Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Ontology-Based Data Access with Closed Predicates Is Inherently Intractable (Sometimes)

Carsten Lutz and **İnanç Seylan**
Fachbereich Informatik
Universität Bremen, Germany

Frank Wolter
Department of Computer Science
University of Liverpool, UK

Abstract

When answering queries in the presence of ontologies, adopting the closed world assumption for some predicates easily results in intractability. We analyze this situation on the level of individual ontologies formulated in the description logics DL-Lite and $\mathcal{EL}$ and show that in all cases where answering conjunctive queries (CQs) with (open and) closed predicates is tractable, it coincides with answering CQs with all predicates assumed open. In this sense, CQ answering with closed predicates is inherently intractable. Our analysis also yields a dichotomy between AC^0 and coNP for CQ answering w.r.t. ontologies formulated in DL-Lite and a dichotomy between PTIME and coNP for $\mathcal{EL}$. Interestingly, the situation is less dramatic in the more expressive description logic $\mathcal{ELI}$, where we find ontologies for which CQ answering is in PTIME, but does not coincide with CQ answering where all predicates are open.

1 Introduction

Description logics (DLs) increasingly find application in ontology-based data access (OBDA), where an ontology is used to enrich instance data and a chief problem is to provide efficient query answering services. In this context, it is common to make the open world assumption (OWA). Indeed, there are applications where the data is inherently incomplete and the OWA is semantically adequate, such as when the data is extracted from the web. In other applications, however, it is more reasonable to make a closed world assumption (CWA) for some predicates in the data. In particular, when the data is taken from a relational database, then the CWA can be appropriate for some of the data predicates. As a concrete example, consider a geographical database such as OpenStreetMap which contains pure geographical data as well as rich annotations, stating for example that a certain spatial area is the location of a 'popular Thai restaurant'. As argued in [Hübner *et al.*, 2004; Codescu *et al.*, 2011], it is useful to pursue an OBDA approach to take full advantage of the annotations, where one would naturally interpret the geographical data as closed and the annotations as open.

In the DL literature, there are a variety of approaches to imposing a partial CWA, often based on epistemic operators or rules [Calvanese *et al.*, 2007b; Donini *et al.*, 2002; Grimm and Motik, 2005; Motik and Rosati, 2010; Sengupta *et al.*, 2011]. In this paper, we adopt the standard semantics from relational databases, which is both natural and straightforward: CWA predicates have to be interpreted exactly as described in the data, making the standard names assumption for data constants; for example, when A is a closed concept name and $\mathcal{A}$ an ABox, then in any model $\mathcal{I}$ of $\mathcal{A}$ we must have $A^\mathcal{I} = \{a \mid A(a) \in \mathcal{A}\}$. Note that this semantics is also used in the recently proposed DBoxes [Seylan *et al.*, 2009]. In fact, the setup considered in this paper generalizes both standard OBDA (only open predicates permitted) and DBoxes (only closed predicates permitted in data) by allowing to freely mix open and closed predicates both in the ontology and in the data.

A major problem in admitting closed predicates in OBDA is that query answering easily becomes intractable regarding data complexity, where the ontology and query are assumed to be fixed and thus of constant size. In fact, answering conjunctive queries (CQs) is coNP-hard already when ontologies are formulated in inexpressive DLs such as DL-Lite and $\mathcal{EL}$ [Franconi *et al.*, 2011]. While this is an interesting first step, it was recently demonstrated in [Lutz and Wolter, 2012] in the context of OBDA with more expressive DLs that a 'non-uniform' analysis of data complexity, which considers individual ontologies instead of entire logics, can reveal a much more detailed and subtle picture. In our context, we work with ontologies of the form $(\mathcal{T}, \Sigma)$, where $\mathcal{T}$ is a DL TBox and Σ a set of predicates (concept and role names) declared to be closed. We say that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ *is in* PTIME if for every CQ $q(\vec{x})$, there exists a polytime algorithm that computes for a given ABox $\mathcal{A}$ the certain answers to q in $\mathcal{A}$ given $(\mathcal{T}, \Sigma)$; *CQ answering w.r.t.* $(\mathcal{T}, \Sigma)$ *is* coNP-*hard* if there is a Boolean CQ q such that, given an ABox $\mathcal{A}$, it is coNP-hard to decide whether q is entailed by $\mathcal{A}$ given $\mathcal{T}$. Other complexity classes are defined analogously.

The aim of this paper is to carry out a non-uniform analysis of the data complexity of query answering with closed predicates, when TBoxes are formulated in the DLs DL-Lite$_\mathcal{R}$ and $\mathcal{EL}$, underpinning the OWL 2 profiles OWL 2 QL and OWL 2 EL, respectively [Calvanese *et al.*, 2007a; Artale *et al.*, 2009; Baader *et al.*, 2005]. Our main results are (i) characterizations that separate the tractable cases from the intractable ones and map out the frontier of tractability in a transparent way; (ii) a proof that, for every tractable case $(\mathcal{T}, \Sigma)$, CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ coincides with CQ answering w.r.t. the ontology $(\mathcal{T}, \emptyset)$ that treats all predicates as open, for ABoxes that are satisfiable w.r.t. $(\mathcal{T}, \Sigma)$; (iii) a dichotomy for the data complexity of CQ answering between AC^0 and CONP for TBoxes formulated in DL-Lite$_\mathcal{R}$, and between PTIME and CONP for $\mathcal{EL}$; and (iv) algorithms for deciding in PTIME whether a given $(\mathcal{T}, \Sigma)$ admits tractable CQ-answering or not.

Point (ii) can be interpreted as showing that OBDA with closed predicates is inherently intractable since, in all tractable cases, the declaration of closed predicates does not have an impact on query answers (it only results in imposing integrity constraints on the ABox). This rather negative result is relativized by the observation made at the end of the paper that inherent intractability does not transfer to more expressive description logics such as $\mathcal{ELI}$, which is essentially the union of DL-Lite and $\mathcal{EL}$: there are TBoxes with closed predicates $(\mathcal{T}, \Sigma)$ with $\mathcal{T}$ formulated in $\mathcal{ELI}$ such that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is tractable, but does *not* coincide with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$. Further exploring this encouraging observation is left as future work. We also note that, even in cases where no additional answers to CQs are obtained, there is a certain benefit of closed predicates: we can go beyond CQs and admit full first-order expressive power for the 'closed part' of queries without increasing the data complexity. We propose a concrete query language that implements this idea and show that AC^0 data complexity is preserved for DL-Lite$_\mathcal{R}$ TBoxes and PTIME data complexity is preserved for $\mathcal{EL}$-TBoxes.

