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

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 languages supporting the 'classical' reasoning tasks that are used for ontology design such as satisfiability and subsumption. 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, are emerging. 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 series of papers (Calvanese et al. 2005; 2006; 2008) with the ultimate aim 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. This *rewriting approach* to ontology-based data access has been implemented in various systems such as QuOnto, Owlgres and REQUIEM. Unfortunately, experiments have revealed that these systems do not provide sufficient scalability even for medium-size ontologies (with a few hundred of 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 for which conjunctive query answering is in AC^0 for data complexity, has been proposed (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 an RDBMS 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 combined approach to conjunctive query answering for *DL-Lite* ontologies. In particular, we present *polynomial* rewriting techniques for both data and query in the case when ontologies are formulated in *DL-Lite*$_{horn}^{\mathcal{N}}$ (properly containing *DL-Lite*$_{\mathcal{F},\sqcap}$ of (Calvanese et al. 2006)). For ontologies in *DL-Lite*$_{horn}^{(\mathcal{HN})}$, extending *DL-Lite*$_{horn}^{\mathcal{N}}$ with role inclusions, we could not avoid an exponential blowup (but only in the number of roles) of the rewritten queries, while keeping the expanded data polynomial. 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 is not 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 $\mathcal{D}'$ whose size is $\mathcal{O}(|\mathcal{D}| \cdot |\mathcal{T}| + |\mathcal{T}|^2)$ in the worst case. When 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 $\mathcal{O}(|q|^2)$ for *DL-Lite*$_{horn}^{\mathcal{N}}$ ontologies. The rewriting $q^{\dagger}$ is drastically different from the rewriting of (Kontchakov et al. 2009), which involves an exponential blowup in the worst case. For a *DL-Lite*$_{horn}^{(\mathcal{HN})}$ 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 for role atoms in q. We also show that the expansion of data can be implemented using views in the RDBMS, which has the nice effect that the expanded database is automatically and transparently adjusted when the underlying data is updated. As a by-product of the view construction, we obtain a novel way of pure query rewriting for *DL-Lite*$_{horn}^{\mathcal{N}}$. When applied to *DL-Lite*$_{\mathcal{F}}$ of (Calvanese et al. 2006), which disallows conjunction and all number restrictions but existential quantifiers and functionality constraints, this new technique blows up the rewritten query only polynomially. As no views are involved in this case, we obtain a first polynomial query rewriting technique for a *DL-Lite* logic. A full version of the paper is available at `http://www.dcs.bbk.ac.uk/~roman/`.

Preliminaries

We briefly introduce *DL-Lite*$_{horn}^{(\mathcal{HN})}$, the most expressive dialect of *DL-Lite* for which conjunctive 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*$_{horn}^{\mathcal{N}}$. 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 N_R^- to denote the set of all roles. In DL, ontologies are represented as TBoxes. A *DL-Lite*$_{horn}^{(\mathcal{HN})}$ *TBox* is a finite set $\mathcal{T}$ of *concept* and *role inclusions* (*CIs* and *RIs*), which take the form $C_1 \sqcap \dots \sqcap C_n \sqsubseteq C$ and $R_1 \sqsubseteq R_2$, respectively. Denote by $\sqsubseteq_{\mathcal{T}}^*$ the transitive-reflexive closure of 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*$_{horn}^{\mathcal{N}}$ is the fragment of *DL-Lite*$_{horn}^{(\mathcal{HN})}$ 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 $\mathcal{A}$. 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*$_{horn}^{(\mathcal{HN})}$ is defined in the standard way based on interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$; for details consult (Baader et al. 2003). Throughout the paper, we 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.

Given an interpretation $\mathcal{I}$ and an inclusion or assertion α, we write $\mathcal{I} \models \alpha$ to say that α is true in $\mathcal{I}$. 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 $\vec{v} = v_1, \dots, v_k$ 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 $t, t' \in \mathsf{N_T}$. 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 $\vec{a} = a_1, \dots, a_k$ a k-tuple of individual names. We say that $\vec{a}$ is an *answer* to q in an interpretation $\mathcal{I}$ and write $\mathcal{I} \models q[\vec{a}]$ if $\mathcal{I}$ satisfies q under the assignment π which sets $\pi(v_i) = a_i^{\mathcal{I}}$, $i \leq k$. Such an assignment is called a *match* of q in $\mathcal{I}$. We 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.

ABox Extension

First, we describe the ABox extension part of the combined approach to CQ answering in *DL-Lite*$_{horn}^{\mathcal{N}}$. 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 $R_0, \dots, R_n = R$ such that the following conditions hold:

(agen) $\mathcal{K} \models \exists R_0(a)$ but $R_0(a, b) \notin \mathcal{A}$ for all $b \in \mathsf{Ind}(\mathcal{A})$ (written $a \leadsto c_{R_0}$),

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

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:

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

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

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

Then $\Delta^{\mathcal{I}_\mathcal{K}} = \{a, b, c_P\}$, $P^{\mathcal{I}_\mathcal{K}} = \{(a, c_P), (b, c_P)\}$, and so $c_P \in (\geq 2\, P^-)^{\mathcal{I}_\mathcal{K}}$ 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 by two examples.

Example 2 (i) 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 $\Delta^{\mathcal{I}} = \{a, 1, 2, \dots\}$, $P^{\mathcal{I}} = \{(a, 1)\} \cup \{(n, n+1) \mid n \geq 1\}$ and $A^{\mathcal{I}} = \{a\}$ is a model of $\mathcal{K}$, but $\mathcal{I} \not\models q$. Thus $\mathcal{K} \not\models q$.

