Conjunctive Query Answering in the Description Logic $\mathcal{EL}$ Using a Relational Database System

Carsten Lutz
Universität Bremen, Germany
clu@informatik.uni-bremen.de

David Toman
University of Waterloo, Canada
david@uwaterloo.ca

Frank Wolter
University of Liverpool, UK
wolter@liverpool.ac.uk

Abstract

Conjunctive queries (CQ) are fundamental for accessing description logic (DL) knowledge bases. We study CQ answering in (extensions of) the DL $\mathcal{EL}$, which is popular for large-scale ontologies and underlies the designated OWL2-EL profile of OWL2. Our main contribution is a novel approach to CQ answering that enables the use of standard relational database systems as the basis for query execution. We evaluate our approach using the IBM DB2 system, with encouraging results.

1 Introduction

One of the main applications of ontologies in computer science is in data access, where an ontology formalizes conceptual information about data that is stored in one or multiple data sources, and this information is used to derive answers to queries over such sources. This general setup plays a central role in, e.g., ontology-based information integration and peer-to-peer data management. In these and similar applications, Description Logics (DLs) are popular ontology languages and conjunctive queries (CQs) are used as a fundamental querying tool. Hence, efficient and scalable approaches to CQ answering over DL ontologies are of great interest.

Calvanese et al. have argued that, in the short run, true scalability of conjunctive query answering over DL ontologies can only be achieved by making use of standard relational database management systems (RDBMSs) [2007b]. Alas, this is not straightforward as RDBMSs are unaware of TBoxes (the DL mechanism for storing conceptual information) and adopt the closed-world semantics. In contrast, ABoxes (the DL mechanism for storing data) and the associated ontologies employ the open-world semantics. Existing approaches to overcome these differences have serious limitations. For example, the approach of [Calvanese *et al.*, 2007b] applies only to DLs with data complexity of CQ answering in LOGSPACE whereas for many DLs, this problem is complete for PTIME or co-NP. In particular, the above approach cannot be directly used for DLs that admit qualified existential restrictions, which play an important role in many ontologies. This limitation is shared by the rule-based approach presented in [Wu *et al.*, 2008].

In this paper, we present a novel approach to using RDBMSs for CQ answering over DL ontologies that, in particular, accommodates qualified existential restrictions. We apply it to the $\mathcal{EL}$ family of DLs [Baader *et al.*, 2008], whose members are widely used as ontology languages for large-scale bio-medical ontologies such as SNOMED CT and (early versions of) NCI. Our main result shows that CQ answering in $\mathcal{ELH}_{\bot}^{dr}$, the extension of basic $\mathcal{EL}$ with the bottom concept, role inclusions, and domain and range restrictions, can be implemented using an RDBMS. This result is of particular relevance as $\mathcal{ELH}_{\bot}^{dr}$ can be viewed as the core of the designated OWL-EL profile of the upcoming OWL Version 2 ontology language. We evaluate our approach using the IBM DB2 RDBMS and show that it scales to TBoxes with more than 50.000 axioms where answer times typically range from a fraction of a second to a few seconds.

The central idea of our approach is to incorporate the consequences of the TBox $\mathcal{T}$ into the relational instance corresponding to the given ABox $\mathcal{A}$. To capture this formally, we introduce the notion of *combined first-order (FO) rewritability*. A DL enjoys combined FO rewritability if it is possible to effectively rewrite (i) $\mathcal{A}$ and $\mathcal{T}$ into an FO structure (independently of q) and (ii) q and (possibly) $\mathcal{T}$ into an FO query q^* (independently of $\mathcal{A}$) such that query answers are preserved, i.e, the answer to q^* over the FO structure is the same as the answer to q over $\mathcal{A}$ and $\mathcal{T}$. The connection to RDBMSs then relies on the well-known equivalence between FO structures and relational databases, and FO queries and SQL queries. The notion of combined FO rewritability generalizes the notion of *FO reducibility*, where the TBox is incorporated into the query q rather than into the ABox $\mathcal{A}$ while the ABox itself is used as a relational instance without any modification [Calvanese *et al.*, 2007b]. Notable properties of our approach include:

1. It applies to DLs for which data complexity of CQ answering is PTIME-complete, such as $\mathcal{ELH}_{\bot}^{dr}$.
2. For $\mathcal{ELH}_{\bot}^{dr}$, both rewriting steps can be carried out in polynomial time and produce only a polynomial blowup; moreover, the query rewriting only depends on the input CQ and the role inclusions in $\mathcal{T}$ (usually only few), but not on $\mathcal{T}$'s concept inclusions (usually very many).

In contrast and to the best of our knowledge, all existing approaches to (non-combined) FO reducibility generate a

rewritten query of size m^n, with m the number of symbols in the query and the TBox and n the size of the query.

In addition, we analyze the limitations of our approach and show that DLs with data complexity exceeding PTIME cannot enjoy *polynomial* combined FO rewritability, where the expansion of the ABox due to rewriting (but not necessarily of query rewriting) is bounded by a polynomial. We believe that this is a significant result as combined FO rewritability involving an exponential blowup of the data does not seem to be practical. In particular, the result implies that expressive DLs such as $\mathcal{ALC}$ cannot enjoy polynomial FO rewritability. The same holds even for $\mathcal{EL}$ enriched with negated ABox assertions and negated query atoms. For the latter case, we sketch an approach to query answering that involves only a polynomial blowup of the ABox, but is incomplete (in a way precisely characterized by an alternative semantics).

Due to space limitations, detailed proofs are given in the technical report [Lutz *et al.*, 2009].

2 Preliminaries

In $\mathcal{ELH}^{dr}_{\bot}$, *concepts* are built according to the syntax rule

$$C ::= A \mid \top \mid \bot \mid C \sqcap D \mid \exists r.C$$

where, here and in the remaining paper, A ranges over *concept names* taken from a countably infinite set $\mathsf{N_C}$, r ranges over *role names* taken from a countably infinite set $\mathsf{N_R}$, and C, D range over concepts. A *TBox* is a finite set of *concept inclusions* $C \sqsubseteq D$, *role inclusions* $r \sqsubseteq s$, *domain restrictions* $\mathsf{dom}(r) \sqsubseteq C$, and *range restrictions* $\mathsf{ran}(r) \sqsubseteq C$. An ABox is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$, where a, b range over a countably infinite set $\mathsf{N_I}$ of *individual names*. A *knowledge base* is a pair $(\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ a TBox and $\mathcal{A}$ an ABox.