Point (iii) is interesting when contrasted with CQ answering w.r.t. TBoxes that are formulated in the expressive DLs $\mathcal{ALC}$ and $\mathcal{ALCI}$, without closed predicates. There, the data complexity is also between AC^0 and CONP, but the existence of a dichotomy between PTIME and CONP is a deep open question that is equivalent to the Feder-Vardi conjecture for the existence of a dichotomy between PTIME and NP in constraint satisfaction problems [Lutz and Wolter, 2012]. In this sense, the space of ontologies $(\mathcal{T}, \Sigma)$ studied in this paper is more well-behaved than the space of all $\mathcal{ALC}$-ontologies.

Some proof details are deferred to the appendix of the long version, http://cgi.csc.liv.ac.uk/~frank/publ/publ.html.

2 Preliminaries

We use standard notation from description logic [Baader *et al.*, 2003]. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite sets of *concept* and *role names*. A *DL-Lite concept* is either a concept name from $\mathsf{N_C}$ or a concept of the form $\exists r.\top$ or $\exists r^-.\top$, where $r \in \mathsf{N_R}$. We call r^- an *inverse role* and set $s^- = r$ if $s = r^-$ and $r \in \mathsf{N_R}$. A *role* is of the form r or r^-, with $r \in \mathsf{N_R}$. A *DL-Lite concept inclusion* is an expression of the form $B_1 \sqsubseteq B_2$ or $B_1 \sqsubseteq \neg B_2$, where B_1, B_2 are DL-Lite concepts. A *role inclusion* is an expression of the form $r \sqsubseteq s$, where r, s are roles. A *DL-Lite$_{core}$ TBox* is a finite set of DL-Lite concept inclusions and a DL-Lite$_\mathcal{R}$ TBox is a finite set of DL-Lite concept inclusions and role inclusions.

$\mathcal{EL}$ *concepts* are constructed according to the rule $C, D := \top \mid A \mid C \sqcap D \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. An $\mathcal{EL}$ *concept inclusion* is an expression of the form $C \sqsubseteq D$, where C, D are $\mathcal{EL}$ concepts. An $\mathcal{EL}$ *TBox* is a finite set of $\mathcal{EL}$ concept inclusions. $\mathcal{ELI}$ is the extension of $\mathcal{EL}$ with existential restrictions $\exists r^-.C$, where r^- is an inverse role.

An *ABox* is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ with $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and a, b individual names from a countably infinite set $\mathsf{N_I}$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individual names used in the ABox $\mathcal{A}$ and take the freedom to write $r^-(a, b) \in \mathcal{A}$ instead of $r(b, a) \in \mathcal{A}$.

An interpretation $\mathcal{I}$ is a pair $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty set called the *domain* of $\mathcal{I}$ and $\cdot^\mathcal{I}$ maps each concept name A to a subset $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and each role name r to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is extended to compound concepts in the usual way. An interpretation $\mathcal{I}$ *satisfies* a concept inclusion $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a role inclusion $r \sqsubseteq s$ if $r^\mathcal{I} \subseteq s^\mathcal{I}$, a concept assertion $A(a)$ if $a \in A^\mathcal{I}$ and a role assertion $r(a, b)$ if $(a, b) \in r^\mathcal{I}$. Note that this interpretation of ABox assertions corresponds to making the standard names assumption (SNA), which stipulates that every ABox individual is interpreted as itself; the SNA implies the unique name assumption (UNA). An interpretation is a *model* of a TBox $\mathcal{T}$ if it satisfies all inclusions in $\mathcal{T}$ and a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. A concept C (ABox $\mathcal{A}$) is *satisfiable w.r.t. a TBox* $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ with $C^\mathcal{I} \neq \emptyset$ (and of $\mathcal{A}$).

A *predicate* is a concept or role name. A *signature* Σ is a finite set of predicates. The signature $\mathsf{sig}(C)$ of a concept C, $\mathsf{sig}(r)$ of a role r, and $\mathsf{sig}(\mathcal{T})$ of a TBox $\mathcal{T}$, is the set of predicates that occur in C, r, and $\mathcal{T}$, respectively.

For being able to declare predicates as closed, we add an additional component to TBoxes. A pair $(\mathcal{T}, \Sigma)$ with $\mathcal{T}$ a TBox and Σ a signature is *a TBox with closed predicates*. For any ABox $\mathcal{A}$, a *model* $\mathcal{I}$ *of* $(\mathcal{T}, \Sigma)$ *and* $\mathcal{A}$ is an interpretation $\mathcal{I}$ with $\mathsf{Ind}(\mathcal{A}) \subseteq \Delta^\mathcal{I}$ that satisfies $\mathcal{T}$ and $\mathcal{A}$ and such that the extension of all closed predicates agrees with what is explicitly stated in the ABox, that is,

$$
\begin{aligned}
A^\mathcal{I} &= \{a \mid A(a) \in \mathcal{A}\} && \text{for all } A \in \Sigma \cap \mathsf{N_C} \\
r^\mathcal{I} &= \{(a, b) \mid r(a, b) \in \mathcal{A}\} && \text{for all } r \in \Sigma \cap \mathsf{N_R}.
\end{aligned}
$$

An ABox $\mathcal{A}$ is *satisfiable w.r.t.* $(\mathcal{T}, \Sigma)$ if there is a model of $(\mathcal{T}, \Sigma)$ and $\mathcal{A}$.

Example 2.1. *In a geographical database, complete information is typically available for predicates that are tied closely to geographical location and do not change frequently, such as the concept name* ScandinavianCountry *used to identify regions that describe the spatial extension of a scandinavian country and the role name* neighbor *used to relate regions that describe neighboring countries. These predicates should therefore be treated as closed. For other predicates, especially those that are less intimately linked to geographical location, complete information is often not available. Examples include concept names such as* OilExportingCountry *or roles such as* tradingPartner.