(ii) Consider the query $q = \exists v_2\, (P(v_1, v_2) \wedge P(v_3, v_2))$ 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 $a c_{R_1} \cdots c_{R_n}$, $n \geq 0$, such that $a \in \mathsf{Ind}(\mathcal{A})$ and $R_1, \dots, R_n$ satisfy **(agen)** and **(rgen)** (that is, $a \rightsquigarrow c_{R_1}$ and $c_{R_i} \rightsquigarrow c_{R_{i+1}}$). We denote by $\mathsf{paths}(\mathcal{I}_\mathcal{K})$ the set of paths in $\mathcal{I}_\mathcal{K}$ and by $\mathsf{tail}(\sigma)$ the last element in $\sigma \in \mathsf{paths}(\mathcal{I}_\mathcal{K})$. The interpretation $\mathcal{U}_\mathcal{K}$ is then defined by taking:

$$
\begin{aligned}
\Delta^{\mathcal{U}_\mathcal{K}} &= \mathsf{paths}(\mathcal{I}_\mathcal{K}),\\
a^{\mathcal{U}_\mathcal{K}} &= a, \text{ for all } a \in \mathsf{Ind}(\mathcal{A}),\\
A^{\mathcal{U}_\mathcal{K}} &= \{\sigma \mid \mathsf{tail}(\sigma) \in A^{\mathcal{I}_\mathcal{K}}\},\\
P^{\mathcal{U}_\mathcal{K}} &= \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup\\
&\ \{(\sigma, \sigma \cdot c_P) \mid \sigma \cdot c_P \in \mathsf{paths}(\mathcal{I}_\mathcal{K})\} \cup\\
&\ \{(\sigma \cdot c_{P^-}, \sigma) \mid \sigma \cdot c_{P^-} \in \mathsf{paths}(\mathcal{I}_\mathcal{K})\},
\end{aligned}
$$

where '$\cdot$' denotes concatenation. Notice that the interpretations $\mathcal{I}$ constructed in Example 2 are isomorphic to the respective models $\mathcal{U}_\mathcal{K}$. 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 $E = \{(\sigma, \sigma \cdot c_R) \mid \sigma \cdot c_R \in \mathsf{paths}(\mathcal{I}_\mathcal{K})\}$ 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 Example 2, 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 3 *For every consistent DL-Lite*$^{\mathcal{N}}_{horn}$ *KB* $\mathcal{K}$ *and every CQ* q*, we have* $\mathsf{cert}(q, \mathcal{K}) = \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.

Note that Theorem 3 requires consistency of $\mathcal{K}$. We shall see below that there is an FO query $q_\bot^{\mathcal{T}}$ such that $\mathsf{ans}(q_\bot^{\mathcal{T}}, \mathcal{A}) = \emptyset$ 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 $\mathcal{O}(|q|^2 + |q| \cdot |\mathcal{T}|)$ (and can be made $\mathcal{O}(|q|^2)$ by adding an auxiliary database table).

By Theorem 3, 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 with the set of its atoms and assume that $P^-(v, u) \in q$ iff $P(u, v) \in q$. The rewriting $q^\dagger$ of a given CQ $q = \exists \vec{u}\, \varphi$ is defined by taking

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

where φ_1, φ_2 and φ_3 are Boolean combinations of built-in equality predicates over terms in q and constants $c_R \in \mathsf{N}_\mathsf{I}^{\mathcal{T}}$ (which means that they can be evaluated by a RDBMS over the result of evaluating φ). The formula φ_1 is defined as

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

and its purpose 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.

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

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

If a tree witness for $R(t, t')$ in q exists, then it is unique and denoted by $f_{R(t,t')}$.

Intuitively, the tree witness $f_{R(t,t')}$ deals with matches π in $\mathcal{I}_\mathcal{K}$ where $\pi(t') = c_R$. 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, $f_{R(t,t')}$ 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: $f_{R(t,t')}(s) = R_1 \cdots R_k$ 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 $\mathcal{I}_\mathcal{K}$. This is precisely what φ_2 is for:

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

Example 5 We illustrate φ_2 using the query $q = \exists v\, P(v, v)$ and KB $\mathcal{K}$ from Example 2 (i). 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 witness tree $f_{P(v,v)}$ because otherwise we would have $f_{P(v,v)}(v) = P$ and, by $P(v, v) \in q$ and $P \neq P^-$, also $f_{P(v,v)}(v) = P \cdot P$, contrary to $f_{P(v,v)}$ 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 $f_{R(t,t')}$ exists 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 $f_{R(t,t')}$ 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. This is the case precisely when the terms in $\mathfrak{D}$ are matched in $\mathcal{I}_\mathcal{K}$ as prescribed by $f_{R(t,t')}$:

- if $f_{R(t,t')}(s) = R_1 \cdots R_n$, for $n > 0$, then $\pi(s) = c_{R_n}$;
- if $f_{R(t,t')}(s) = \varepsilon$, then $\pi(s) = \pi(t)$.

Both of these conditions are in fact guaranteed by φ_3:

$$\varphi_3 = \bigwedge_{\substack{R(t,t') \in q \\ f_{R(t,t')} \text{ exists}}} \big((t' = c_R) \quad \to \bigwedge_{f_{R(t,t')}(s) = \varepsilon} (s = t)\big).$$

Example 6 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 2 (ii). 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 $(v_2 = c_P) \to (v_1 = v_3)$ as a conjunct to $P(v_1, v_2) \wedge P(v_3, v_2)$. This is achieved using φ_3 as follows: we have $f_{P(v_1,v_2)}(v_2) = P$ and $f_{P(v_1,v_2)}(v_3) = \varepsilon$. Thus, φ_3 contains the conjunct $(v_2 = c_P) \to (v_1 = v_3)$, 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 $\mathsf{N}_\mathsf{I}^{\mathcal{T}}$. In particular, the required functions $f_{R(t,t')}$ can be computed in polytime by a straightforward algorithm akin to breadth-first search, which also decides their existence. The length of $q^\dagger$ is of size $\mathcal{O}(|q|^2 + |q| \cdot |\mathcal{T}|)$, where the $|\mathcal{T}|$ factor is solely due to φ_1. 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 $\mathcal{O}(|q|^2)$.

The main result of this paper is the following theorem:

Theorem 7 *For every DL-Lite$^{\mathcal{N}}_{horn}$ KB $\mathcal{K}$ and every CQ q,* $\mathsf{ans}(q^\dagger, \mathcal{I}_\mathcal{K}) = \mathsf{ans}(q, \mathcal{U}_\mathcal{K})$.