The semantics of concepts, inclusions, restrictions, and assertions is given as usual in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$; we refer to [Baader *et al.*, 2008] for details. An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ (ABox $\mathcal{A}$) if it satisfies all inclusions and restrictions in $\mathcal{T}$ (assertions in $\mathcal{A}$). It is a *model* of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of $\mathcal{T}$ and $\mathcal{A}$. A knowledge base that has a model is *consistent*. For a concept inclusion, role inclusion, or assertion α, we write $\mathcal{K} \models \alpha$ if α is satisfied in all models of $\mathcal{K}$.

Let $\mathsf{N_V}$ be a countably infinite set of *variables*. Together, the sets $\mathsf{N_V}$ (of variables) and $\mathsf{N_I}$ (of individual names) form the set $\mathsf{N_T}$ of *terms*. A *first-order (FO) query* q is a first-order formula built from $\mathsf{N_T}$ and the unary and binary predicates from $\mathsf{N_C}$ and $\mathsf{N_R}$. We write $q = \varphi(\vec{v})$ to indicate that q is the FO formula φ whose free variables are among the variables $\vec{v} = v_1, \ldots, v_k$. Variables in $\vec{v}$ are the *answer variables* of q and q is *k-ary* if there are k answer variables. A *conjunctive query* is an FO query q of the form $\exists \vec{u}.\psi(\vec{u}, \vec{v})$, where ψ is a conjunction of *concept atoms* $A(t)$ and *role atoms* $r(t, t')$. The variables in $\vec{u}$ are the *quantified variables* of q. We use $\mathsf{var}(q)$ to denote the set of all variables in $\vec{u}$ and $\vec{v}$, $\mathsf{qvar}(q)$ for the set of quantified variables, $\mathsf{avar}(q)$ for the answer variables, and $\mathsf{term}(q)$ for the terms in q. Abusing notation, we write $\alpha \in q$ if the concept or role atom α occurs in q.

For $\mathcal{I}$ an interpretation, $q = \varphi(\vec{v})$ a k-ary FO query, and $a_1, \ldots, a_k \in \mathsf{N_I}$, we write $\mathcal{I} \models q[a_1, \ldots, a_k]$ if $\mathcal{I}$ satisfies q with v_i assigned to $a_i^{\mathcal{I}}$, $1 \leq i \leq k$. A *certain answer* for a k-ary conjunctive query q and a knowledge base $\mathcal{K}$ is a tuple $(a_1, \ldots, a_k)$ such that $a_1, \ldots, a_k$ occur in $\mathcal{K}$ and $\mathcal{I} \models q[a_1, \ldots, a_k]$ for each model $\mathcal{I}$ of $\mathcal{K}$. We use $\mathsf{cert}(q, \mathcal{K})$ to denote the set of all certain answers for q and $\mathcal{K}$. This defines the query answering problem studied in this paper: given an $\mathcal{ELH}^{dr}_{\bot}$ knowledge base $\mathcal{K}$ and a CQ q, to compute $\mathsf{cert}(q, \mathcal{K})$.

Note that CQ answering generalizes *instance checking*, the problem of deciding whether $\mathcal{K} \models C(a)$, for C an $\mathcal{ELH}^{dr}_{\bot}$-concept: any such C can be easily unfolded into a conjunctive query q_C such that $a \in \mathsf{cert}(q_C, \mathcal{K})$ iff $\mathcal{K} \models C(a)$.

In the remainder of this paper we use the *unique name assumption*: $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all interpretations $\mathcal{I}$ and all $a, b \in \mathsf{N_I}$ with $a \neq b$. It is not hard to see that this has no impact on certain answers. We also assume, w.l.o.g., that (i) queries contain only individual names that occur in the KB against which they are asked, (ii) TBoxes do not contain domain restrictions, (iii) TBoxes contain exactly one range restriction per role name, (iv) if $\mathcal{K} \models r \sqsubseteq s$ and $\mathsf{ran}(r) \sqsubseteq C, \mathsf{ran}(s) \sqsubseteq D$ are in $\mathcal{T}$, then $C \sqsubseteq_{\mathcal{T}} D$, and (v) there are no $r, s \in \mathsf{N_R}$ with $r \neq s$, $\mathcal{K} \models r \sqsubseteq s$, and $\mathcal{K} \models s \sqsubseteq r$. Justifications for these assumptions are given in [Lutz *et al.*, 2009].

3 ABox Rewriting / Canonical Models

The rewriting of the ABox consists of an extension of the ABox to a canonical model of the knowledge base. For the remainder of this section, we fix an $\mathcal{ELH}^{dr}_{\bot}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. We use $\mathsf{sub}(\mathcal{T})$ to denote the set of all subconcepts of concepts that occur in $\mathcal{T}$, $\mathsf{rol}(\mathcal{T})$ for the set of role names that occur in $\mathcal{T}$, and $\mathsf{Ind}(\mathcal{A})$ for the set of individual names that occur in $\mathcal{A}$. We also use $\mathsf{ran}_{\mathcal{T}}(r)$ to denote the (unique) concept C with $\mathsf{ran}(r) \sqsubseteq C \in \mathcal{T}$, and set

$$\begin{aligned} \mathsf{ran}(\mathcal{T}) &:= \{\mathsf{ran}_{\mathcal{T}}(r) \mid r \in \mathsf{rol}(\mathcal{T})\} \\ \mathsf{NI}_{\mathsf{aux}} &:= \{x_{C,D} \mid C \in \mathsf{ran}(\mathcal{T}) \text{ and } D \in \mathsf{sub}(\mathcal{T})\}, \end{aligned}$$

assuming $\mathsf{N_I} \cap \mathsf{N_{I_{aux}}} = \emptyset$. The canonical model $\mathcal{I}_{\mathcal{K}}$ of $\mathcal{K}$ is defined in Figure 1. It is standard to show the following (similar to proofs in [Baader *et al.*, 2005b; Lutz and Wolter, 2007]):

Proposition 1. *If $\mathcal{K}$ is consistent, then $\mathcal{I}_{\mathcal{K}}$ is a model of $\mathcal{K}$.*