Fix a countably infinite set of variables V. A *first-order query (FOQ)* $q(\vec{x})$ is a first-order formula constructed from atoms $A(x)$, $r(x, y)$, and $x = y$, where x, y range over V and $\vec{x} = x_1, \ldots, x_k$ contains all free variables of q. We call $\vec{x}$

the *answer variables of* $q(\vec{x})$ and say that $q(\vec{x})$ is *Boolean* if it has no answer variables. A *conjunctive query (CQ)* $q(\vec{x})$ is a FOQ using conjunction and existential quantification, only. A tuple $\vec{a} = a_1, \ldots, a_k \subseteq \mathsf{Ind}(\mathcal{A})$ *is a certain answer to* $q(\vec{x})$ *in* $\mathcal{A}$ *given* $(\mathcal{T}, \Sigma)$, in symbols $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q(\vec{a})$, if $\mathcal{I} \models q[a_1, \ldots, a_k]$ for all models $\mathcal{I}$ of $(\mathcal{T}, \Sigma)$ and $\mathcal{A}$. If $\Sigma = \emptyset$, then we simply omit Σ, speak of *certain answers to* $q(\vec{x})$ *in* $\mathcal{A}$ *given* $\mathcal{T}$, and write $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$. A CQ $q(x)$ with one answer variable x is a *directed tree CQ* if it is tree-shaped with root x when viewed as a directed graph and a *tree CQ* if the same is true when q is viewed as an undirected graph.

The following example shows that closing predicates can result in more complete query answers.

Example 2.2. *The TBox* $\mathcal{T}$ *consists of the inclusion*

$$\mathsf{ScandComp} \sqsubseteq \exists \mathsf{based\_in}.\mathsf{ScandCountry}$$

where ScandComp *and* ScandCountry *are short for* ScandinavianCompany *and* ScandinavianCountry. *The ABox* $\mathcal{A}$ *consists of the assertions*

$$\begin{array}{l}\mathsf{ScandComp}(cp), \mathsf{ScandCountry}(denmark),\\ \mathsf{ScandCountry}(norway), \mathsf{ScandCountry}(sweden),\\ \mathsf{TimberExporter}(denmark), \mathsf{TimberExporter}(norway)\\ \mathsf{TimberExporter}(sweden).\end{array}$$

Note that there is no information in $\mathcal{A}$ *about the concrete scandinavian country in which the company cp is based. For*

$$q = \exists y\, \mathsf{based\_in}(x, y) \ \wedge \mathsf{TimberExporter}(y),$$

cp is not a certain answer to $q(x)$ *in* $\mathcal{A}$ *given* $\mathcal{T}$. *In contrast, when closing* ScandCountry *by setting* $\Sigma = \{\mathsf{ScandCountry}\}$, *we have* $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q[cp]$.

As illustrated by Example 2.2, we are interested in reasoning with a mix of closed predicates and open predicates. Note that TBox statements which *only* involve closed predicates act as integrity constraints in the standard database sense [Abiteboul *et al.*, 1995]. As an example, consider $\mathcal{T} = \{A \sqsubseteq B\}$ and $\Sigma = \{A, B\}$. Then $(\mathcal{T}, \Sigma)$ imposes the integrity constraint that if $A(a)$ is contained in an ABox, then so must be $B(a)$. In particular, an ABox $\mathcal{A}$ is satisfiable w.r.t. $(\mathcal{T}, \Sigma)$ iff $\mathcal{A}$ satisfies this integrity constraint. For ABoxes $\mathcal{A}$ that are satisfiable w.r.t. $\mathcal{T}$, $(\mathcal{T}, \Sigma)$ has no further effect on query answers. In a DL context, integrity constraints are discussed in [Calvanese *et al.*, 2007b; Donini *et al.*, 2002; Mehdi *et al.*, 2011; Motik *et al.*, 2009; Motik and Rosati, 2010]. We now fix the relevant notions of complexity, inspired by [Lutz and Wolter, 2012]. When speaking of complexity, we *always* mean data complexity.

Definition 2.3. *For* $(\mathcal{T}, \Sigma)$ *a TBox with closed predicates,*

- CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in PTIME *if for every CQ* $q(\vec{x})$ *there is a polytime algorithm that computes, for a given ABox* $\mathcal{A}$, *all* $\vec{a} \subseteq \mathsf{Ind}(\mathcal{A})$ *with* $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q(\vec{a})$;
- CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is CONP-hard *if there is a Boolean CQ q such that it is* CONP-*hard to decide, given an ABox* $\mathcal{A}$, *whether* $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q$.

For other classes of queries such as tree-shaped CQs, analogous notions can be defined. It is shown in [Franconi *et al.*, 2011] that there are DL-Lite$_{\text{core}}$ TBoxes with closed predicates $(\mathcal{T}, \Sigma)$ such that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is CONP-hard. The proof is easily strengthened to directed tree CQs and adapted to $\mathcal{EL}$. CQ answering w.r.t. both DL-Lite$_{\mathcal{R}}$ and $\mathcal{EL}$-TBoxes is known to be in CONP. Without closed predicates (that is, when $\Sigma = \emptyset$), CQ answering is in PTIME for $\mathcal{EL}$ TBoxes [Calvanese *et al.*, 2007a; Lutz *et al.*, 2009] and in AC^0 for DL-Lite$_{\mathcal{R}}$ TBoxes [Calvanese *et al.*, 2007a; Artale *et al.*, 2009].

The following property plays a central role in our complexity analysis as it turns out to identify the borderline between tractability and CONP-hardness of CQ answering. It is also studied intensively in [Lutz and Wolter, 2012], where convexity is called the ABox disjunction property.

Definition 2.4. *A TBox with closed predicates* $(\mathcal{T}, \Sigma)$ *is* convex *if for all ABoxes* $\mathcal{A}$ *and tree CQs* $q_1(x)$, $q_2(x)$, $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q_1 \vee q_2[a]$ *implies* $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q_i[a]$ *for some* $i \in \{1, 2\}$.

It is well-known that, without closed predicates, every TBox formulated in DL-Lite$_{\mathcal{R}}$ or $\mathcal{EL}$ is convex [Lutz and Wolter, 2012]. On the other hand, it can be shown that the TBox $(\mathcal{T}, \Sigma)$ from Example 2.2 is not convex.