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$^{\mathcal{N}}_{horn}$* KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ can be constructed by means of FO queries. This allows us to implement the construction of $\mathcal{I}_\mathcal{K}$ in an RDBMS using (potentially 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$^{\mathcal{N}}_{horn}$* 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 the interpretation $\mathcal{I}_\mathcal{A}$ defined by taking:

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

Equivalently, we could rely on domain independence of queries and use the active domain semantics (Abiteboul et al. 1995). We now construct $q_A^{\mathcal{T}}(x)$ in three steps. First, for each concept C in $\mathcal{T}$, we define inductively a query $\mathsf{exp}_C^{\mathcal{T}}(x)$ which determines $C^{\mathcal{I}_\mathcal{A}}$. 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 $\mathcal{T}$. In any case, the extended TBox is only linearly larger than the original one. Set

$$\mathsf{exp}_\bot^{\mathcal{T},0}(x) = \bot, \qquad \mathsf{exp}_A^{\mathcal{T},0}(x) = A(x),$$

$$\mathsf{exp}_{\exists R}^{\mathcal{T},0}(x) = (x = c_{R^-}) \lor \exists y\, R(x, y),$$

$$\mathsf{exp}_{\geq mR}^{\mathcal{T},0}(x) = \exists y_1, \dots, y_m \,\big(\bigwedge_{1 \leq i \leq m} R(x, y_i) \land \bigwedge_{i \neq j} (y_i \neq y_j)\big),$$

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

$$\mathsf{exp}_C^{\mathcal{T},j}(x) = \mathsf{exp}_C^{\mathcal{T},j-1}(x) \ \lor \bigvee_{C_1 \sqcap \dots \sqcap C_k \sqsubseteq C} \ \bigwedge_{i=1}^{k} \mathsf{exp}_{C_i}^{\mathcal{T},j-1}(x).$$

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 8 *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 $q_P^{\mathcal{T}}(x, y)$ computing $P^{\mathcal{I}_\mathcal{K}}$. Let $\mathsf{rgen}_{R_n}^{\mathcal{T}}$ be the set of pairs (R_0, R_{n-1}) of roles in $\mathcal{T}$ such that there is a sequence $R_0, \dots, R_n$ 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

$$q_P^{\mathcal{T}}(x, y) = P(x, y) \lor \big(\mathsf{gen}_P^{\mathcal{T}}(x) \land (y = c_P)\big) \ \lor \big(\mathsf{gen}_{P^-}^{\mathcal{T}}(y) \land (x = c_{P^-})\big),$$

$$\mathsf{gen}_R^{\mathcal{T}}(x) = \mathsf{agen}_R^{\mathcal{T}}(x) \ \lor \bigvee\nolimits_{(R_0, S) \in \mathsf{rgen}_R^{\mathcal{T}}} \big(\exists z\, \mathsf{agen}_{R_0}^{\mathcal{T}}(z) \land (x = c_S)\big),$$

$$\mathsf{agen}_R^{\mathcal{T}}(x) = \mathsf{exp}_{\exists R}^{\mathcal{T}}(x) \land \neg\exists y\, R(x, y) \land \bigwedge\nolimits_{c_S \in \mathsf{N}_\mathsf{I}^{\mathcal{T}}} (x \neq c_S).$$

Lemma 9 *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 8, to restrict $\mathsf{exp}_A^{\mathcal{T}}(x)$ to the domain of $\mathcal{I}_\mathcal{K}$. So as the third step we set $q_A^{\mathcal{T}}(x) = \mathsf{exp}_A^{\mathcal{T}}(x) \land D(x)$, where

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

Lemma 10 *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 queries for constructing $\mathcal{I}_\mathcal{K}$ also provide us with the previously announced query $q_\bot^{\mathcal{T}} = \exists x\, (\mathsf{exp}_\bot^{\mathcal{T}}(x) \land D(x))$ which can be used to check consistency of $\mathcal{K}$ (see the remark after Theorem 3): KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent if and only if $\mathsf{ans}(q_\bot^{\mathcal{T}}, \mathcal{I}_\mathcal{A}) = \emptyset$.

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*$_\mathcal{F}$ (Calvanese et al. 2006), 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*$_\mathcal{F}$ 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*$_\mathcal{F}$ produced exponential results. Our technique can also be applied to *DL-Lite*$^{\mathcal{N}}_{core}$, which extends *DL-Lite*$_\mathcal{F}$ 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 $\mathsf{exp}_C^{\mathcal{T},j+1}(x)$ then similar considerations also apply to TBoxes in the full language of *DL-Lite*$^{\mathcal{N}}_{horn}$. Without views or aggregation, the definition of $\mathcal{I}_\mathcal{K}$ is exponential.

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 polytime) 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 7 to answer the UCQ q_h in a component-wise fashion.

We construct q_h in two steps. First, we transform q into a positive existential query q' by replacing each atom $R(t, t')$ in q with the disjunction $\bigvee_{S \sqsubseteq^*_\mathcal{T} R} S(t, t')$. Since the RIs were removed from $\mathcal{T}$, this is now compensated by the construction of q'. 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 11 *For every consistent* $\textit{DL-Lite}^{(\mathcal{HN})}_{\textit{horn}}$ *KB* $\mathcal{K}$ *and positive existential query* q, $\mathsf{cert}(q, \mathcal{K}) = \mathsf{ans}(q^\dagger_h, \mathcal{I}_{\mathcal{K}_h})$.

It is to be noted that if RIs are only used to express symmetry of roles then the exponential blowup can be avoided by modifying the definition of tree witnesses.

Query Answering in $\textit{DL-Lite}^{(\mathcal{HF})}_{\textit{horn}}$ 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. Without UNA, CQ answering in $\textit{DL-Lite}^{\mathcal{N}}_{\textit{core}}$ (with or without $\approx$ and $\not\approx$) becomes coNP-hard for data complexity (Artale et al. 2009). If, however, $\textit{DL-Lite}^{(\mathcal{HN})}_{\textit{horn}}$ 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 $\textit{DL-Lite}^{(\mathcal{HF})}_{\textit{horn}}$) then query answering is 'only' PTime-complete for data complexity. If concepts $\geq m\, R$ with $m \geq 2$ are disallowed then query answering is LogSpace-complete (due to equalities $a \approx b$). In the combined approach, both equality and functionality constraints can be eliminated at the stage of constructing the canonical interpretation. Indeed, given a $\textit{DL-Lite}^{\mathcal{F}}_{\textit{horn}}$ 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 saying in the $\mathsf{exp}^{\mathcal{T},j}_C(x)$ that they can also be obtained from $\mathsf{exp}^{\mathcal{T},j-1}_C(y)$ and $\mathsf{eq}_{\mathcal{K}}(x, y)$; the queries $q^{\mathcal{T}}_P(x, y)$ are modified accordingly. The RIs in $\textit{DL-Lite}^{(\mathcal{HF})}_{\textit{horn}}$ can be treated at the query rewriting stage similarly to $\textit{DL-Lite}^{(\mathcal{HN})}_{\textit{horn}}$ under UNA.