Note that the cardinality of $\Delta^{\mathcal{I}_{\mathcal{K}}}$ is only quadratic in the size of $\mathcal{K}$, and linear if $\mathcal{T}$ does not contain range restrictions. The model $\mathcal{I}_{\mathcal{K}}$ can be computed in polynomial time since subsumption and instance checking in $\mathcal{ELH}^{dr}_{\bot}$ can be decided in poly-time [Baader *et al.*, 2008]. In fact, there is an easy rule-based procedure for computing the canonical model and the rules can be implemented as standard database operations; see the workshop publication [Lutz *et al.*, 2008] for details. Consistency of $\mathcal{K}$ can also be checked in polynomial time [Baader *et al.*, 2008]. $\mathcal{I}_{\mathcal{K}}$ can be used for instance checking: it can be shown that $\mathcal{K} \models C(a)$ iff $\mathcal{I}_{\mathcal{K}} \models C(a)$, for $\mathcal{K}$ consistent, C an $\mathcal{ELH}^{dr}_{\bot}$-concept, and $a \in \mathsf{Ind}(\mathcal{A})$. Unfortunately, an analogous statement for conjunctive queries, namely $(a_1, \ldots, a_k) \in \mathsf{cert}(q, \mathcal{K})$ iff $\mathcal{I}_{\mathcal{K}} \models q[a_1, \ldots, q_k]$,

$$
\begin{array}{rcl}
\Delta^{\mathcal{I}_\mathcal{K}} & := & \mathsf{Ind}(\mathcal{A}) \uplus \mathsf{NI}_{\mathsf{aux}} \text{ and } a^{\mathcal{I}_\mathcal{K}} := a \text{ for all } a \in \mathsf{Ind}(\mathcal{A}) \\
A^{\mathcal{I}_\mathcal{K}} & := & \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \{x_{C,D} \in \mathsf{NI}_{\mathsf{aux}} \mid \mathcal{K} \models C \sqcap D \sqsubseteq A\} \\
r^{\mathcal{I}_\mathcal{K}} & := & \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid s(a,b) \in \mathcal{A} \text{ and } \mathcal{K} \models s \sqsubseteq r\} \cup \\
 & & \{(a, x_{C,D}) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{NI}_{\mathsf{aux}} \mid \mathcal{K} \models \exists s.D(a), \mathsf{ran}_\mathcal{T}(s) = C, \text{ and } \mathcal{K} \models s \sqsubseteq r\} \cup \\
 & & \{(x_{C,D}, x_{C',D'}) \in \mathsf{NI}_{\mathsf{aux}} \times \mathsf{NI}_{\mathsf{aux}} \mid \mathcal{K} \models C \sqcap D \sqsubseteq \exists r.D', \mathsf{ran}_\mathcal{T}(s) = C', \text{ and } \mathcal{K} \models s \sqsubseteq r\}
\end{array}
$$

Figure 1: The canonical model $\mathcal{I}_\mathcal{K}$.

does not hold. This is due to two reasons, the first one being that $\mathcal{I}_\mathcal{K}$ may contain unnecessary elements:

Example 2. Take $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ with $\mathcal{T}_1 = \{A \sqsubseteq A\}$ and $\mathcal{A}_1 = \{B(a)\}$, and $q_1 = \exists u.(B(v) \wedge A(u))$. Then $x_{\top,A} \in A^{\mathcal{I}_{\mathcal{K}_1}}$ and so $\mathcal{I}_{\mathcal{K}_1} \models q_1[a]$, but clearly $a \notin \mathsf{cert}(q_1, \mathcal{K}_1)$.

This deficiency of $\mathcal{I}_\mathcal{K}$ is easily repaired by restricting it to elements reachable from some $a^{\mathcal{I}_\mathcal{K}}$ with $a \in \mathsf{Ind}(\mathcal{A})$. Formally, we define $\mathsf{Ind}(\mathcal{A})^\mathcal{I} = \{a^\mathcal{I} \mid a \in \mathsf{Ind}(\mathcal{A})\}$ for each interpretation $\mathcal{I}$. A *path* in $\mathcal{I}$ is a finite sequence $d_0 r_1 d_1 \cdots r_n d_n$, $n \geq 0$, where $d_0 \in \mathsf{Ind}(\mathcal{A})^\mathcal{I}$ and $(d_i, d_{i+1}) \in r_{i+1}^\mathcal{I}$ for all $i < n$. We use $\mathsf{paths}_\mathcal{A}(\mathcal{I})$ to denote the set of all paths in $\mathcal{I}$ and for all $p \in \mathsf{paths}_\mathcal{A}(\mathcal{I})$, $\mathsf{tail}(p)$ to denote the last element d_n in p.

Now $\mathcal{I}_\mathcal{K}^r$ denotes the restriction of $\mathcal{I}_\mathcal{K}$ to those $d \in \Delta^{\mathcal{I}_\mathcal{K}}$ for which there exists a $p \in \mathsf{paths}_\mathcal{A}(\mathcal{I}_\mathcal{K})$ such that $d = \mathsf{tail}(p)$. Then $\mathcal{I}_\mathcal{K}^r$ provides the correct certain answers to the query q_1 from Example 2. A much more severe deficiency of $\mathcal{I}_\mathcal{K}$ (and $\mathcal{I}_\mathcal{K}^r$) is the following:

Example 3. Take $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A}_2)$ with $\mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \exists s.B\}$ and $\mathcal{A}_2 = \{A(a)\}$, and $q_2 = \exists u.(r(v,u) \wedge s(v,u))$. Then $(a, x_{\top,B}) \in r^{\mathcal{I}_{\mathcal{K}_2}}$ and $(a, x_{\top,B}) \in s^{\mathcal{I}_{\mathcal{K}_2}}$ and therefore $\mathcal{I}_{\mathcal{K}_2} \models q_2[a]$, but clearly $a \notin \mathsf{cert}(q_2, \mathcal{K}_2)$.