Example 2.5. *Let* $\mathcal{A}'$ *be the extension of the ABox* $\mathcal{A}$ *from Example 2.2 with the assertions* $\mathsf{MilkExporter}(sweden)$, $\mathsf{MilkExporter}(denmark)$, *and* $\mathsf{OilExporter}(norway)$. *Set* $\Sigma = \{\mathsf{ScandCountry}\}$ *and take the tree CQs*

$$\begin{array}{l}q_1(x) = \exists y\, \mathsf{based\_in}(x, y) \wedge \mathsf{MilkExporter}(y)\\ q_2(x) = \exists y\, \mathsf{based\_in}(x, y) \wedge \mathsf{OilExporter}(y).\end{array}$$

Then $\mathcal{T}, \mathcal{A}' \models_{c(\Sigma)} q_1 \vee q_2[cp]$, *but* $\mathcal{T}, \mathcal{A}' \not\models_{c(\Sigma)} q_i[cp]$ *for any* $i \in \{1, 2\}$. *The former is a consequence of the fact that, in any model* $\mathcal{I}$ *of* $(\mathcal{T}, \Sigma)$ *and* $\mathcal{A}'$, *at least one of* $(cp, denmark)$, $(cp, sweden)$, $(cp, norway)$ *must be in* $\mathsf{based\_in}^{\mathcal{I}}$. *To see that* $\mathcal{T}, \mathcal{A}' \not\models_{c(\Sigma)} q_1[cp]$, *note that it is possible to obtain a model of* $(\mathcal{T}, \Sigma)$ *and* $\mathcal{A}'$ *by viewing* $\mathcal{A}'$ *as an interpretation and adding* $(cp, norway)$ *to the extension of* based_in. *For* $\mathcal{T}, \mathcal{A}' \not\models_{c(\Sigma)} q_2[cp]$, *add* $(cp, denmark)$.

We use *tree* CQs in Definition 2.4 as this allows us to derive stronger lower bounds, which refer to this more restricted class of queries. Note that tree CQs are also known as $\mathcal{ELI}$ *instance queries* and directed tree CQs as $\mathcal{EL}$ *instance queries*, both common in OBDA. All our results remain true when tree CQs are replaced with CQs in Definition 2.4.

3 Results for DL-Lite

We start with an example of a DL-Lite$_{\text{core}}$ TBox that is not convex, essentially by recasting Example 2.5, which is based on an $\mathcal{EL}$ TBox, in this language.

Example 3.1. *Let*

$$\begin{array}{rcl}\mathcal{T} &=& \{A \sqsubseteq \exists r.\top, \exists r^-.\top \sqsubseteq B\} \text{ and } \Sigma = \{B\}\\ \mathcal{A} &=& \{A(a), B(b_1), A_1(b_1), B(b_2), A_2(b_2)\}\\ q_i &=& \exists y\, r(x, y) \wedge A_i(y) \text{ for } i \in \{1, 2\}.\end{array}$$

Then $(\mathcal{T}, \Sigma)$ *is not convex because* $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q_1 \vee q_2[a]$, *whereas* $\mathcal{T}, \mathcal{A} \not\models_{c(\Sigma)} q_i[a]$ *for any* $i \in \{1, 2\}$.

The failure of convexity for the TBox $(\mathcal{T}, \Sigma)$ in Example 3.1 results in a choice which can be used to prove that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is coNP-hard. Specifically, the proof is by reduction of 2+2-SAT, a variant of propositional satisfiability where each clause contains precisely two positive literals and two negative literals [Schaerf, 1993]. The queries q_1 and q_2 from the example are used as subqueries of the query constructed in the reduction, where they serve the purpose of distinguishing truth values of propositional variables. The CQ used in the reduction is actually a tree CQ.

It turns out that this proof of CONP-hardness can be adapted to *any* non-convex DL-Lite$_\mathcal{R}$ TBox. Conversely, we will show that convex DL-Lite$_\mathcal{R}$ TBoxes admit CQ answering in AC^0, thus identifying convexity as the borderline between tractability and intractability of CQ answering, and establishing a dichotomy between AC^0 and CONP for CQ answering w.r.t. DL-Lite$_\mathcal{R}$ TBoxes with closed predicates.

Since analyzing DL-Lite$_\mathcal{R}$ TBoxes turns out to be somewhat more technical than analyzing DL-Lite$_{\text{core}}$ TBoxes, we start with the latter as a warmup. The following definition introduces a property of DL-Lite$_{\text{core}}$ TBoxes with closed predicates that we will prove to coincide with convexity, but which is much more concrete.

Definition 3.2 (Safe DL-Lite$_{\text{core}}$ TBox). *A DL-Lite$_{core}$ TBox with closed predicates $(\mathcal{T}, \Sigma)$ is* safe *if there are no DL-Lite concepts B_1, B_2 and role r such that the following conditions are satisfied:*

1. *B_1 is satisfiable w.r.t. $\mathcal{T}$;*
2. *$\mathcal{T} \models B_1 \sqsubseteq \exists r.\top$ and $\mathcal{T} \models \exists r^-.\top \sqsubseteq B_2$;*
3. *$B_1 \neq \exists r.\top$;*
4. *$\mathsf{sig}(B_2) \subseteq \Sigma$ and $\mathsf{sig}(r) \cap \Sigma = \emptyset$.*

Note that Definition 3.2 is essentially a slight generalization of Example 3.1. In particular, the pattern in Point 2 of Definition 3.2 can be found in Example 3.1 (where it is crucial that $r \notin \Sigma$ and $B \in \Sigma$). The following theorem summarizes our results for DL-Lite$_{\text{core}}$.

Theorem 3.3 (Results for DL-Lite$_{\text{core}}$). *Let $(\mathcal{T}, \Sigma)$ be a DL-Lite$_{core}$ TBox with closed predicates. Then*

1. *If $(\mathcal{T}, \Sigma)$ is not safe, then $(\mathcal{T}, \Sigma)$ is not convex and answering tree CQs w.r.t. $(\mathcal{T}, \Sigma)$ is* CONP*-hard.*
2. *If $(\mathcal{T}, \Sigma)$ is safe, then*
   (a) *CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ coincides with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$ for all ABoxes that are satisfiable w.r.t. $(\mathcal{T}, \Sigma)$, and $(\mathcal{T}, \Sigma)$ is convex;*
   (b) *CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in* AC^0.