Note also that $\textit{DL-Lite}^{(\mathcal{HF})}_{\textit{horn}}$ 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 FO 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 $\textit{DL-Lite}_{\textit{core}}$, the common fragment of $\textit{DL-Lite}^{\mathcal{N}}_{\textit{horn}}$ and the logic underlying QuOnto. Among the ontologies considered are the $\textit{DL-Lite}_{\textit{core}}$ approximation *Galen-Lite* of the medical ontology *Galen* (consisting of the $\textit{DL-Lite}_{\textit{core}}$ 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 $\textit{DL-Lite}_{\textit{core}}$ 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 shape of the queries for the Galen-lite and Core ontologies is given in the full paper at `http://www.dcs.bbk.ac.uk/~roman/` and, for the Stockexchange and University, in (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, *RIs* were incorporated into the query rewriting as outlined 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 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.) The need for constructing the canonical interpretation, similarly to creating additional auxiliary data structures in standard relational databases (e.g., indices and materialised views), speeds up query processing at the expense of updates. Incremental updates of the ABox can be supported by relying on techniques developed for efficient materialised view maintenance (Colby et al. 1996).

Conclusion

We presented a combined approach to CQ answering in $\textit{DL-Lite}^{\mathcal{N}}_{\textit{horn}}$ and some of its variants and demonstrated that this approach often allows more efficient query execution

	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 2733 concepts, 207 roles, and 4888 axioms	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
	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
	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
	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 81 concepts, 58 roles, and 381 axioms	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
	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
	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
	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 31 concepts, 25 roles, and 103 axioms	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
	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
	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
	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 17 concepts, 12 roles, and 62 axioms	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
	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
	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
	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).

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*$^{\mathcal{N}}_{core}$ in the pure query rewriting approach raises the question whether the exponential blowup can also be avoided in other variants of *DL-Lite*.

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. 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 DC*, vol. 496 of *CEUR-WS.*

Appendix

Proof of Theorem 3

Theorem 3 For every satisfiable *DL-Lite*$^{\mathcal{N}}_{\mathit{horn}}$ 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 $cp(w) = y_R$.

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 $cp(w) = y_S$, 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 $cp(u_i) = y_R$, 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 $a \cdot c_R \in \mathsf{paths}(\mathcal{I}_\mathcal{K})$, $(a, a\cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$, 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, $cp(w) = y_S$. If $c_S \rightsquigarrow c_R$ then $\sigma \cdot c_S \cdot c_R \in \mathsf{paths}(\mathcal{I}_\mathcal{K})$ 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 then $(\sigma \cdot c_S, \sigma) \in R^{\mathcal{U}_\mathcal{K}}$. We then set $h_{n+1}(u_i) = \sigma$, $i \leq l$.

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, $\mathcal{U}_\mathcal{K} \models q[\vec{a}]$. ❑

Proof of Theorem 7

To begin with, we give some more intuition regarding how witness trees are computed. Suppose that we want to compute $f_{R_1(t_0,t_1)}(t_n)$. Assume first that the terms t_0 and t_n are 'connected' in q in the sense that there is a sequence of the form

$$\mathfrak{c} = t_0 R_1 t_1 R_2 t_2 \dots t_{n-1} R_n t_n, \quad R_i(t_{i-1},t_i) \in q. \qquad (1)$$

We will call $\mathfrak{c}$ a *computation of* $f_{R_1(t_0,t_1)}(t_n)$. We can compute $f_{R_1(t_0,t_1)}(t_n)$ by taking first the word $R_1 \cdot R_2 \cdots R_n$ and then successively removing from it all *leftmost* pairs of the form $R \cdot R^-$ (cf. reductions in free groups). If in this removal process we obtain a word of the form $R_i \cdot R_j \cdot w$ with $R_i = R_j^-$ then $f_{R_1(t_0,t_1)}(t_j) = \varepsilon$ and, if $j < n$, $f_{R_1(t_0,t_1)}(t_n)$ is not defined. Otherwise the resulting word is the value of $f_{R_1(t_0,t_1)}(t_n)$, provided that *all* computations of $f_{R_1(t_0,t_1)}(t_n)$ give the same result; if this is not the case then the tree witness for $R_1(t_0,t_1)$ does not exist. If t_0 and t_n are not connected by a computation then $f_{R_1(t_0,t_1)}(t_n)$ is not defined. Note also that if we use the computation $\mathfrak{c}$ to determine $f_{R_n(t_n,t_{n-1})}(t_0)$ then we can take the same word $R_1 \cdot R_2 \cdots R_n$, successively remove from it all *rightmost* pairs of the form $R \cdot R^-$, and then take the inverse. The example below shows that $f_{R_n(t_n,t_{n-1})}(t_0)$ may be undefined even if $f_{R_1(t_0,t_1)}(t_n)$ is defined.

Example 12 Let $q = \{R(t_1,t_2), S(t_2,t_3), S(t_4,t_3)\}$. Then $f_{R(t_1,t_2)}(t_2) = R$, $f_{R(t_1,t_2)}(t_1) = \varepsilon$, $f_{R(t_1,t_2)}(t_3) = R \cdot S$, $f_{R(t_1,t_2)}(t_4) = R$, $f_{S(t_4,t_3)}(t_3) = S$, $f_{S(t_4,t_3)}(t_4) = \varepsilon$, $f_{S(t_4,t_3)}(t_2) = \varepsilon$, and $f_{S(t_4,t_3)}(t_1)$ is not defined. In this case $\varphi_1 = \varphi_2 = \top$ and $\varphi_3 = ((t_3 = c_S) \to (t_2 = t_4))$.