In principle, this problem can be overcome by replacing $\mathcal{I}_\mathcal{K}^r$ with its unraveling into a less constrained, tree-like model. In the following, we introduce unraveling as a general operation on models. Let $\mathcal{R}$ be the set of role inclusions in $\mathcal{K}$. The $(\mathcal{A}, \mathcal{R})$-*unraveling* $\mathcal{J}$ of $\mathcal{I}$ is defined as follows:

$$
\begin{array}{rcl}
\Delta^\mathcal{J} & := & \mathsf{paths}_\mathcal{A}(\mathcal{I}) \text{ and } a^\mathcal{J} := a^\mathcal{I} \text{ for all } a \in \mathsf{Ind}(\mathcal{A}) \\
A^\mathcal{J} & := & \{p \mid \mathsf{tail}(p) \in A^\mathcal{I}\} \\
r^\mathcal{J} & := & \{(d,e) \mid d, e \in \mathsf{Ind}(\mathcal{A})^\mathcal{I} \wedge (d,e) \in r^\mathcal{I}\} \cup \\
 & & \{(p, p \cdot se) \mid p, p \cdot se \in \Delta^\mathcal{J} \text{ and } \mathcal{R} \models s \sqsubseteq r\}
\end{array}
$$

where "$\cdot$" denotes concatenation. Denote by $\mathcal{U}_\mathcal{K}$ the $(\mathcal{A}, \mathcal{R})$-unraveling of $\mathcal{I}_\mathcal{K}^r$. Notice that the construction of $\mathcal{U}_\mathcal{K}$ from $\mathcal{I}_\mathcal{K}^r$ does not depend on the concept inclusions in $\mathcal{T}$, but only on $\mathcal{R}$. The following result is proved similarly to the analogous result for the DL $\mathcal{ELI}^f$ in [Krisnadhi and Lutz, 2007].

Proposition 4. *If $\mathcal{K}$ is consistent, then for all k-ary conjunctive queries q and all $a_1, \ldots, a_k \in \mathsf{Ind}(\mathcal{A})$, we have $(a_1, \ldots, a_k) \in \mathsf{cert}(q, \mathcal{K})$ iff $\mathcal{U}_\mathcal{K} \models q[a_1, \ldots, a_k]$.*

By Proposition 4, $\mathcal{U}_\mathcal{K}$ gives the correct answers to conjunctive queries, but in contrast to $\mathcal{I}_\mathcal{K}^r$, it is typically infinite. Thus, we do not use it as a target for ABox rewriting and work with $\mathcal{I}_\mathcal{K}^r$. To overcome the problem indicated in Example 3 and similar ones (see [Lutz *et al.*, 2008]), we use query rewriting.

4 Query Rewriting

Our aim is to rewrite the original CQ q into an FO query $q_\mathcal{R}^*$ such that $\mathcal{U}_\mathcal{K} \models q[a_1, \ldots, a_k]$ iff $\mathcal{I}_\mathcal{K}^r \models q_\mathcal{R}^*[a_1, \ldots, a_k]$ for all $a_1, \ldots, a_k \in \mathsf{Ind}(\mathcal{A})$. By Proposition 4, we obtain the desired answers $\mathsf{cert}(q, \mathcal{K})$ by using $\mathcal{I}_\mathcal{K}^r$ (the rewriting of $\mathcal{A}$) as a relational database instance and replacing q with $q_\mathcal{R}^*$. We now formulate the main result of this paper.

Theorem 5. *For every finite set of role inclusions $\mathcal{R}$ and k-ary CQ q, one can construct in polynomial time a k-ary FO query $q_\mathcal{R}^*$ such that for all $\mathcal{ELH}_\bot^{dr}$-KBs $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{R}$ the set of role inclusions in $\mathcal{T}$ and all $a_1, \ldots, a_k \in \mathsf{Ind}(\mathcal{A})$, we have $\mathcal{I}_\mathcal{K}^r \models q_\mathcal{R}^*[a_1, \ldots, a_k]$ iff $\mathcal{U}_\mathcal{K} \models q[a_1, \ldots, a_k]$.*

In the remainder of this section, we show how to construct $q_\mathcal{R}^*$. The query $q_\mathcal{R}^*$ contains one additional unary predicate $\mathsf{Aux}(x)$; we assume that Aux is always interpreted as $\Delta^{\mathcal{I}_\mathcal{K}^r} \setminus \mathsf{Ind}(\mathcal{A})^{\mathcal{I}_\mathcal{K}^r}$ in $\mathcal{I}_\mathcal{K}^r$. Fix a finite set $\mathcal{R}$ of role inclusions and a k-ary conjunctive query q. To construct $q_\mathcal{R}^*$, we use several auxiliary definitions. Let $\sim_q$ denote the smallest relation on $\mathsf{term}(q)$ that includes the identity relation, is transitive, and satisfies the following closure condition:

$(*)$ if $r_1(s,t), r_2(s',t') \in q$ with $t \sim_q t'$, then $s \sim_q s'$.

The relation $\sim_q$ is central to our rewriting procedure. To see this, let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be such that the role inclusions of $\mathcal{T}$ coincide with $\mathcal{R}$. Intuitively, $\mathcal{U}_\mathcal{K}$ is produced from $\mathcal{I}_\mathcal{K}^r$ by keeping the $\mathsf{Ind}(\mathcal{A})$-part intact and relaxing the Aux-part into a collection of trees. To understand $(*)$, first assume $t = t'$. Then $(*)$ describes a non-tree situation in the query since $t = t'$ has two predecessors s and s'. Therefore, any match of the query in $\mathcal{I}_\mathcal{K}^r$ that maps t to the Aux-part should map s and s' to the same element; otherwise such a match cannot be reproduced in $\mathcal{U}_\mathcal{K}$. The case where $t \sim_q t'$ instead of $t = t'$ can be understood inductively.