In a sense, Theorem 3.3 shows that CQ answering in DL-Lite$_{\text{core}}$ with closed predicates is inherently intractable: in all cases where closing predicates results in additional answers to queries (on satisfiable ABoxes), CQ answering is CONP-hard. In all tractable cases, the only effect that closing predicates can thus have is to act as integrity constraints on the ABox (but see Section 5 for another virtue of closing predicates). Note that all TBoxes that refer *only* to closed predicates (thus express only integrity constaints) are safe.

It is also interesting to note that Theorem 3.3 establishes a dichotomy between AC^0 and CONP for CQ answering w.r.t. DL-Lite$_{\text{core}}$ TBoxes with closed predicates, that is, there is no such TBox whose complexity is truely between AC^0 and CONP. As noted in the introduction, this is in stark contrast to results recently established in [Lutz and Wolter, 2012] in the context of more expressive DLs without closed predicates.

To prove Point 1 of Theorem 3.3, one shows that non-safeness implies non-convexity by constructing an appropriate ABox. CONP-hardness can then be proved by reduction from 2+2-SAT, generalizing the CONP-hardness proof for Example 3.1. The proof of Point 2(a) relies on canonical models for DL-Lite$_{\text{core}}$ TBoxes $\mathcal{T}$ without closed predicates. Specifically, for every ABox $\mathcal{A}$ that is satsfiable w.r.t. $\mathcal{T}$, there is a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that for all CQs q and potential answers $\vec{a}$, we have $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$ iff $\mathcal{I} \models q[\vec{a}]$. To establish Point 2(a), it suffices to show that, when $(\mathcal{T}, \Sigma)$ is safe, then $\mathcal{I}$ is also a model of $(\mathcal{T}, \Sigma)$ and $\mathcal{A}$. Consequently and since closing predicates can only result in additional answers, but not in invalidating answers, CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ coincides with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$ and it remains to recall that DL-Lite$_{\text{core}}$ TBoxes without closed predicates are convex. For Point 2(b), it suffices to show that satisfiability of ABoxes w.r.t. $(\mathcal{T}, \Sigma)$ is in AC^0 when $(\mathcal{T}, \Sigma)$ is safe, which is a consequence of the fact that ABox satisfiability and CQ answering in DL-Lite$_{\text{core}}$ without closed predicates are in AC^0. Specifically, we observe that whenever an ABox $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$, then $\mathcal{A}$ is satisfiable w.r.t. $(\mathcal{T}, \Sigma)$ iff $\mathcal{T}, \mathcal{A} \models B(a)$ implies $B(a) \in \mathcal{A}$ for all DL-Lite concepts B with $\mathsf{sig}(B) \subseteq \Sigma$ and $\mathcal{T}, \mathcal{A} \models r(a, b)$ implies $r(a, b) \in \mathcal{A}$ for all role names r from Σ. Proof details for Theorem 3.3 are skipped as we provide them for the strictly stronger DL-Lite$_\mathcal{R}$ version of this theorem, which is given below.

We now extend Definition 3.2 to DL-Lite$_\mathcal{R}$.

Definition 3.4 (Safe DL-Lite$_\mathcal{R}$ TBox). *A DL-Lite$_\mathcal{R}$ TBox with closed predicates $(\mathcal{T}, \Sigma)$ is* safe *if there are no DL-Lite concepts B_1, B_2 and role r such that the following conditions are satisfied:*

1. *B_1 is satisfiable w.r.t. $\mathcal{T}$;*
2. *$\mathcal{T} \models B_1 \sqsubseteq \exists r.\top$ and $\mathcal{T} \models \exists r^-.\top \sqsubseteq B_2$;*
3. *$B_1 \neq \exists r'.\top$ for any role r' such that $\mathcal{T} \models r' \sqsubseteq r$;*
4. *$\mathsf{sig}(B_2) \subseteq \Sigma$ and $\mathsf{sig}(r') \cap \Sigma = \emptyset$ for any role r' such that $\mathcal{T} \models B_1 \sqsubseteq \exists r'.\top$ and $\mathcal{T} \models r' \sqsubseteq r$.*

Note that the conditions in Definition 3.4 generalize the corresponding ones in Definition 3.2 and in this sense, the addition of role hierarchies does not introduce unexpected ways to cause non-convexity and CONP-hardness.

Example 3.5. *Let $\mathcal{T} = \{A \sqsubseteq \exists r.\top, r \sqsubseteq s\}$ and $\Sigma = \{s\}$. Then $(\mathcal{T}, \Sigma)$ is not safe, which is witnessed by the concepts $B_1 = A$, $B_2 = \exists s^-.\top$, and the role r. Indeed, $(\mathcal{T}, \Sigma)$ is not convex, witnessed for example by the ABox $\{A(a), s(a, b_1), A_1(b_1), s(a, b_2), A_2(b_2)\}$ and the queries $q_i = \exists y.r(x, y) \wedge A_i(y)$ for $i \in \{1, 2\}$.*

Now, Theorem 3.3 generalizes to DL-Lite$_\mathcal{R}$.

Theorem 3.6 (Results for DL-Lite$_\mathcal{R}$). *All statements in Theorem 3.3 are still true if DL-Lite$_{core}$ is replaced with DL-Lite$_\mathcal{R}$.*

The proof strategy for Theorem 3.3 is exactly the one described above for DL-Lite$_{\text{core}}$.

Note that it is easy to check in PTIME whether a given DL-Lite$_\mathcal{R}$ TBox with closed predicates $(\mathcal{T}, \Sigma)$ is safe (consequently: whether CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in AC^0) since it suffices to consider DL-Lite concepts B_1, B_2 and roles r from the signature of $\mathcal{T}$ (of which there are only polynomially many) and subsumption in DL-Lite can be decided in AC^0 [Calvanese *et al.*, 2007a].