The following easily proved lemma will be used throughout this section.

Lemma 13 *Suppose that* $f_{R_1(t_0,t_1)}(t_n)$ *is defined. Let* $\mathfrak{c}$ *be a computation of* $f_{R_1(t_0,t_1)}(t_n)$ *of the form* (1)*. Then there are* $0 = m_0 < m_1 < \dots < m_k = n$, $k \geq 1$, *such that*

- $f_{R_{m_i}(t_{m_i},t_{m_i-1})}(t_{m_{i-1}}) = \varepsilon$, *for* $1 < i \leq k$, *and*
- $f_{R_{m_1}(t_{m_1},t_{m_1-1})}(t_0)$ *is the inverse of* $f_{R_1(t_0,t_1)}(t_n)$.

Moreover, if $f_{R_1(t_0,t_1)}(t_n) = \varepsilon$ *then* $k = 1$.

In what follows, we exploit the following structural properties of $\mathcal{I}_\mathcal{K}$ and $\mathcal{U}_\mathcal{K}$ that are immediate from the definitions:

(p1) If $(d,d') \in R^{\mathcal{I}_\mathcal{K}}$, then one of the following holds: (i) $d,d' \in \mathsf{Ind}(\mathcal{A})$; (ii) $d' = c_R$ and $d \neq c_{R^-}$; or (iii) $d = c_{R^-}$ and $d' \neq c_R$.

(p2) If $(\sigma,\sigma') \in R^{\mathcal{U}_\mathcal{K}}$, then one of the following holds: (i) $\sigma,\sigma' \in \mathsf{Ind}(\mathcal{A})$; (ii) $\sigma' = \sigma \cdot c_R$ and $\mathsf{tail}(\sigma) \neq c_{R^-}$; or (iii) $\sigma = \sigma' \cdot c_{R^-}$ and $\mathsf{tail}(\sigma') \neq c_R$.

We now prove the main result of this paper:

Theorem 7 For every *DL-Lite*$^{\mathcal{N}}_{\mathit{horn}}$ KB $\mathcal{K}$ and CQ q, we have $\mathsf{ans}(q^\dagger,\mathcal{I}_\mathcal{K}) = \mathsf{ans}(q,\mathcal{U}_\mathcal{K})$.

Proof. ($\Leftarrow$) Let τ be an $\vec{a}$-match for $\mathcal{U}_\mathcal{K}$ and 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 notion. Let $R(t,t') \in q$. Define a relation $X_{R(t,t')} \subseteq \mathsf{term}(q) \times (\mathsf{N}_\mathsf{R}^-)^*$ by taking

$$X_{R(t,t')} = \bigcup_{i \geq 0} X^{(i)}_{R(t,t')},$$

where

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

$X_{R(t,t')}$ is clearly well-defined. Moreover, it should also be clear that if it is a partial function then $f_{R(t,t')} = X_{R(t,t')}$.

Lemma 14 *Suppose that $R(t,t') \in q$ and $\pi(t') = c_R$. Then $(s, S_1 \cdots S_k) \in X_{R(t,t')}$ implies $\tau(s) = \tau(t) \cdot c_{S_1} \cdots c_{S_k}$. It follows that $f_{R(t,t')} = X_{R(t,t')}$.*

Proof. By the definition of $X_{R(t,t')}$, it suffices to show that, for every $i \geq 0$, if $(s, S_1 \cdots S_k) \in X^{(i)}_{R(t,t')}$ 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 consider $(t', R) \in X^{(0)}_{R(t,t')}$. Since $\pi(t') = c_R$, we have $\mathsf{tail}(\tau(t')) = c_R$. So, using $(\tau(t), \tau(t')) \in R^{\mathcal{U}_\mathcal{K}}$ and **(p2)**, we obtain $\tau(t') = \tau(t) \cdot c_R$ as required. For the induction step, we consider two cases. (i) Assume that $(s', R_1 \cdots R_m \cdot S') \in X^{(i+1)}_{R(t,t')}$, $(s, R_1 \cdots R_m) \in X^{(i)}_{R(t,t')}$, $S'(s,s') \in q$ and $S' \neq R_m^-$. Then, by IH, $\tau(s') = \tau(t) \cdot c_{R_1} \cdots c_{R_m}$. Since $S' \neq R_m^-$ and $(\tau(s), \tau(s')) \in (S')^{\mathcal{U}_\mathcal{K}}$, **(p2)** gives us $\tau(s') = \tau(s) \cdot c_{S'}$ as required. (ii) Suppose that $(s', R_1 \cdots R_m) \in X^{(i+1)}_{R(t,t')}$, $(s, R_1 \cdots R_m \cdot S) \in X^{(i)}_{R(t,t')}$ and $S^-(s,s') \in q$. Then, by IH, $\tau(s) = \tau(t) \cdot c_{R_1} \cdots c_{R_m} \cdot c_S$. In view of **(p2)** and $(\tau(s'), \tau(s)) \in S^{\mathcal{U}_\mathcal{K}}$, we obtain $\tau(s') = \tau(t) \cdot c_{R_1} \cdots c_{R_m}$ 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 $\tau(v) \neq c_R$ for any $v \in \mathsf{avar}(q)$ and $c_R \in \mathsf{N}_\mathsf{I}^\mathcal{T}$. 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 $f_{R(t,t')}$ exists, which follows from Lemma 14. Thus, $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$.

φ_3**:** Let $R(t,t') \in q$, $f_{R(t,t')}(s) = \varepsilon$ and $\pi(t') = c_R$. By Lemma 14, $\tau(s) = \tau(t)$, from which $\pi(s) = \pi(t)$. Thus, $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$.

This completes the proof of ($\Leftarrow$).

($\Rightarrow$) 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 problematic.

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 $R(t', t) \in q$.

