Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning

Query Containment in Description Logics Reconsidered

Meghyn Bienvenu
Laboratoire de Recherche en Informatique
CNRS & Université Paris Sud, France

Carsten Lutz
Fachbereich Informatik
Universität Bremen, Germany

Frank Wolter
Department of Computer Science
University of Liverpool, UK

Abstract

While query answering in the presence of description logic (DL) ontologies is a well-studied problem, questions of static analysis such as query containment and query optimization have received less attention. In this paper, we study a rather general version of query containment that, unlike the classical version, cannot be reduced to query answering. First, we allow a restriction to be placed on the vocabulary used in the instance data, which can result in shorter equivalent queries; and second, we allow each query its own ontology rather than assuming a single ontology for both queries, which is crucial in applications to versioning and modularity. We also study global minimization of queries in the presence of DL ontologies, which is more subtle than for classical databases as minimal queries need not be isomorphic.

1 Introduction

In ontology-based data access (OBDA), an ontology is used to improve query answering over instance data in various ways, for example by providing a semantics for the data vocabulary, by enriching the query vocabulary, and by translating between data and query vocabularies when they diverge. In the past decade, this paradigm has received significant attention, with a focus on using description logics (DLs) as the ontology language. In particular, answering conjunctive queries (CQs) and the simpler instance queries (IQs) in DL-based OBDA has been extensively studied, so that today, various algorithmic approaches are known, and the computational complexity is well-understood. On the one hand, for expressive DLs such as $\mathcal{ALC}$ and $\mathcal{SHIQ}$, CQ answering is typically EXPTIME- or 2-EXPTIME-complete for combined complexity and NP-complete for data complexity (Calvanese, De Giacomo, and Lenzerini 1998; Hustadt, Motik, and Sattler 2005; Glimm et al. 2008; Lutz 2008; Ortiz, Rudolph, and Simkus 2011). On the other hand, for so-called lightweight DLs such as DL-Lite and $\mathcal{EL}$, CQ answering is in PTIME for data complexity and can be implemented efficiently using off-the-shelf relational database systems (Calvanese et al. 2007; Lutz, Toman, and Wolter 2009; Pérez-Urbina, Horrocks, and Motik 2009; Kontchakov et al. 2010; Calvanese et al. 2011).

While query answering in DLs has been studied intensively, little attention has been paid to the query containment problem, which consists in deciding, given a DL ontology (TBox) $\mathcal{T}$ and two queries q_1 and q_2 of same arity, whether for every data instance (ABox), the answers to q_1 given $\mathcal{T}$ are a subset of the answers to q_2 given $\mathcal{T}$. This is in contrast to relational databases, where query containment is a crucial and widely studied problem due to the central role it plays in query optimization (Abiteboul, Hull, and Vianu 1995). In particular, Chandra and Merlin observed in a classical paper that minimal CQs are unique up to isomorphism, which also means that the unique minimal CQ for a given CQ q can be produced by the following simple procedure: start with q and repeatedly remove atoms that are redundant in the sense that dropping them preserves equivalence; the order in which atoms are dropped is irrelevant and the only non-trivial part is checking equivalence, implemented as two query containment checks (Chandra and Merlin 1977).

Clearly, query optimization is important also in OBDA. For example, in the combined approach to CQ answering presented in (Lutz, Toman, and Wolter 2009; Kontchakov et al. 2010), the CQ is passed virtually unchanged to a relational database system for execution, and thus prior optimization improves performance. The relative lack of interest in OBDA query containment is somewhat surprising and seems to stem mainly from the fact that, for most query languages including CQs and IQs, the problem can be polynomially reduced to query answering and vice versa; thus, algorithms and complexity results transfer (a notable exception are regular path queries, whose containment problem was recently studied in a DL context in (Calvanese, Ortiz, and Simkus 2011)). The aim of this paper is to reconsider CQ- and IQ-containment in DL-based OBDA by (i) proposing a generalized version of containment that enables novel applications and cannot be polynomially reduced to query answering, (ii) giving algorithms and complexity results for this problem, with a focus on lightweight DLs of the DL-Lite and $\mathcal{EL}$ families, and (iii) showing that while naive Chandra-Merlin-minimization as described above fails in the presence of ontologies, by applying slightly refined strategies one can still achieve strong guarantees for the produced minimal queries.

Regarding (i), we generalize OBDA query containment in two directions. First, we pick up the observation of (Baader et al. 2010) that, when an ontology is used to enrich the query vocabulary with symbols that do not occur in the data, then it is useful to carefully distinguish between the data vocabulary and the ontology/query vocabulary. Specifically, we use the data vocabulary Σ as an additional input to query containment, which is then refined to quantify only over Σ-ABoxes. While this sometimes increases the complexity of containment, it can lead to signficantly smaller queries in query minimization.