4 Results for $\mathcal{EL}$

As illustrated by Example 2.5, the effect that causes non-convexity and thus coNP-hardness of DL-Lite TBoxes with closed predicates can also be observed in $\mathcal{EL}$. In the simplest form, this is shown by the TBox with closed predicates $(\mathcal{T}, \Sigma)$ with $\mathcal{T} = \{A \sqsubseteq \exists r.B\}$ and $\Sigma = \{B\}$, which is not convex. However, in $\mathcal{EL}$ there is an additional (and more subtle) cause for non-tractability. The simplest illustrating example uses exactly the same TBox $\mathcal{T}$, but swaps the Σ-memberships of r and B.

Example 4.1. *Let*

$$\begin{aligned}\mathcal{T} &= \{A \sqsubseteq \exists r.B\} \text{ and } \Sigma = \{r\}\\ \mathcal{A} &= \{A(a), r(a, b_1), A_1(b_1), r(a, b_2), A_2(b_2)\}\\ q_i &= \exists y\, r(x, y) \wedge A_i(y)\end{aligned}$$

Then $(\mathcal{T}, \Sigma)$ is not convex because $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q_1 \vee q_2[a]$, whereas $\mathcal{T}, \mathcal{A} \not\models_{c(\Sigma)} q_i[a]$ for any $i \in \{1, 2\}$.

We now give a definition of safeness of $\mathcal{EL}$-TBoxes with closed predicates that captures both causes of non-convexity and, as in the DL-Lite case, coincides both with convexity and with tractability of CQ-answering. We call a concept E a *top-level conjunct (tlc) of* an $\mathcal{EL}$ concept C if C is of the form $D_1 \sqcap \cdots \sqcap D_n$ and $E = D_i$ for some i.

Definition 4.2 (Safe $\mathcal{EL}$ TBox). *An $\mathcal{EL}$ TBox with closed predicates $(\mathcal{T}, \Sigma)$ is* safe *if there exists no $\mathcal{EL}$ inclusion $C \sqsubseteq \exists r.D$ such that*

1. *$\mathcal{T} \models C \sqsubseteq \exists r.D$;*
2. *there does not exist a tlc $\exists r.C'$ of C with $\mathcal{T} \models C' \sqsubseteq D$;*
3. *one of the following is true:*
   - *(s1) $r \notin \Sigma$ and $\mathsf{sig}(D) \cap \Sigma \neq \emptyset$;*
   - *(s2) $r \in \Sigma$, $\mathsf{sig}(D) \not\subseteq \Sigma$ and there is no Σ-concept E with $\mathcal{T} \models C \sqsubseteq \exists r.E$ and $\mathcal{T} \models E \sqsubseteq D$.*

Conditions 3(s1) and 3(s2) reflect the two causes of non-convexity in $\mathcal{EL}$ with closed predicates. The following example illustrates the requirement in Condition 3(s2) that no "interpolating" Σ-concept E exists.

Example 4.3. *Let $\mathcal{T} = \{A \sqsubseteq \exists r.E, E \sqsubseteq B\}$ and first assume that $\Sigma = \{r\}$. Then the inclusion $A \sqsubseteq \exists r.B$ satisfies Condition 3(s2) and thus $(\mathcal{T}, \Sigma)$ is not safe. Now assume $\Sigma = \{r, E\}$. Then, the inclusion $A \sqsubseteq \exists r.B$ does not violate safeness because E can be used as a 'Σ-interpolant'. Note that the ABox $\mathcal{A}$ from Example 4.1, which we used to refute convexity in a very similar situation, is simply unsatisfiable w.r.t. $(\mathcal{T}, \Sigma)$. Indeed, it can be shown that $(\mathcal{T}, \Sigma)$ is safe.*

The following theorem summarizes our main results for $\mathcal{EL}$.

Theorem 4.4 (Main Results for $\mathcal{EL}$). *Let $(\mathcal{T}, \Sigma)$ be an $\mathcal{EL}$ TBox with closed predicates. Then*

1. *If $(\mathcal{T}, \Sigma)$ is not safe, then $(\mathcal{T}, \Sigma)$ is not convex and answering directed tree CQs w.r.t. $(\mathcal{T}, \Sigma)$ is* coNP*-hard.*
2. *If $(\mathcal{T}, \Sigma)$ is safe, then*
   - *(a) CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ coincides with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$ for all ABoxes that are satisfiable w.r.t. $(\mathcal{T}, \Sigma)$, and $(\mathcal{T}, \Sigma)$ is convex;*
   - *(b) CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in* PTIME.

As mentioned before, directed tree CQs are also called $\mathcal{EL}$ instance queries in the literature. The mention of directed tree CQs in Point 1 of Theorem 4.4 thus implies that our results hold for CQs and $\mathcal{EL}$ instance queries alike.

Point 1 of Theorem 4.4 is proved by showing that non-safeness implies non-convexity, which involves two separate constructions that address Cases (s1) and (s2) from Definition 4.2. The proof of Point 2(a) of Theorem 4.4 is again via canonical models, which have to be defined in a rather careful way to make the proof go through. Establishing Point 2(b) involves showing that satisfiability of ABoxes w.r.t. safe $\mathcal{EL}$ TBoxes with closed predicates can be decided in PTIME.

Whereas it is obvious how to check the safeness of a DL-Lite TBox with closed predicates, this is not the case for $\mathcal{EL}$ TBoxes since Definition 4.2 quantifies over all concepts C, D, and E, of which there are infinitely many. In the following, we show that, nevertheless, safeness of an $\mathcal{EL}$-TBox with closed predicates $(\mathcal{T}, \Sigma)$ can be decided in PTIME. The first step is to convert $\mathcal{T}$ into a TBox $\mathcal{T}^*$ that is normalized in the sense that it satisfies the following properties:

(t1) $\mathcal{T}^*$ contains no CI of the form $C \sqsubseteq D_1 \sqcap D_2$;

(t2) if $C \sqsubseteq \exists r.D \in \mathcal{T}^*$, then there is no tlc $\exists r.C'$ of C with $\mathcal{T}^* \models C' \sqsubseteq D$.