Lemma 15 *For every non-root $t \in \mathsf{term}(q)$, there is some $R(s,s') \in q$ such that s is a root, $\pi(s') = c_R$ and $f_{R(s,s')}(t)$ is defined.*

Proof. Suppose $t \in \mathsf{term}(q)$ is not a root. By the definition of roots, this means that 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 **(p1)**, 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 $t_j = t_k$. Due to $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$, the tree witness $f_{R_k(t_{k+1}, t_k)}$ exists. Since $R_i \neq R_{i+1}^-$ for all $i \geq 0$, it is easy to see that $f_{R_k(t_{k+1}, t_k)}(t_i) = R_k R_{k-1} \cdots R_i$ for all $i \leq k$. We thus obtain $f_{R_k(t_{k+1}, t_k)}(t_j) = R_k \cdots R_j$, contrary to $f_{R_k(t_{k+1}, t_k)}(t_k) = R_k$ 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 $t_0 = t$, t_ℓ 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 $f_{R_{\ell-1}(t_\ell, t_\ell - 1)}(t) = R_{\ell-1} \cdots R_0$. As $\pi(t_{\ell-1}) = c_{R_{\ell-1}}$, it follows that the atom $R_{\ell-1}(t_\ell, t_{\ell-1})$ is as required. ❑

Example 16 Consider the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with

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

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

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

Consider again the query q from Example 12. 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, $f_{R(t_1,t_2)}(t_4) = R$, while $f_{R(t_4,t_3)}(t_1)$ is not defined.

We shall also require the following two lemmas.

Lemma 17 *If $f_{S(s,s')}(t) = w \cdot R$ and $\pi(s') = c_S$, then $\pi(t) = c_R$.*

Proof. The proof is by induction on the number of steps required to compute $f_{S(s,s')}(t)$. The basis of induction, that is, the case $t = s'$, is trivial. For the induction step, we consider two cases. (i) Assume that $f_{S(s,s')}(t') = w' \cdot Q$, $Q \neq R^-$, $R(t', t) \in q$, and so $f_{S(s,s')}(t) = w' \cdot Q \cdot R$. By IH, $\pi(t') = c_Q$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi$, we have $(c_Q, \pi(t)) \in R^{\mathcal{I}_\mathcal{K}}$. So, by **(p1)**, $\pi(t) = c_R$. (ii) Suppose that $f_{S(s,s')}(t') = w' \cdot Q$, $Q^-(t', t) \in q$, and so $f_{S(s,s')}(t) = w' = w \cdot R$. By IH, $\pi(t') = c_Q$. Take the nearest predecessor t'' of t' in the computation of $f_{S(s,s')}(t')$ with $f_{S(s,s')}(t'') = w \cdot R$, which must exist by the definition of tree witnesses. By IH, $\pi(t'') = c_R$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_2$ and $\pi(t') = c_Q$, $f_{Q(t,t')}$ exists. Moreover, by the choice of t'', we clearly have $f_{Q(t,t')}(t'') = \varepsilon$, from which, by $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, $\pi(t) = \pi(t'') = c_R$. ❑

Lemma 18 *Suppose $f_{S(s,s')}(t)$ is defined and nonempty, $\pi(s') = c_S$, $f_{R(t,t')}(r)$ is defined and $\pi(t') = c_R$. Then $f_{S(s,s')}(r)$ is defined and*

$$f_{S(s,s')}(r) = f_{S(s,s')}(t) \cdot f_{R(t,t')}(r).$$

Proof. Let $f_{S(s,s')}(t) = w \cdot Q$. By Lemma 17, $\pi(t) = c_Q$. In view of $\pi(t') = c_R$ and **(p1)**, $Q \neq R^-$. We can now prove our claim by an easy induction on the number of steps required to compute $f_{S(s,s')}(r)$. The basis of induction—i.e., the case $r = t'$—follows from $f_{S(s,s')}(t) = w \cdot Q$ and $Q \neq R^-$, as we have $f_{S(s,s')}(t') = w \cdot Q \cdot R$. Suppose next that $f_{R(t,t')}(r') = w' \cdot T'$, $T(r', r) \in q$ and $T' \neq T^-$, from which $f_{R(t,t')}(r) = w' \cdot T' \cdot T$. By IH, $f_{S(s,s')}(r') = w \cdot Q \cdot w' \cdot T'$, and so $f_{S(s,s')}(r) = w \cdot Q \cdot w' \cdot T' \cdot T$. Finally, if $f_{R(t,t')}(r') = w' \cdot T$ and $T^-(r', r) \in q$ then $f_{R(t,t')}(r) = w'$. By IH, $f_{S(s,s')}(r') = w \cdot Q \cdot w' \cdot T$, and so $f_{S(s,s')}(r) = w \cdot Q \cdot w'$. ❑

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

Example 19 In the environment of Example 16, 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 $f_{S(t_4,t_3)}(t_1)$ not defined). On the contrary, root t_4 is not initial because $f_{R(t_1,t_2)}(t_4) = R$.

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, $\pi(s') = c_S$, $f_{S(s,s')}(t)$ is defined and $f_{S(s,s')}(t) \neq \varepsilon$. By Lemma 17, we have $\pi(t) = c_R \in \mathsf{N}_\mathsf{I}^\mathcal{T}$ contrary to $\pi(t) \in \mathsf{Ind}(\mathcal{A})$. ❑

We now strengthen Lemma 15 to the following:

Lemma 21 *For each $t \in \mathsf{term}(q)$, either t is an initial root or there is some $R(s, s') \in q$ such that s is an initial root, $\pi(s') = c_R$, $f_{R(s,s')}(t)$ is defined and nonempty.*

Proof. Let $t \in \mathsf{term}(q)$. If t is an initial root, we are done. Suppose now that t is a root but not initial. Then there is some $R_0(s_0, s'_0) \in q$ such that s_0 is a root, $\pi(s'_0) = c_{R_0}$, $f_{R_0(s_0,s'_0)}(t)$ is defined and $f_{R_0(s_0,s'_0)}(t) \neq \varepsilon$. If s_0 is initial, we are done. Otherwise, there is some $R_1(s_1, s'_1) \in q$ such that s_1 is a root, $\pi(s'_1) = c_{R_1}$, $f_{R_1(s_1,s'_1)}(s_0)$ is defined and $f_{R_1(s_1,s'_1)}(s_0) \neq \varepsilon$. Moreover, by Lemma 18, $f_{R_1(s_1,s'_1)}(t) = f_{R_1(s_1,s'_1)}(s_0) \cdot f_{R_0(s_0,s'_0)}(t)$. We can repeat this argument, and each time the word $f_{R_i(s_i,s'_i)}(t)$ becomes strictly longer. Due to finiteness of q, we thus eventually reach an initial root. Finally, if t is not a root, then Lemma 15 gives some $R(s, s') \in q$ such that s is a root, $\pi(s') = c_R$ and $f_{R(s,s')}(t)$ is defined. We can then proceed as in the previous case. ❑

We now show the following:

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

Proof. Consider four possible cases.