The second generalization is to associate a separate ontology $\mathcal{T}_i$ with each query q_i instead of assuming a single ontology $\mathcal{T}$ for both queries q_1 and q_2. This is natural from a traditional database perspective, where the ontology would likely be viewed as a component of the query rather than as an independent object. It also enables applications to *ontology versioning* and *ontology modules*. In the former, a typical scenario is that a new version $\mathcal{T}_{\mathsf{new}}$ of a reference ontology $\mathcal{T}_{\mathsf{ref}}$ has to be adopted in an existing application; for example, $\mathcal{T}_{\mathsf{ref}}$ could be the medical terminology SNOMED CT or the National Cancer Institute Ontology NCI (IHTSDO 2008; Sioutos et al. 2006), both widely used and frequently updated. To verify that the update does not affect the application, the user wants to check, for each relevant query q, whether q given $\mathcal{T}_{\mathsf{new}}$ is equivalent to q given $\mathcal{T}_{\mathsf{ref}}$ (see Section 2 for more details). Applications to ontology modules are in a similar spirit: assume that a large ontology $\mathcal{T}$ is replaced with a smaller module $\mathcal{T}' \subseteq \mathcal{T}$ to speed up query processing. When $\mathcal{T}'$ was generated manually or using a technique that does not guarantee preservation of query answers, then the user wants to check for each relevant query q, whether q given $\mathcal{T}$ is equivalent to q given $\mathcal{T}'$.

Regarding (ii), we consider the complexity of generalized query containment both for CQs and IQs, and for a variety of DLs from the DL-Lite-, $\mathcal{EL}$-, and $\mathcal{ALC}$-families. An interesting first observation is that the two proposed extensions of query containment are intimately related. In fact, containment with an ABox signature Σ *and* two TBoxes can, in most cases, be polynomially reduced to containment with an ABox signature but only one TBox, and to containment without an ABox signature but with two TBoxes. Another relevant observation is that query emptiness, as studied in (Baader et al. 2010), is a special case of our version of query containment, and thus lower complexity bounds carry over. In particular, this means that containment is undecidable in $\mathcal{ALCF}$, the extension of $\mathcal{ALC}$ with functional roles, both for IQs and CQs. For weaker DLs, we exhibit a rich complexity landscape that ranges from PTIME (for IQ-containment in DL-Lite$_{\mathsf{core}}$ and IQ-containment w.r.t. acyclic $\mathcal{EL}$-TBoxes) via Π_2^p-completeness (for CQ-containment in DL-Lite$_{\mathsf{core}}$ and DL-Lite$_{\mathsf{horn}}$) and PSPACE-completeness (for CQ-containment w.r.t. acyclic $\mathcal{EL}$-TBoxes) to EXPTIME-completeness (for IQ-containment and CQ-containment w.r.t. general $\mathcal{EL}$-TBoxes and $\mathcal{EL}_\bot$-TBoxes). We also show decidability of IQ-containment in $\mathcal{ALC}$, with a $\mathrm{P}^{\mathrm{NExp}}$ upper bound. The precise complexity remains open, and so does the decidability of CQ-containment in $\mathcal{ALC}$.

Regarding (iii), we develop strategies for minimizing queries in the presence of ontologies formulated in DL-Lite and $\mathcal{EL}$. We show that, by adopting a suitable minimization strategy, the uniqueness of minimal queries can be regained in DL-Lite, and the resulting queries have an optimal relational structure in the sense that this structure can be found as a subquery in any equivalent query. For $\mathcal{EL}$, we show how to produce an equivalent acyclic query whenever it exists. In this part, we work with the classical notion of query containment instead of with the generalized one.

Throughout the paper, we mostly confine ourselves to proof sketches and (without further notice) defer full proof details to the long version, which is made available at http://www.informatik.uni-bremen.de/∼clu/papers/

2 Preliminaries

We use standard notation for the syntax and semantics of DLs, please see (Baader et al. 2003) for details. As usual, $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ denote countably infinite sets of concept names, role names, and individual names, C, D denote (potentially) composite concepts, A, B concept names, r, s role names, and a, b individual names. We consider the following three families of DLs.