Specifically, $\mathcal{T}^*$ can be produced by exhaustively replacing each CI $C \sqsubseteq D_1 \sqcap D_2$ with the two CIs $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$, and each CI $C \sqcap \exists r.C' \sqsubseteq \exists r.D$ where the TBox entails $C' \sqsubseteq D$ with the CI $C' \sqsubseteq D$. It is easy to see that the conversion takes only polynomial time (since subsumption in $\mathcal{EL}$ can be decided in PTIME) and that $\mathcal{T}^*$ is equivalent to $\mathcal{T}$, thus $\mathcal{T}^*$ is safe iff $\mathcal{T}$ is.

It thus suffices to consider $\mathcal{EL}$ TBoxes $(\mathcal{T}, \Sigma)$ where $\mathcal{T}$ is normalized. In this case, the following, stronger version of safeness is equivalent to the original version.

Definition 4.5. *An $\mathcal{EL}$ TBox with closed predicates $(\mathcal{T}, \Sigma)$ is* strongly safe *if there exists no $\mathcal{EL}$ inclusion $C \sqsubseteq \exists r.D \in \mathcal{T}$ such that one of the following is true:*

(c1) $r \notin \Sigma$ and there is some concept E such that $\mathcal{T} \models D \sqsubseteq E$ and $\mathsf{sig}(E) \cap \Sigma \neq \emptyset$;

(c2) $r \in \Sigma$, $\mathsf{sig}(D) \not\subseteq \Sigma$, and there is no Σ-concept E with $\mathcal{T} \models C \sqsubseteq \exists r.E$ and $\mathcal{T} \models E \sqsubseteq D$.

Note that, in Definition 4.5, the concepts C and D are now restricted to subconcepts of $\mathcal{T}$ (whereas E can still be any concept). The following is proved in the long version.

Lemma 4.6. *If $\mathcal{T}$ satisfies Conditions (t1) and (t2), then $(\mathcal{T}, \Sigma)$ is safe iff it is strongly safe.*

It thus remains to deal with the quantification over the concept E in Conditions (c1) and (c2). In the long version, we show that Condition (c1) can be checked in PTIME by carrying out a reachability test in a suitable canonical model of $\mathcal{T}$, and Condition (c2) can be checked in PTIME by executing a polynomial number of subsumption tests (the correctness of the latter relies on $\mathcal{EL}$ having a certain interpolation property). In summary, we obtain the following result.

Theorem 4.7. *Deciding safeness of $\mathcal{EL}$ TBoxes with closed predicates is* PTIME*-complete.*

5 First-Order Queries over Closed Predicates

As observed in [Reiter, 1992; Calvanese *et al.*, 2007b], closing predicates allows to use more expressive query languages without increasing the complexity of query answering. Indeed, mixing open and closed predicates seems particularly useful when large parts of the data stem from a relational database, as in the geographical database application from Section 2. In such a setup, one would typically like to use full FOQs or, in other words, SQL queries. We consider a query language that combines FOQs for closed predicates with CQs for open predicates. For safe TBoxes with closed predicates, such queries can be answered as efficiently as CQs both in the case of DL-Lite and of $\mathcal{EL}$.

As in the relational database setting, we allow only FOQs that are domain-independent and thus correspond to expressions of relational algebra (and SQL queries), see [Abiteboul *et al.*, 1995].

Definition 5.1 ($\text{CQ}^{\text{FO}(\Sigma)}$ queries). *Let Σ be a signature that declares closed predicates. A* conjunctive query with FO(Σ) plugins (abbreviated $\text{CQ}^{\text{FO}(\Sigma)}$) *is of the form $\exists x_1 \cdots \exists x_n(\varphi_1 \wedge \ldots \wedge \varphi_m)$, where $n \geq 0$, $m \geq 1$, and each φ_i is an atom or a domain-independent FOQ with $\mathsf{sig}(\varphi_i) \subseteq \Sigma$.*

The next theorem shows that, for safe TBoxes with closed predicates, switching from CQs to $\text{CQ}^{\text{FO}(\Sigma)}$s does not increase data complexity. Thus, in addition to enforcing integrity constraints, such TBoxes have the virtue of admitting more expressive queries without an increase in complexity.

Theorem 5.2.

1. *For safe DL-Lite$_\mathcal{R}$ TBoxes with closed predicates $(\mathcal{T}, \Sigma)$, $CQ^{FO(\Sigma)}$-answering w.r.t. $(\mathcal{T}, \Sigma)$ is in* AC^0.
2. *For safe $\mathcal{EL}$ TBoxes with closed predicates $(\mathcal{T}, \Sigma)$, $CQ^{FO(\Sigma)}$-answering w.r.t. $(\mathcal{T}, \Sigma)$ is in* PTIME.

While the proof of Theorem 5.2 is not intricate, we believe that $\text{CQ}^{\text{FO}(\Sigma)}$ can be very useful for applications. Note that the query language EQL-Lite(CQ) from [Calvanese *et al.*, 2007b] can be viewed as a fragment of $\text{CQ}^{\text{FO}(\Sigma)}$ in which *only* closed predicates are admitted.

6 The Case of $\mathcal{ELI}$

We consider TBoxes formulated in $\mathcal{ELI}$, the extension of $\mathcal{EL}$ with inverse roles. $\mathcal{ELI}$ can be regarded as the logical core of expressive Horn DLs such as Horn-$\mathcal{SHIQ}$ [Hustadt *et al.*, 2007; Eiter *et al.*, 2008]. In contrast to the cases of DL-Lite$_\mathcal{R}$ and $\mathcal{EL}$, CQ answering with closed predicates turns out to not be inherently intractable in $\mathcal{ELI}$: there are $\mathcal{ELI}$ TBoxes with closed predicates $(\mathcal{T}, \Sigma)$ such that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in PTIME, but does *not* coincide with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$ for all ABoxes that are satisfiable w.r.t. $(\mathcal{T}, \Sigma)$. We use the $\mathcal{ELI}$-TBox

$$\mathcal{T} = \{\ \top \sqsubseteq \exists r.A \qquad \exists r^-.A \sqsubseteq B \qquad \exists r.(A \sqcap B) \sqsubseteq A\ \}$$

and the signature $\Sigma = \{r, B\}$. It is not hard to see that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ does not coincide with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$. In particular, for

$$\mathcal{A} = \{r(a,a), B(a)\} \quad \text{and} \quad q() = \exists x\, r(x,x) \wedge A(x)$$

one can verify that $\mathcal{T}, \mathcal{A} \not\models q()$, but $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q()$; moreover, it is straightforward to construct a model which shows that $\mathcal{A}$ is satisfiable w.r.t. $(\mathcal{T}, \Sigma)$.