Case 1: $f_{S(s,s')}(r) = f_{R(t,t')}(r) = \varepsilon$. As $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, we have $\pi(t) = \pi(r) = \pi(s)$.

Case 2: $f_{S(s,s')}(r) = \varepsilon$ and $f_{R(t,t')}(r) \neq \varepsilon$. Consider some shortest computation of $f_{S(s,s')}(r)$, say, $sSs'S_2s_2 \dots s_{n-1}S_nr$. As $f_{S(s,s')}(r) = \varepsilon$, $S_n = S^-$. And in view of $\mathcal{I}_\mathcal{K} \models^\pi \varphi_3$, we have $\pi(s) = \pi(r)$. It follows that $\pi(s_{n-1}) = \pi(s') = c_S$. By Lemma 13, $f_{S(r,s_{n-1})}(s) = \varepsilon$. And by Lemma 18, $f_{R(t,t')}(s) = f_{R(t,t')}(r) \cdot f_{S(r,s_{n-1})}(s) \neq \varepsilon$, contrary to s being an initial root.

Case 3: $f_{S(s,s')}(r) \neq \varepsilon$ and $f_{R(t,t')}(r) = \varepsilon$ is similar to Case 2.

Case 4: $f_{R(t,t')}(r) \neq \varepsilon$ and $f_{S(s,s')}(r) \neq \varepsilon$. Consider again the shortest computation $sSs'S_2s_2 \dots s_{n-1}S_nr$ of $f_{S(s,s')}(r)$. We claim that $f_{R(t,t')}(s) = \varepsilon$. Indeed, as s is an initial root, $f_{R(t,t')}(s)$ cannot be defined and nonempty. And if $f_{R(t,t')}(s)$ is not defined then there must exist some i, $1 \leq i < n$, such that $f_{R(t,t')}(s_i) = \varepsilon$ (here we set $s' = s_1$). But, as we saw in Case 2, this is impossible because $f_{S(s,s')}(s_i) \neq \varepsilon$. Thus, $f_{R(t,t')}(s) = \varepsilon$ and so, by φ_3, we have $\pi(t) = \pi(s)$. It should also be clear from the discussion after the definition of tree witnesses that $f_{R(t,t')}(r) = f_{S(s,s')}(r)$. ❑

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 $\tau : \mathsf{term}(q) \to \Delta^{\mathcal{U}_\mathcal{K}}$ 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 $R(t, t') \in q$, t is an initial root, $s \in \mathsf{term}(q)$, $\pi(t') = c_R$, $f_{R(t,t')}(s)$ is defined and $f_{R(t,t')}(s) = S_1 \cdots S_k$, then set $\tau(s) = \tau(t) \cdot c_{S_1} \cdots c_{S_k}$.

By Lemma 21, τ is total. To see that it is well-defined, it is 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. It thus 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 17, we have

$$\mathsf{tail}(\tau(t)) = \pi(t), \quad \text{for all } t \in \mathsf{term}(q). \tag{2}$$

By the definition of $\mathcal{U}_\mathcal{K}$ and (2), all concept atoms in q are satisfied by τ. Now let $R(t, t')$ be a role atom in q. The following six cases are possible:

Case 1: $\pi(t)$ and $\pi(t')$ are defined in **(a)**. Then we have $\tau(t) = \pi(t) \in \mathsf{Ind}(\mathcal{A})$ and $\tau(t') = \pi(t') \in \mathsf{Ind}(\mathcal{A})$. Since

$\mathcal{I}_\mathcal{K} \models^\pi q$, we have $((\pi(t), \pi(t')) \in R^{\mathcal{I}_\mathcal{K}}$. By the definition of $\mathcal{U}_\mathcal{K}$, $(\tau(t), \tau(t')) \in R^{\mathcal{U}_\mathcal{K}}$.

Case 2: $\pi(t)$ is defined in **(a)** and $\pi(t')$ in **(b)**. Then $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ and t' is a root. Since $\pi(t') \notin \mathsf{Ind}(\mathcal{A})$ and $R(t,t') \in q$, we obtain $\pi(t') \neq c_R$. As $(\pi(t), \pi(t')) \in R^{\mathcal{I}_\mathcal{K}}$ and in view of **(p1)**, this is a contradiction, so this case is impossible.

Case 3: $\pi(t)$ is defined in **(a)** and $\pi(t')$ in **(c)**. Then we have $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ and $\pi(t') \notin \mathsf{Ind}(\mathcal{A})$. By **(p1)**, $(\pi(t), \pi(t')) \in R^{\mathcal{I}_\mathcal{K}}$ implies $\pi(t') = c_R$. By Lemma 20, t is an initial root. By the definition of tree witnesses, $f_{R(t,t')}(t') = R$. By Lemma 22, **(c)** and **(a)**, we thus have $\tau(t') = \pi(t) \cdot c_R$. By the definition of $\mathcal{U}_\mathcal{K}$, $(\pi(t), c_R) \in R^{\mathcal{I}_\mathcal{K}}$ implies $(\pi(t), \pi(t) \cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$.

Case 4: $\pi(t)$ and $\pi(t')$ are defined in **(b)**. Then $\pi(t) = c_S$ for some S such that there is no $S(s,t) \in q$. Clearly, this implies $S \neq R^-$. By $(\mathbf{p1})$ and since $(\pi(t), \pi(t')) \in R^{\mathcal{I}_\mathcal{K}}$, it follows that $\pi(t') = c_R$, contrary to $\pi(t')$ being a root. So this case is impossible.