The DL-Lite family. The basic member is DL-Lite$_{\mathsf{core}}$, where *TBoxes* are finite sets of *concept inclusions (CIs)* of the forms

$$B_1 \sqsubseteq B_2 \quad \text{and} \quad B_1 \sqcap B_2 \sqsubseteq \bot$$

with B_1 and B_2 concepts of the form $\exists r$, $\exists r^-$, $\top$, $\bot$, or A. In the extension DL-Lite$_{\mathsf{horn}}$, we additionally allow conjunction, thus obtaining CIs of the forms

$$B_1 \sqcap \cdots \sqcap B_n \sqsubseteq B \quad \text{and} \quad B_1 \sqcap \cdots \sqcap B_n \sqsubseteq \bot$$

cf. (Calvanese et al. 2007; Artale et al. 2009).

The $\mathcal{EL}$-family. Its basic member $\mathcal{EL}$ offers the concept constructors $\top$, $C \sqcap D$, and $\exists r.C$. The extension of $\mathcal{EL}$ with the bottom concept $\bot$ is denoted $\mathcal{EL}_\bot$. In both cases, a *TBox* is a finite set of CIs $C \sqsubseteq D$ with C and D (potentially) compound concepts. We use *concept definitions* $A \equiv C$ in TBoxes as abbreviations for two CIs $A \sqsubseteq C$ and $C \sqsubseteq A$. See for example (Baader, Brandt, and Lutz 2005) for more information on the $\mathcal{EL}$-family of DLs.

The $\mathcal{ALC}$-family. The basic member $\mathcal{ALC}$ offers the concept constructors $\neg C, C \sqcap D$, and $\exists r.C$. $\mathcal{ALCI}$ is the extension of $\mathcal{ALC}$ with the $\exists r^-.C$ constructor, where r^- denotes an *inverse role*, and $\mathcal{ALCF}$ the extension with functional roles. Sometimes we mention $\mathcal{ALCFI}$, which is the union of $\mathcal{ALCI}$ and $\mathcal{ALCF}$ and contains all DLs studied in this paper as a fragment. In $\mathcal{ALC}$ and its extensions, a *TBox* is again a finite set of CIs $C \sqsubseteq D$. See (Baader et al. 2003) for more details.

In any of these DLs, data is stored in an *ABox*, which is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individual names used in the ABox $\mathcal{A}$.

The semantics of DLs is based on interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as usual, see (Baader et al. 2003). An interpretation is a *model* of a TBox $\mathcal{T}$ (resp. ABox $\mathcal{A}$) if it satisfies all concept inclusions in $\mathcal{T}$ (resp. assertions in $\mathcal{A}$), where satisfac-

tion is defined in the standard way. An ABox $\mathcal{A}$ is *consistent* w.r.t. a TBox $\mathcal{T}$ if $\mathcal{A}$ and $\mathcal{T}$ have a common model.

Instance queries (IQs) take the form $A(x)$ and *conjunctive queries (CQs)* take the form $\exists\vec{x}.\varphi(\vec{x},\vec{y})$, where $x,\vec{x},\vec{y}$ denote (tuples of) variables taken from a set $\mathsf{N_V}$ and φ is a conjunction of atoms of the form $A(t)$ and $r(t,t')$ with t,t' *terms*, i.e., individual names or variables from $\vec{x}\cup\vec{y}$. We call the variables in $\vec{y}$ the *answer variables* and those in $\vec{x}$ the *quantified* variables. The *arity* of a CQ is the number of answer variables and $\mathsf{term}(q)$ denotes the set of terms used in q. In what follows, we sometimes slightly abuse notation and use CQ to denote the class of all conjunctive queries and likewise for IQ. Whenever convenient, we treat a CQ (and even an IQ) as a *set* of atoms.

Let $\mathcal{I}$ be an interpretation and q an (instance or conjunctive) query with answer variables $x_1,\dots,x_k$. For $\vec{a} = a_1,\dots,a_k \in \mathsf{N_I}$, an *$\vec{a}$-match* for q in $\mathcal{I}$ is a mapping $\pi : \mathsf{term}(q) \to \Delta^{\mathcal{I}}$ such that $\pi(x_i) = a_i^{\mathcal{I}}$ for $1 \le i \le k$, $\pi(a) = a^{\mathcal{I}}$ for all $a \in \mathsf{term}(q)\cap \mathsf{N_I}$, $\pi(t) \in A^{\mathcal{I}}$ for all $A(t) \in q$, and $(\pi(t_1),\pi(t_2)) \in r^{\mathcal{I}}$ for all $r(t_1,t_2) \in q$. We write $\mathcal{I} \models q[\vec{a}]$ if there is an $\vec{a}$-match of q in $\mathcal{I}$. For a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, we write $\mathcal{T},\mathcal{A} \models q[\vec{a}]$ if $\mathcal{I} \models q[\vec{a}]$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. In this case, $\vec{a}$ is a *certain answer* to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$. We use $\mathsf{cert}_{\mathcal{T}}(q,\mathcal{A})$ to denote the set of all certain answers to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$.

