By the definition of tree witnesses, we then have $f_{R', r}(t) = R_{\ell-1} \cdots R_0 \neq \varepsilon$, where $R' = R_{\ell-1}$.

If r is an initial root, we set $s = r$. Otherwise, there is some $R_0(s_0, s'_0) \in q$ such that s_0 is a root, $\pi(s'_0) = c_{R_0}$, $f_{R_0, s_0}(r)$ is defined and $f_{R_0, s_0}(r) \neq \varepsilon$. If s_0 is initial, we set $s = s_0$. Otherwise, there is some $R_1(s_1, s'_1) \in q$ such that s_1 is a root, $\pi(s'_1) = c_{R_1}$, $f_{R_1, s_1}(s_0)$ is defined and $f_{R_1, s_1}(s_0) \neq \varepsilon$. Let $f_{R_1, s_1}(s_0) = w \cdot R$. By Lemma 18, $\pi(s_0) = c_R$ and, in view of $\mathcal{I}_\mathcal{K} \models^\pi R_0(s_0, s'_0)$, $\pi(s'_0) = c_{R_0}$ and **(rgen)**, we have $R \neq R_0^-$. By Lemma 16, $f_{R_1, s_1}(r) = f_{R_1, s_1,}(s_0) \cdot f_{R_0, s_0}(r)$. We can repeat this argument, and each time the word $f_{R_i, s_i}(r)$ becomes strictly longer. Due to finiteness of q, we thus eventually reach an initial root s such that $R(s, s') \in q$, $\pi(s') = c_R$ and $f_{R,s}(r)$ is defined and non-empty. To complete the proof, we notice that $f_{R,s}(t)$ is also defined and $f_{R,s}(t) = f_{R,s}(r) \cdot f_{R', r}(t) \neq \varepsilon$. ❑

Lemma 22 *Suppose* $R(r, r'), S(s, s') \in q$ *are such that both* r *and* s *are initial roots,* $\pi(r') = c_R$*,* $\pi(s') = c_S$ *and* $f_{R,r}(t)$*,* $f_{S,s}(t)$ *are defined. Then* $f_{R,r}(t) = f_{S,s}(t)$ *and* $\pi(r) = \pi(s)$*.*

Proof. Assume for definiteness that $|f_{R,r}(t)| \leq |f_{S,s}(t)|$. Without loss of generality we assume $(t, w_0 \cdot w_1) \in X_{S,s}$ and $(t, w_2 \cdot w_1) \in X_{R,r}$, where w_2 is either empty or its last symbol is distinct from the last symbol of w_0. Then we have $(r, w_0 \cdot w_2^-) \in X_{S,s}$, where w_2^- is the inverse of w_2. Therefore, $f_{S,s}(r) = w_0 \cdot w_2^-$. As r is an initial root, $f_{S,s}(r) = \varepsilon$, and so both w_0 and w_2 are empty and $f_{S,s}(t) = f_{R,r}(t)$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, we obtain $\pi(r) = \pi(s)$. ❑

We are now in a position to define an $\vec{a}$-match τ for $\mathcal{U}_\mathcal{K}$ and q. For each $c_R \in \Delta^{\mathcal{I}_\mathcal{K}}$, we choose a $\gamma(c_R) \in \Delta^{\mathcal{U}_\mathcal{K}}$ with $\mathsf{tail}(\gamma(c_R)) = c_R$ and define a map $\tau\colon \mathsf{term}(q) \to \Delta^{\mathcal{U}_\mathcal{K}}$ as follows:

(a) if $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ then we set $\tau(t) = \pi(t)$;

(b) if $\pi(t) \notin \mathsf{Ind}(\mathcal{A})$ and t is an initial root, then we set $\tau(t) = \gamma(\pi(t))$;

(c) if $f_{R,s}(t)$ is defined for an initial root $s \in \mathsf{term}(q)$ and $f_{R,s}(t) = S_1 \cdots S_k \neq \varepsilon$, then set $\tau(t) = \tau(s) \cdot c_{S_1} \cdots c_{S_k}$.

By Lemma 21, τ is total. To see that it is well-defined, it suffices to observe that, by Lemma 20, cases **(a)**–**(c)** are disjoint (i.e., for each $t \in \mathsf{term}(q)$, $\tau(t)$ is defined in only one of them) and that **(c)** is well-defined by Lemma 22. Thus, it remains to show that τ is an $\vec{a}$-match for $\mathcal{U}_\mathcal{K}$ and q.

Note first that, by the definition of τ and Lemma 18, we have

$$\mathsf{tail}(\tau(t)) = \pi(t), \quad \text{for all } t \in \mathsf{term}(q). \qquad (1)$$

By the definition of $\mathcal{U}_\mathcal{K}$ and (1), all concept atoms in q are satisfied by τ. Let $R(t, t') \in q$ such that $\mathcal{I}_\mathcal{K} \models^\pi R(t, t')$. It remains to show that $\mathcal{U}_\mathcal{K} \models^\tau R(t, t')$. The following six cases are possible:

1. $\pi(t)$ and $\pi(t')$ are defined in **(a)**. Then $(\tau(t), \tau(t'))$ equals $(\pi(t), \pi(t')) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A})$. Thus, $\mathcal{U}_\mathcal{K} \models^\tau R(t, t')$.
2. $\pi(t)$ is defined in **(a)** and $\pi(t')$ in **(b)**. Then $\pi(t)$ is in $\mathsf{Ind}(\mathcal{A})$. By the definition of $\mathcal{I}_\mathcal{K}$, $\pi(t') = c_R$ contrary to t' being a root as $R(t, t') \in q$. So this case is impossible.
3. $\pi(t)$ is defined in **(a)** and $\pi(t')$ in **(c)**. Then $\tau(t) = \pi(t) \in \mathsf{Ind}(\mathcal{A})$ and, by Lemma 20, t is an initial root. By the definition of $\mathcal{I}_\mathcal{K}$, $\pi(t') = c_R$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, $f_{R,t}(t')$ is defined and $f_{R,t}(t') = R$. By Lemma 22, **(c)** and **(a)**, we thus have $\tau(t') = \pi(t) \cdot c_R$. Clearly, $\mathcal{U}_\mathcal{K} \models^\tau R(t, t')$.
4. $\pi(t)$ and $\pi(t')$ are defined in **(b)**. Then $\pi(t) = c_S$ for some S such that there is no $S(s, t) \in q$, and so $S \neq R^-$. By the definition of $\mathcal{I}_\mathcal{K}$, $\pi(t') = c_R$, contrary to $\pi(t')$ being a root as $R(t, t') \in q$, so this case is impossible.
5. $\pi(t)$ is defined in **(b)** and $\pi(t')$ in **(c)**. Then $\pi(t) = c_S$ for some S with no $S(s, t) \in q$, and so $S \neq R^-$. By the definition of $\mathcal{I}_\mathcal{K}$, $\pi(t') = c_R$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, $f_{R,t}(t')$ is defined and, clearly, $f_{R,t}(t') = R$. By Lemma 22, **(c)** and **(b)**, we thus have $\tau(t) = \gamma(\pi(t))$ and $\tau(t') = \gamma(\pi(t)) \cdot c_R$. Clearly, $\mathcal{U}_\mathcal{K} \models^\tau R(t, t')$.
6. $\pi(t)$ and $\pi(t')$ are defined in **(c)**. Then there is an initial root s with $f_{S,s}(t) = S_1 \cdots S_k$ and $\tau(t) = \tau(s) \cdot c_{S_1} \cdots c_{S_k}$. By Lemma 22, the definition of $\tau(t')$ does not depend on the choice of a particular initial root, so we assume it is s. If $R \neq S_k^-$ then $f_{S,s}(t') = S_1 \cdots S_k \cdot R$ and thus $\tau(t') = \tau(s) \cdot c_{S_1} \cdots c_{S_k} \cdot c_R$. Otherwise, i.e., if $R = S_k^-$ then $f_{S,s}(t') = S_1 \cdots S_{k-1}$ and thus $\tau(t') = \tau(s) \cdot c_{S_1} \cdots c_{S_{k-1}}$. In either case, by (1), $\pi(t) = \mathsf{tail}(\tau(t))$ and $\pi(t') = \mathsf{tail}(\tau(t'))$. Also, in both cases $\mathcal{I}_\mathcal{K} \models^\pi R(t, t')$, and by the definition of $\mathcal{U}_\mathcal{K}$, $\mathcal{U}_\mathcal{K} \models^\tau R(t, t')$.

Thus, $\mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K}) \subseteq \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$, which completes the proof of Theorem 10. ❑

References

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

Acciarri, A.; Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Palmieri, M.; Rosati, R. 2005. QuOnto: Querying ontologies. In *Proc. of AAAI*, 1670–1671.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research* 36:1–69.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

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

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

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning* 39(3):385–429.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2008. Inconsistency tolerance in P2P data integration: an epistemic logic approach. *Information Systems* 33(4):360–384.

Colby, L. S.; Griffin, T.; Libkin, L.; Mumick, I. S.; and Trickey, H. 1996. Algorithms for deferred view maintenance. In *ACM SIGMOD: Management of Data*, 469–480.

Dolby, J.; Fokoue, A.; Kalyanpur, A.; Ma, L.; Schonberg, E.; Srinivas, K.; and Sun, X. 2008. Scalable grounded conjunctive query evaluation over large and expressive knowledge bases. In *Proc. of ISWC*, 403–418.

Heymans, S. *et al.* 2008. Ontology reasoning with large data repositories. In *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications*, Springer. 89–128.

Kazakov, Y. 2009. Consequence-driven reasoning for Horn-$\mathcal{SHIQ}$ ontologies. In *Proc. of IJCAI*, 2040–2045.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2009. Combined FO rewritability for conjunctive query answering in *DL-Lite*. In *Proc. of DL*, vol. 477 of *CEUR-WS*.

Kowalski, R., and Kuehner, D. 1971. Linear resolution with selection function. *Artificial Intelligence* 2:227–260.

Lutz, C.; Toman, D.; and Wolter, F. 2009. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. of IJCAI*, 2070–2075.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2008. Efficient Query Answering for OWL 2. In *Proc. of ISWC*, 489–504.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2009. A comparison of query rewriting techniques for *DL-Lite*. In *Proc. of DL*, vol. 477 of *CEUR-WS*.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J. on Data Semantics* 10:133–173.

Poggi, A.; Rodriguez, M.; and Ruzzi, M. 2008. Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protégé. In *Proc. of OWLED 2008*, vol. 496 of *CEUR-WS*.

Stocker, M.; and Smith, M. 2008. Owlgres: A scalable OWL reasoner. In *Proc. of OWLED 2008*, vol. 496 of *CEUR-WS*.