It is not hard to verify that $\sim_q$ is an equivalence relation and can be computed in time polynomial in the size of q. For $t \in \mathsf{term}(q)$, we use $[t]$ to denote the equivalence class of t w.r.t. $\sim_q$ and define, for any equivalence class ζ of $\sim_q$, the sets:

$$\mathsf{pre}(\zeta) := \{t \mid r(t,t') \in q \text{ for some } r \in \mathsf{N_R} \text{ and } t' \in \zeta\}, \text{and}$$
$$\mathsf{in}(\zeta) := \{r \mid r(t,t') \in q \text{ for some } t \in \mathsf{term}(q) \text{ and } t' \in \zeta\}.$$

For $R \subseteq \mathsf{N_R}$ and $r \in \mathsf{N_R}$, r is called an *implicant* of R if $\mathcal{R} \models r \sqsubseteq s$ for all $s \in R$. It is called a *prime implicant* if, additionally, $\mathcal{R} \not\models r \sqsubseteq r'$ for all implicants r' of R with $r \neq r'$. By assumption (v) in Section 2, there is a prime implicant for any set $R \subseteq \mathsf{N_R}$ for which there is an implicant. We define:

- $\mathsf{Fork}_=$ is the set of pairs $(\mathsf{pre}(\zeta), \zeta)$ with $\mathsf{pre}(\zeta)$ of cardinality at least two;

- $\mathsf{Fork}_{\neq}$ is the set of variables $v \in \mathsf{qvar}(q)$ such that there is no implicant of $\mathsf{in}([v])$;
- $\mathsf{Fork}_{\mathcal{H}}$ is the set of of pairs (I, ζ) such that $\mathsf{pre}(\zeta) \neq \emptyset$, there is a prime implicant of $\mathsf{in}(\zeta)$ that is not contained in $\mathsf{in}(\zeta)$, and I is the set of all prime implicants of $\mathsf{in}(\zeta)$;
- Cyc is the set of variables $v \in \mathsf{qvar}(q)$ such that there are $r_0(t_0, t_0'), \ldots, r_m(t_m, t_m'), \ldots, r_n(t_n, t_n') \in q$, $n, m \geq 0$, with $v \sim_q t_i$ for some $i \leq n$, $t_i' \sim_q t_{i+1}$ for all $i < n$, and $t_n' \sim_q t_m$.

It is not hard to see that $\mathsf{Fork}_{=}$, $\mathsf{Fork}_{\neq}$, $\mathsf{Fork}_{\mathcal{H}}$, and Cyc can be computed in time polynomial in the size of q. For each equivalence class ζ of $\sim_q$, choose a representative $t_\zeta \in \zeta$ and if $\mathsf{pre}(\zeta) \neq \emptyset$, choose a $t_\zeta^{\mathsf{pre}} \in \mathsf{pre}(\zeta)$. For $q = \exists \vec{u}.\psi$, the rewritten query $q^*_{\mathcal{R}}$ is now defined as $\exists \vec{u}.(\psi \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3)$, where φ_1, φ_2, and φ_3 are as follows:

$$\varphi_1 \quad := \quad \bigwedge_{v \in \mathsf{avar}(q) \cup \mathsf{Fork}_{\neq} \cup \mathsf{Cyc}} \neg \mathsf{Aux}(v)$$

$$\varphi_2 \quad := \quad \bigwedge_{(\{t_1, \ldots, t_k\}, \zeta) \in \mathsf{Fork}_{=}} (\, \mathsf{Aux}(t_\zeta) \rightarrow \bigwedge_{1 \leq i < k} t_i = t_{i+1} \,)$$

$$\varphi_3 \quad := \quad \bigwedge_{(I, \zeta) \in \mathsf{Fork}_{\mathcal{H}}} (\, \mathsf{Aux}(t_\zeta) \rightarrow \bigvee_{r \in I} r(t_\zeta^{\mathsf{pre}}, t_\zeta) \,)$$

We can show that $q^*_{\mathcal{R}}$ is as required [Lutz *et al.*, 2009]. The following examples illustrate the definition of $q^*_{\mathcal{R}}$. Since φ_3 is simply true when $\mathcal{R} = \emptyset$, we omit it in this case.

(1) Let $\mathcal{R} = \emptyset$ and consider $q = \exists u.\psi$ with $\psi = r(v, u) \wedge r(v', u)$. This query illustrates the role of $\mathsf{Fork}_{=}$. $\sim_q$ consists of the equivalence classes $\{v, v'\}$ and $\{u\}$. We have $\mathsf{pre}(\{u\}) = \{v, v'\}$ and $\mathsf{in}(\{u\}) = \{r\}$. Hence $\mathsf{Fork}_{=} = \{(\{v, v'\}, \{u\})\}$ and $\mathsf{Fork}_{\neq} = \mathsf{Fork}_{\mathcal{H}} = \mathsf{Cyc} = \emptyset$. We obtain

$$q^*_{\mathcal{R}} = \exists u.(\psi \wedge \neg \mathsf{Aux}(v) \wedge \neg \mathsf{Aux}(v') \wedge (\mathsf{Aux}(u) \rightarrow v = v')).$$

(2) Let $\mathcal{R} = \emptyset$ and consider $q = \exists u.(r(v, u) \wedge s(u, u))$. This query illustrates the role of Cyc. We have $u \in \mathsf{Cyc}$ and so

$$q^*_{\mathcal{R}} = \exists u.(r(v, u) \wedge s(u, u) \wedge \neg \mathsf{Aux}(v) \wedge \neg \mathsf{Aux}(u)).$$

(3) Let $\mathcal{R} = \emptyset$ and consider $q = \exists u.(r(v, u) \wedge s(v, u))$ from Example 3. This query illustrates the role of $\mathsf{Fork}_{\neq}$. We have $\mathsf{in}(\{u\}) = \{r, s\}$. There is no implicant of $\mathsf{in}(\{u\})$ in $\mathsf{in}(\{u\})$ and thus $u \in \mathsf{Fork}_{\neq}$. We obtain

$$q^*_{\mathcal{R}} = \exists u.(r(v, u) \wedge s(v, u) \wedge \neg \mathsf{Aux}(v) \wedge \neg \mathsf{Aux}(u)).$$

For $\mathcal{R} = \{s \sqsubseteq r\}$ and the same query q, s is an implicant of $\mathsf{in}(\{u\})$ in $\mathsf{in}(\{u\})$. Thus $u \notin \mathsf{Fork}_{\neq}$. Observe that $\mathsf{Fork}_{\mathcal{H}} = \emptyset$ as the prime implicant s of $\mathsf{in}(\{u\})$ is contained in $\mathsf{in}(\{u\})$. We obtain

$$q^*_{\mathcal{R}} = \exists u.(r(v, u) \wedge s(v, u) \wedge \neg \mathsf{Aux}(v)).$$

Finally, assume $\mathcal{R} = \{r_0 \sqsubseteq r, r_0 \sqsubseteq s\}$. Again $u \notin \mathsf{Fork}_{\neq}$, but now the prime implicant r_0 of $\mathsf{in}(\{u\})$ is not contained in $\mathsf{in}(\{u\})$. Thus, $\mathsf{Fork}_{\mathcal{H}} = \{(\{r_0\}, \{u\})\}$ and we obtain

$$q^*_{\mathcal{R}} = \exists u.(r(v, u) \wedge s(v, u) \wedge \neg \mathsf{Aux}(v) \wedge (\mathsf{Aux}(u) \rightarrow r_0(v, u))).$$