Case 5: $\pi(t)$ is defined in **(b)** and $\pi(t')$ in **(c)**. Then $\pi(t) = c_S$ for some S such that there is no $S(s,t) \in q$, and so $S \neq R^-$. By $(\mathbf{p1})$ and since $(\pi(t), \pi(t')) \in R^{\mathcal{I}_\mathcal{K}}$, it follows that $\pi(t') = c_R$. By the definition of tree witnesses, $f_{R(t,t')}(t') = R$. By Lemma 22, **(c)** and **(b)**, we thus have $\tau(t) = \gamma(\pi(t))$ and $\tau(t') = \gamma(\pi(t)) \cdot c_R$. By the definition of $\mathcal{U}_\mathcal{K}$, $(\pi(t), c_R) \in R^{\mathcal{I}_\mathcal{K}}$ implies $(\gamma(\pi(t)), \gamma(\pi(t)) \cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$.

Case 6: $\pi(t)$ and $\pi(t')$ are defined in **(c)**. Then there is an $S(s,s') \in q$ such that s is an initial root, $\pi(s') = c_S$, $f_{S(s,s')}(t) = S_1 \cdots S_k$, and $\tau(t) = \tau(s) \cdot c_{S_1} \cdots c_{S_k}$. Since $R(t,t') \in q$, we have to consider two subcases:

Case 6.1: $R \neq S_k^-$. By the definition of tree witnesses, $f_{S(s,s')}(t') = S_1 \cdots S_k \cdot R$. By Lemma 22 and the formulation of **(c)**, we thus have $\tau(t') = \tau(s) \cdot c_{S_1} \cdots c_{S_k} \cdot c_R$. By (2), $\pi(t) = c_{S_k}$ and $\pi(t') = c_R$. Thus $(c_{S_k}, c_R) \in R^{\mathcal{I}_\mathcal{K}}$ and $(\tau(t) \cdot c_{S_1} \cdots c_{S_k}, \tau(t) \cdot c_{S_1} \cdots c_{S_k} \cdot c_R) \in R^{\mathcal{U}_\mathcal{K}}$ by the definition of $\mathcal{U}_\mathcal{K}$.

Case 6.2: $R = S_k^-$. By the definition of tree witnesses, $f_{S(s,s')}(t') = S_1 \cdots S_{k-1}$. By Lemma 22 and the formulation of **(c)**, we thus have $\tau(t') = \tau(s) \cdot c_{S_1} \cdots c_{S_{k-1}}$. By (2), $\pi(t) = c_{S_k}$ and $\pi(t') = c_{S_{k-1}}$. Thus $(c_{S_k}, c_{S_{k-1}}) \in R^{\mathcal{I}_\mathcal{K}}$ and $(\tau(t) \cdot c_{S_1} \cdots c_{S_k}, \tau(t) \cdot c_{S_1} \cdots c_{S_{k-1}}) \in R^{\mathcal{U}_\mathcal{K}}$ by the definition of $\mathcal{U}_\mathcal{K}$.

This completes the proof of Theorem 7. ❑

Proof of Theorem 11

Theorem 11. For every consistent *DL-Lite*$_{horn}^{(\mathcal{HN})}$-KB $\mathcal{K}$ and positive existential query q, $\mathsf{cert}(q, \mathcal{K}) = \mathsf{ans}(q_h^\dagger, \mathcal{I}_{\mathcal{K}_h})$.

Proof. $(\Rightarrow)$ Let q and q_h be k-ary and $\vec{a}$ a k-tuple of individual names from $\mathcal{A}$. First assume that $(\mathcal{T}_h, \mathcal{A}) \not\models q_h[\vec{a}]$. Then there is a model $\mathcal{I}_h$ of $(\mathcal{T}_h, \mathcal{A})$ 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

$$P^\mathcal{I} = \{(d_1, d_2) \mid (d_1, d_2) \in S^{\mathcal{I}_h} \text{ for } S \in \mathsf{N}_\mathsf{R}^-,\ S \sqsubseteq_\mathcal{T}^* P\}$$

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

(i) By definition, $\mathcal{I} \models R \sqsubseteq S$ whenever $R \sqsubseteq S \in \mathcal{T}$. It remains to show that $\mathcal{I} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$, for $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C \in \mathcal{T}$. Let $d \in (C_1 \sqcap \cdots \sqcap C_n)^\mathcal{I}$. We have to show that $d \in C^\mathcal{I}$. This follows if $d \in (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_h}$ because $\mathcal{I} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ 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 $\geq m\, R$. But then there exists d' and S such that $(d, d') \in S^{\mathcal{I}_h}$, $S \sqsubseteq_\mathcal{T}^* R$, and $S \neq R$. By the definition of *DL-Lite*$_{horn}^{(\mathcal{HN})}$, this implies 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.

(ii) Assume that $\mathcal{I} \models^\pi q$. We show that $\mathcal{I}_h \models^\pi q_h$ and thus derive a contradiction. As q_h 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. (i) For $A(t)$, we have $\mathcal{I}_h \models^\pi A(t)$ whenever $\mathcal{I} \models^\pi A(t)$ because $A^\mathcal{I} = A^{\mathcal{I}_h}$. (ii) 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 construction of $\mathcal{I}$, there exists S with $S \sqsubseteq_\mathcal{T}^* R$ and $(\pi(t), \pi(t')) \in S^{\mathcal{I}_h}$ and thus, $\mathcal{I}_h \models^\pi \bigvee_{S \sqsubseteq_\mathcal{T}^* R} S(t, t')$. As q_h is a built from these subformulas using only conjunction and disjunction, we have $\mathcal{I}_h \models^\pi q_h$ whenever $\mathcal{I} \models^\pi q$. Thus, $\mathcal{I}_h \models q_h[\vec{a}]$.

$(\Leftarrow)$ Let $(\mathcal{T}_h, \mathcal{A}) \models q_h[\vec{a}]$. Then $(\mathcal{T}_h \cup \mathcal{T}, \mathcal{A}) \models q[\vec{a}]$. But then $\mathcal{T} \models \mathcal{T}_h$ implies $(\mathcal{T}, \mathcal{A}) \models q[\vec{a}]$. ❑