To prove our claim, it thus remains to show that CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ is in PTIME. Let $\mathcal{A}$ be an ABox. The interpretation $\mathcal{I}$ is defined as follows:

(a) start with $\mathcal{A}$ viewed as an interpretation;
(b) add $a \in \mathsf{Ind}(\mathcal{A})$ to $A^\mathcal{I}$ if $r(a,b) \in \mathcal{A}$ implies $B(b) \in \mathcal{A}$;
(c) add $a \in \mathsf{Ind}(\mathcal{A})$ to $A^\mathcal{I}$ when $a \in (\exists r.(A \sqcap B))^\mathcal{I}$, repeat exhaustively.

Clearly, $\mathcal{I}$ can be constructed in polynomial time. The following lemma thus shows that CQ-answering w.r.t. $(\mathcal{T}, \Sigma)$ is in PTIME, and so is satisfiability of ABoxes w.r.t. $(\mathcal{T}, \Sigma)$. Note that $\mathcal{I}$ can be viewed as a canonical model of $(\mathcal{T}, \Sigma)$ and $\mathcal{A}$.

Lemma 6.1.

1. *$\mathcal{A}$ is satisfiable w.r.t. $(\mathcal{T}, \Sigma)$ iff $\mathcal{I}$ is a model of $(\mathcal{T}, \Sigma)$ and $\mathcal{A}$;*
2. *if $\mathcal{A}$ is satisfiable w.r.t. $(\mathcal{T}, \Sigma)$, then for all CQs q and $\vec{a} \subseteq \mathsf{Ind}(\mathcal{A})$, we have $\mathcal{T}, \mathcal{A} \models_{c(\Sigma)} q[\vec{a}]$ iff $\mathcal{I} \models q[\vec{a}]$.*

We have thus shown that $\mathcal{ELI}$ behaves differently from DL-Lite$_\mathcal{R}$ and $\mathcal{EL}$. This raises a number of questions, dicussed in the next section.

7 Future Work

We have observed that, for simple DLs such as DL-Lite$_{\text{core}}$, DL-Lite$_\mathcal{R}$, and $\mathcal{EL}$, CQ answering with closed predicates is inherently intractable, while this is not the case for more expressive DLs such as $\mathcal{ELI}$. It would be interesting to conduct a broader study to fully understand this phenomenon, including additional TBox languages such as other extensions of DL-Lite, versions of Horn-$\mathcal{SHIQ}$, and possibly even members of the Datalog$^\pm$ family of ontology languages [Calì *et al.*, 2012].

Concerning the concrete case of $\mathcal{ELI}$, the observation presented in Section 6 raises the question whether there is a dichotomy between PTIME and CONP for CQ answering w.r.t. $\mathcal{ELI}$-TBoxes with closed predicates, and how the PTIME cases can be characterized. It also asks for a characterization of those $\mathcal{ELI}$-TBoxes with closed predicates $(\mathcal{T}, \Sigma)$ for which CQ answering w.r.t. $(\mathcal{T}, \Sigma)$ coincides with CQ answering w.r.t. $(\mathcal{T}, \emptyset)$. We leave these questions as interesting future work.

Acknowledgments. Carsten Lutz and İnanç Seylan were supported by the DFG SFB/TR 8 "Spatial Cognition".

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuiness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, 2005.

[Calì *et al.*, 2012] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics*, 14:57–83, 2012.

[Calvanese *et al.*, 2007a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2007b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. EQL-Lite: Effective first-order query processing in description logics. In *IJCAI*, 2007.

[Codescu *et al.*, 2011] Mihai Codescu, Gregor Horsinka, Oliver Kutz, Till Mossakowski, and Rafaela Rau. DO-ROAM: Activity-oriented search and navigation with OpenStreetMap. In *GeoSpatial Semantics*, 2011.

[Donini *et al.*, 2002] Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic*, 3(2):177–225, 2002.

[Eiter *et al.*, 2008] Thomas Eiter, Georg Gottlob, Magdalena Ortiz, and Mantas Simkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *JELIA*, 2008.

[Franconi *et al.*, 2011] Enrico Franconi, Yazmin Angélica Ibáñez-García, and Inanç Seylan. Query answering with DBoxes is hard. *Electronic Notes in Theoretical Computer Science*, 278:71–84, 2011.

[Grimm and Motik, 2005] Stephan Grimm and Boris Motik. Closed world reasoning in the semantic web through epistemic operators. In *OWLED*, 2005.

[Hübner *et al.*, 2004] Sebastian Hübner, Rainer Spittel, Ubbo Visser, and Thomas J. Vögele. Ontology-based search for interactive digital maps. *IEEE Intelligent Systems*, 19(3):80–86, 2004.

[Hustadt *et al.*, 2007] Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Reasoning in description logics by a reduction to disjunctive datalog. *Journal of Automated Reasoning*, 39(3):351–384, 2007.

[Lutz and Wolter, 2012] Carsten Lutz and Frank Wolter. Non-uniform data complexity of query answering in description logics. In *KR*, 2012.

[Lutz *et al.*, 2009] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *IJCAI*, 2009.

[Mehdi *et al.*, 2011] Anees Mehdi, Sebastian Rudolph, and Stephan Grimm. Epistemic querying of OWL knowledge bases. In *ESWC*, 2011.

[Motik and Rosati, 2010] Boris Motik and Riccardo Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):1–62, 2010.

[Motik *et al.*, 2009] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between OWL and relational databases. *Journal of Web Semantics*, 7(2):74–89, 2009.

[Reiter, 1992] Raymond Reiter. What should a database know? *Journal of Logic Programming*, 14(1&2):127–153, 1992.

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

[Sengupta *et al.*, 2011] Kunal Sengupta, Adila Alfa Krisnadhi, and Pascal Hitzler. Local closed world semantics: Grounded circumscription for OWL. In *ISWC*, 2011.

[Seylan *et al.*, 2009] Inanç Seylan, Enrico Franconi, and Jos de Bruijn. Effective query rewriting with ontologies over DBoxes. In *IJCAI*, 2009.