(4) For queries $q_C = \exists \vec{v}.\psi$ that result from the unfolding of an $\mathcal{EL}$-concept C or have no quantified variables almost no query rewriting is needed: in both cases we have

$$q^*_{\mathcal{R}} = \exists \vec{v}.(\psi \wedge \textstyle\bigwedge_{v \in \mathsf{avar}(q)} \neg \mathsf{Aux}(v)).$$

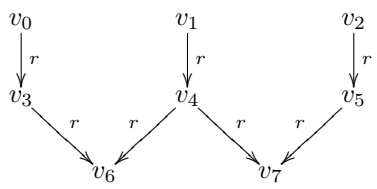

Figure 2: Query for Example (5).

(5) Let $\mathcal{R} = \emptyset$ and let $q = \exists v_0, \ldots, v_7.\psi$ be the query shown in Figure 2, where all variables are quantified. Then $\sim_q$ consists of the equivalence classes $\{v_0, v_1, v_2\}$, $\{v_3, v_4, v_5\}$, $\{v_6\}$, and $\{v_7\}$. Assume that the chosen representative for $\{v_3, v_4, v_5\}$ is v_3. Then, we have

$$\begin{aligned} q^*_{\mathcal{R}} = \exists v_0, \ldots, v_7.(\; & \psi \wedge \\ & \mathsf{Aux}(v_6) \rightarrow (v_3 = v_4) \wedge \\ & \mathsf{Aux}(v_7) \rightarrow (v_4 = v_5) \wedge \\ & \mathsf{Aux}(v_3) \rightarrow ((v_0 = v_1) \wedge (v_1 = v_2)) \end{aligned}$$

In the following, we comment on certain relevant properties of $q^*_{\mathcal{R}}$ and on consequences of our approach regarding the computational complexity of query answering in $\mathcal{ELH}^{dr}_{\bot}$. Firstly, the size of $q^*_{\mathcal{R}}$ is bounded by $\mathcal{O}(nm)$, where n is the size of q and m is $\mathsf{max}\{1, |\mathcal{R}|\}$. To see this, note that the number of conjuncts in of φ_1 is bounded by the number of variables in q. For φ_2, let $\mathsf{Fork}_{=} = \{(T_1, \zeta_1), \ldots, (T_\ell, \zeta_\ell)\}$. Then $|T_1| + \cdots + |T_\ell|$ is bounded by the number of role atoms in q, and thus the size of φ_2 is $\mathcal{O}(n)$. Finally, the number of conjuncts in φ_3 is bounded by the number of quantified variables in q and each conjunct has at most m disjuncts. Note that $q^*_{\mathcal{R}}$ is thus of size $\mathcal{O}(n)$ when $\mathcal{R} = \emptyset$.

Secondly, since ψ is a conjunct of the body of $q^*_{\mathcal{R}}$ it is readily checked that $q^*_{\mathcal{R}}$ is domain independent, and thus can be expressed as an SQL query. Moreover it is of the form $\exists \vec{u}.\psi$ with ψ quantifier-free. Thus, we obtain that the combined complexity of deciding whether $\mathcal{K} \models q[a_1, \ldots, a_k]$ is in NP: construct (in poly-time) $\mathcal{I}^r_{\mathcal{K}}$ and $q^*_{\mathcal{R}} = \exists \vec{u}.\psi$, then check whether $\mathcal{I}^r_{\mathcal{K}} \models \exists \vec{u}.\psi[a_1, \ldots, a_k]$ using the obvious NP algorithm for model checking this class of formulas. Since an NP lower bound is trivial, we obtain NP-completeness. See [Krötzsch *et al.*, 2007; Rosati, 2007] for similar results regarding other variants of $\mathcal{EL}$.

5 Implementation and Experiments

To validate the value of the proposed approach, we have conducted a series of experiments based on the NCI thesaurus (version 08.08d), which is a well-known ontology from the bio-medical domain [Sioutos *et al.*, 2006]. We have extracted an $\mathcal{EL}$-TBox that contains approximately 65 thousand (65K) primitive concept names, 70 primitive roles, and over 70K concept inclusions and concept definitions. The auxiliary part of canonical models, which is independent of the ABox, consists of 702K concept assertions and 171K role assertions (tuples). For the experiments reported below, we used the IBM DB2 DBMS (version 9.5.0 running on SUN Fire-280R server with two UltraSPARC III 1.2GHz CPUs, 4GB memory, and 1TB storage under Solaris 5.10).

For rewriting ABoxes into canonical models, we have used a rule-based approach implemented via iterated querying; see [Lutz *et al.*, 2008] for more details. As relational systems are not optimized to handle tens of thousands of relatively small relations, one for each concept and role name in the ontology, we have used only two relations to represent $\mathcal{I}^r_{\mathcal{K}}$:

`acbox(conceptid,indid)` and
`arbox(roleid,domain-indid,range-indid)`,

where `conceptid` and `roleid` are numerical identifiers for concept names and roles names, `indid`, `domain-indid`, and `range-indid` are numerical identifiers for individuals from $\mathsf{N_I} \cup \mathsf{N_{I_{aux}}}$, `acbox` represents concept memberships, and `arbox` role memberships. Indexes were generated on the attributes `conceptid,indid` and on `roleid,domain-indid` and `roleid,range-indid`, using B+trees. We distinguish individuals from $\mathsf{N_I}$ and $\mathsf{N_{I_{aux}}}$ by positive and negative identifiers, and thus need not store Aux as a relation. As an example for this representation, take the query $q = \mathit{Nerve}(x) \wedge \neg \mathsf{Aux}(x)$, which translates into the SQL statement

```
select indid from  acbox
where  conceptid=141723 and indid > 0
```

Data sets (ABox instances) for our experiments were generated randomly. When generating concept assertions, we have focused on most specific concept names, i.e., concept names without any subsumees in the TBox. The generation of role assertions was guided by the domain and range restrictions in NCI. The numbers of concept and role assertions in the initial and rewritten ABoxes used in our experiments are reported in Figure 3 in thousands/millions. Due to implementation particularities of existing DB systems, ABox rewriting took up to several hours. We point out that (i) this high time consumption is not inherent to our approach and (ii) ABox rewriting can be implemented as an offline task in typical applications such as online analytical processing (OLAP) and data warehousing.