We use the term *predicate* to refer to a concept name or role name and *signature* to refer to a set of predicates. Then $\mathsf{sig}(q)$ denotes the set of predicates used in the query q, and similarly $\mathsf{sig}(\mathcal{T})$ (resp. $\mathsf{sig}(\mathcal{A})$) refers to the signature of a TBox $\mathcal{T}$ (resp. ABox $\mathcal{A}$). Given a signature Σ, a *Σ-ABox* is an ABox using predicates from Σ only.

In the context of query answering in DLs, it is sometimes useful to adopt the unique name assumption (UNA), which requires that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all interpretations $\mathcal{I}$ and all $a,b \in \mathsf{N_I}$ with $a \neq b$. The results obtained in this paper do not depend on the UNA. In fact, it is well-known that in all DLs studied here (with the exception of $\mathcal{ALCF}$), query answers with and without UNA coincide.

The following definition provides the general perspective on query containment in the presence of DL TBoxes that we propose in this paper.

Definition 1. Let $\mathcal{T}_1,\mathcal{T}_2$ be TBoxes, q_1,q_2 CQs with the same arity, and Σ an ABox signature. Then $(\mathcal{T}_1,q_1)$ *is contained in* $(\mathcal{T}_2,q_2)$ *w.r.t.* Σ, written $(\mathcal{T}_1,q_1) \subseteq_\Sigma (\mathcal{T}_2,q_2)$, if for all Σ-ABoxes $\mathcal{A}$ that are consistent w.r.t. $\mathcal{T}_1$ and $\mathcal{T}_2$, we have $\mathsf{cert}_{\mathcal{T}_1}(q_1,\mathcal{A}) \subseteq \mathsf{cert}_{\mathcal{T}_2}(q_2,\mathcal{A})$.

As discussed in the introduction, this definition generalizes the traditional view of query containment in DLs in two directions: by admitting two distinct TBoxes for the two queries and by allowing a restriction to be placed on the ABox signature. If there is only a single TBox $\mathcal{T}$, we write $q_1 \subseteq_{\mathcal{T},\Sigma} q_2$ instead of $(\mathcal{T},q_1) \subseteq_\Sigma (\mathcal{T},q_2)$. When $\Sigma = \mathsf{N_R}\cup\mathsf{N_C}$, we say that the ABox signature Σ is *full* and simply omit it in the subscript of "$\subseteq$". We say that q_1 *and* q_2 *are equivalent w.r.t.* Σ *and* $\mathcal{T}$, written $q_1 \equiv_{\mathcal{T},\Sigma} q_2$, if $q_1 \subseteq_{\mathcal{T},\Sigma} q_2$ and $q_2 \subseteq_{\mathcal{T},\Sigma} q_1$. Again, Σ is omitted if it is full.

To illustrate Definition 1, consider the following TBox $\mathcal{T}$, a slightly simplified fragment of the SNOMED CT ontology:

$$\begin{aligned}
\mathsf{LabTest} &\sqsubseteq \mathsf{LabProc}\sqcap\mathsf{EvalProc}\\
\mathsf{VenipunctBloodTest} &\sqsubseteq \exists\mathsf{focus}.\mathsf{LabTest}\\
\mathsf{VenipunctBloodTest} &\equiv \exists\mathsf{purpose}.\mathsf{BloodSmpl}\sqcap \mathsf{Venipuncture}
\end{aligned}$$

Assume that ABoxes provide data about venipunctures and their purpose, thus the ABox signature Σ contains the concept names Venipuncture and BloodSmpl and the role name purpose, but no other symbols from $\mathcal{T}$. Let

$$q(x) = \exists y.(\mathsf{focus}(x,y) \wedge \mathsf{EvalProc}(y)).$$

Then $q(x) \equiv_{\mathcal{T},\Sigma} \mathsf{VenipunctBloodTest}(x)$. Note that, when Σ is full, $q(x)$ is not equivalent to any query with only one atom.