Figure 3 summarizes the running times (in seconds) for each of the test queries for varying sizes of data; for each query we list the number of concept and role atoms in parentheses; the structure of the queries ranges from simple chains (Q1) and star queries (Q2,Q3,Q4) to queries with cycles in their bodies (Q5). We show only a few representative samples here.

The experimental results can be interpreted as follows. Firstly, the rewriting of moderately-sized ABoxes into the canonical model $\mathcal{I}^r_{\mathcal{K}}$ is well within the storage and query capabilities of existing relational technology. Secondly, query performance scales well even when the naive physical design described above is used. Thirdly, query rewriting does not increase the query processing times; the performance of rewritten queries is almost identical to the performance of the original queries over the completed ABox.

6 Limitations / Negation

Recall from Section 1 that a DL enjoys *polynomial* combined FO rewritability if it has combined FO rewritability such that the blowup of ABox rewriting (but not necessarily of query rewriting) is at most polynomial. Since ABoxes in realistic applications are large, combined FO rewritability that is not polynomial in this sense does not seem to be of much use. The following result gives a fundamental limitation of polynomial combined FO rewritability. Note that *ground* CQ answering, where a CQ may contain only individual names but not variables, is the decisional variant of CQ answering.

Number of assertions in ABox of $\mathcal{K}$									
Concept	100K	100K	100K	200K	200K	200K	400K	800K	1.6M
Role	25K	50K	75K	40K	65K	90K	360K	1.5M	5.8M
Number of assertions in $\mathcal{I}^r_{\mathcal{K}}$									
Concept	440K	440K	441K	683K	683K	684K	1.3M	2.6M	5.1M
Role	197K	237K	273K	323K	371K	414K	986K	2.7M	8.2M
Query Execution Time in seconds									
Q1 (2c1r)	0.19	0.19	0.20	0.23	0.25	0.24	0.27	0.46	0.59
Q2 (3c2r)	0.23	0.22	0.23	0.52	0.25	0.56	0.33	0.42	0.69
Q3 (3c2r)	0.25	0.27	0.26	0.31	0.31	0.31	0.42	0.86	1.13
Q4 (4c3r)	0.24	0.23	0.23	0.25	0.26	0.25	0.31	0.42	1.44
Q5 (5c5r)	0.36	0.36	0.30	0.60	0.34	0.45	2.24	7.93	128[1]

Figure 3: Summary of Experimental Results.

Theorem 6. *If the data complexity of ground CQ answering in a DL $\mathcal{L}$ is not in* PTIME*, then $\mathcal{L}$ does not enjoy polynomial combined FO rewritability.*

Proof. We show the contrapositive. Assume that $\mathcal{L}$ enjoys polynomial combined FO rewritability, i.e., there are effectively computable mappings δ that takes each $\mathcal{L}$-TBox $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ to a first-order structure $\delta(\mathcal{K})$ and γ that takes each pair $(q, \mathcal{T})$, with q a k-ary conjunctive query and $\mathcal{T}$ an $\mathcal{L}$-TBox, to a first-order formula $\gamma(q, \mathcal{T})$ with k free variables such that δ can be computed in polynomial time, the size of $\delta(\mathcal{K})$ is polynomial in the size of $\mathcal{K}$ for all $\mathcal{K}$, and the following condition holds: for all combined $\mathcal{L}$-KBs $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, all k-ary conjunctive queries q, and all tuples $(a_1, \ldots, a_k) \in \mathsf{N_I}$, $\mathcal{K} \models q[a_1, \ldots, a_k]$ iff $\delta(\mathcal{K}) \models \gamma(q, \mathcal{T})[a_1, \ldots, a_k]$.

Then the data complexity of ground CQ answering in $\mathcal{L}$ is in PTIME: to check whether $\mathcal{K} \models q$ with q ground, we can compute $\delta(\mathcal{K})$ in polynomial time and $\gamma(q, \mathcal{T})$ in constant time (as the size of q and $\mathcal{T}$ is constant), and then use FO model checking to decide whether $\delta(\mathcal{K}) \models \gamma(q, \mathcal{T})$. The latter can be done in LOGSPACE. ❑

For expressive DLs such as $\mathcal{ALC}$ and $\mathcal{SHIQ}$, the data complexity of ground CQ answering is co-NP-hard, thus Theorem 6 implies that these DLs do not enjoy polynomial combined FO rewritability unless PTIME = co-NP.

In many applications, it is natural to admit also negated concept assertions in the ABox and to extend conjunctive queries with negated concept atoms in order to query the resulting ABoxes, see e.g. [Patel *et al.*, 2007]. Let us call ABoxes, queries, and KBs of this form *literal*. The result stated in Theorem 6 also applies to literal KBs and literal queries. Since the data complexity of literal CQ answering over literal $\mathcal{EL}$-KBs (where $\mathcal{EL}$ is $\mathcal{ELH}^{dr}_{\bot}$ without $\bot$, role hierarchies, and domain and range restrictions) is co-NP-hard [Schaerf, 1993], it follows that even this mild extension of $\mathcal{EL}$ does not enjoy polynomial combined FO rewritability.

[1] This time is solely due to a large size of the result (60M tuples).

However, there is still a (pragmatic, yet formal) way to add negation to our approach. In the following, we sketch an incomplete approach to literal CQ answering over literal $\mathcal{ELH}_\bot^{dr}$-KBs. Its incompleteness can be precisely characterized in terms of an epistemic semantics for negated concept atoms in literal CQs, inspired by [Calvanese *et al.*, 2007a].

As a preliminary, we assume *standard names*, i.e., interpretation domains $\Delta^\mathcal{I}$ have to be a subset of the (countably infinite) set $\mathsf{N_I}$ of individual names and $a^\mathcal{I} = a$ for all $\mathcal{I}$ and all $a \in \mathsf{N_I}$. For a literal $\mathcal{ELH}_\bot^{dr}$-KB $\mathcal{K}$, an interpretation $\mathcal{I}$, and a literal CQ $q = \exists \vec{u}.\psi(\vec{u}, \vec{v})$ with $\vec{v} = v_1, \ldots, v_k$, we set $\mathcal{I} \models_e q[a_1, \ldots, a_k]$ iff there exists a variable assignment π with $\pi(v_i) = a_i$ for $1 \leq i \leq k$ and

- $\mathcal{I} \models_\pi \alpha$ for all positive atoms α in ψ,
- $\mathcal{K} \models \neg A(a)$ for all $\neg A(a)$ in ψ with $a \in \mathsf{N_I}$, and
- $\mathcal{K} \models \neg A(\pi(v))$ for all $\neg A(v)$ in ψ with $v \in \mathsf{N_V}$.

Now set $\mathcal{K} \models_e q[a_1, \ldots, a_k]$ iff $\mathcal{I} \models_e q[a_1, \ldots, a_k]$ for all models $\mathcal{I}$ of $\mathcal{K}$ with standard names. Thus, answers to negated atoms do not depend on a concrete interpretation $\mathcal{I}$, but only on deducibility. Epistemic semantics is sound: every answer is also an answer under the standard semantics (but not vice versa); it is also conservative: it yields the same answers as standard semantics when either the KB or the query does not contain negation. An example for which the epistemic semantics is different from the standard semantics is given by $\mathcal{A} = \{A'(a), \neg A(a)\}$, $\mathcal{T} = \{A' \sqsubseteq \exists r.\top, \exists r.B \sqsubseteq A\}$, and $q = \exists u.r(a, u) \wedge \neg B(u)$. Then $\mathcal{K} \models q$ but $\mathcal{K} \not\models_e q$.

We give a simple (and poly-time) reduction of epistemic literal CQ answering over literal $\mathcal{ELH}_\bot^{dr}$-KBs to standard CQ answering over $\mathcal{ELH}_\bot^{dr}$-KBs. Via the rewritings presented in the main part of this paper, the reduction enables the use of RDBMSs also for the case of $\mathcal{ELH}_\bot^{dr}$ with negation (under the epistemic semantics).

Given a consistent literal $\mathcal{ELH}_\bot^{dr}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a literal query q, replace every literal $\neg A$ in q with a fresh concept name $\overline{A}$. Additionally, add $\top \sqsubseteq \overline{A}$ to $\mathcal{T}$ whenever $\mathcal{T} \models A \sqsubseteq \bot$, and $\overline{A}(a)$ to $\mathcal{A}$ for all $a \in \mathsf{Ind}(\mathcal{A})$ with $\mathcal{K} \models \neg A(a)$, and then remove all negated assertions from $\mathcal{A}$. Call the result $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ and q'.

Theorem 7. *Let $\mathcal{K}$ be a consistent literal $\mathcal{ELH}_\bot^{dr}$-KB. Then $\mathcal{K}'$ can be computed in polynomial time and $\mathcal{K} \models_e q[a_1, \ldots, a_k]$ iff $\mathcal{K}' \models q'[a_1, \ldots, a_k]$ for all literal CQs q and all $a_1, \ldots, a_k \in \mathsf{Ind}(\mathcal{A})$.*

7 Conclusion

We have proposed a novel approach to CQ answering in DLs using RDBMSs. Unlike previous approaches, it can be used also for DLs for which the data complexity is between LOGSPACE and PTIME. In particular, this includes DLs that fully admit existential restrictions (both on the left- and right-hand side of concept inclusions; see [Calvanese *et al.*, 2006]) and paves the way to using RDBMSs for CQ answering in the $\mathcal{EL}$ family of DLs. Our experiments exhibit a promising performance even without a sophisticated physical design. One drawback of our approach is the blowup of the data, which is polynomial but still considerable on large data sets. As future work, it might be interesting to reduce this blowup by incorporating the TBox partly into the data and partly into the query. We will also develop effective approaches to update the canonical model/auxiliary data when assertions are added to or deleted from the ABox.

References

[Baader *et al.*, 2005b] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI05*, pages 364–369. Professional Book Center, 2005.

[Baader *et al.*, 2008] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In *In Proc. of OWLED08*, 2008.

[Calvanese *et al.*, 2006] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. *Proc. of KR06*, pages 260–270, 2006.

[Calvanese *et al.*, 2007a] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. EQL-Lite: Effective first-order query processing in DLs. In *Proc. of IJCAI07*, pages 274–279. AAAI press, 2007.

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

[Krisnadhi and Lutz, 2007] A. Krisnadhi and C. Lutz. Data complexity in the $\mathcal{EL}$ family of DLs. In *Proc. of LPAR07*, volume 4790 of *LNCS*, pages 333–347. Springer, 2007.

[Krötzsch *et al.*, 2007] M. Krötzsch, S. Rudolph, and P. Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proc. of ISWC07*, volume 4825 of *LNCS*, pages 310–323. Springer, 2007.

[Lutz and Wolter, 2007] C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proc. of CADE21*, volume 4603 of *LNAI*, pages 84–99. Springer, 2007.

[Lutz *et al.*, 2008] C. Lutz, D. Toman, and F. Wolter Conjunctive query answering in $\mathcal{EL}$ using a database system. In *Proc. of OWLED08*, 2008.

[Lutz *et al.*, 2009] C. Lutz, D. Toman, and F. Wolter Conjunctive query answering in the DL $\mathcal{EL}$ using an RDBMS Technical Report, 2009. http://www.informatik.uni-bremen.de/∼clu/papers/

[Patel *et al.*, 2007] C. Patel, J. J. Cimino, J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, E. Schonberg, and K. Srinivas. Matching patient records to clinical trials using ontologies. In *Proc. of ISWC07*, volume 4825 of *LNCS*, pages 816–829. Springer, 2007.

[Rosati, 2007] Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of DL07*, volume 250 of *CEUR-WS*, 2007.

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

[Sioutos *et al.*, 2006] N. Sioutos, S. de Coronado, M.W. Haber, F.W. Hartel, W.L. Shaiu, and L.W. Wright. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *J. of Biomedical Informatics*, 40(1):30–43, 2006.

[Wu *et al.*, 2008] Z. Wu, G. Eadon, S. Das, E. Chong, V. Kolovski, M. Annamalai, and J. Srinivasan. Implementing an inference engine for RDFS/OWL constructs and user-defined rules in oracle. In *Proc. of ICDE08*, pages 1239–1248. IEEE, 2008.