To illustrate the use of multiple TBoxes, we come back to ontology versioning, already mentioned as a relevant application in the introduction; note that versioning is an active research area with a wide variety of approaches, ranging from purely syntactic to fully semantic, logic-based methods (Noy and Musen 2002; Klein et al. 2002; Jimnez-Ruiz et al. 2011; Gonçalves, Parsia, and Sattler 2011; Konev, Walther, and Wolter 2008). Assume that the above TBox $\mathcal{T}$ is updated to the new version $\mathcal{T}'$ in which the first CI is replaced with $\mathsf{LabTest} \sqsubseteq \mathsf{LabProc}$ (this modification corresponds to an update that was made in the 'real' SNOMED CT). To check whether the above query $q(x)$ is unaffected by the update, the user checks whether $(\mathcal{T},q) \subseteq_\Sigma (\mathcal{T}',q)$. In fact, this is not the case, as witnessed by the ABox

$$\mathcal{A} = \{\mathsf{Venipuncture}(a), \mathsf{purpose}(a,b), \mathsf{BloodSmpl}(b)\}$$

where $a \in \mathsf{cert}_{\mathcal{T}}(q,\mathcal{A})$, but $a \notin \mathsf{cert}_{\mathcal{T}'}(q,\mathcal{A})$.

The main reasoning problems studied in this paper are as follows.

Definition 2. Let $Q \in \{\text{CQ},\text{IQ}\}$ and let $\mathcal{L}$ be any of the DLs introduced above. Deciding

1. *Q-containment in $\mathcal{L}$* means to determine, given $q_1,q_2 \in Q$, $\mathcal{L}$-TBoxes $\mathcal{T}_1,\mathcal{T}_2$, and an ABox signature Σ, whether $(\mathcal{T}_1,q_1) \subseteq_\Sigma (\mathcal{T}_2,q_2)$.
2. *single TBox Q-containment in $\mathcal{L}$* means to determine, given $q_1,q_2 \in Q$, an $\mathcal{L}$-TBox $\mathcal{T}$, and an ABox signature Σ, whether $q_1 \subseteq_{\mathcal{T},\Sigma} q_2$.
3. *full signature Q-containment in $\mathcal{L}$* means to determine, given $q_1,q_2 \in Q$ and $\mathcal{L}$-TBoxes $\mathcal{T}_1,\mathcal{T}_2$, whether $(\mathcal{T}_1,q_1) \subseteq (\mathcal{T}_2,q_2)$.
4. *full signature single TBox Q-containment in $\mathcal{L}$* means to determine, given $q_1,q_2 \in Q$ and an $\mathcal{L}$-TBox $\mathcal{T}$, whether $q_1 \subseteq_{\mathcal{T}} q_2$.

Note that Point 4 is the traditional query containment problem in DLs. The first three problems are closely related. In fact, the following lemma shows that, in the presence of an ABox signature, we can eliminate a second TBox.

Theorem 3. *Let $\mathcal{L} \in \{\mathcal{EL},\mathcal{EL}_\bot,\mathcal{ALC},\mathcal{ALCI},\mathcal{ALCF}\}$ and $Q \in \{$CQ, IQ$\}$. Then Q-containment in $\mathcal{L}$ can be polynomially reduced to single TBox Q-containment in $\mathcal{L}$.*

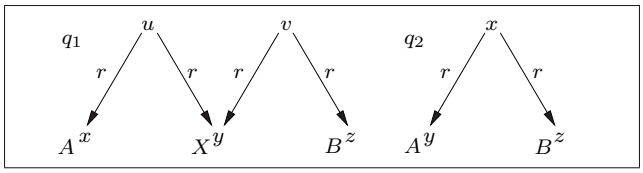

Figure 1: Query containment with restricted ABox signature

Proof. We sketch the proof for $\mathcal{EL}_\bot$ and CQs. Let $\mathcal{T}_1, \mathcal{T}_2$ be $\mathcal{EL}_\bot$-TBoxes, q_1, q_2 CQs of the same arity, and Σ an ABox signature. To simulate $\mathcal{T}_1$ and $\mathcal{T}_2$ using a single TBox, we duplicate the vocabulary. For simplicity, assume first that Σ only contains concept names. We introduce fresh concept names $s_1(A)$ and $s_2(A)$ for every concept name A and fresh role names $s_1(r)$, $s_2(r)$ for every role name r, and extend s_1, s_2 to concepts and CQs in the obvious way; for example, $s_1(C)$ is C with every $X \in \mathsf{N_R} \cup \mathsf{N_C}$ replaced with $s_1(X)$. Let $\mathcal{T}_i' = \{s_i(C) \sqsubseteq s_i(D) \mid C \sqsubseteq D \in \mathcal{T}_i\}$, for $i = 1, 2$ and

$$\mathcal{T} \quad = \quad \mathcal{T}_1' \cup \mathcal{T}_2' \cup \{A \sqsubseteq s_1(A) \sqcap s_2(A) \mid A \in \Sigma \cap \mathsf{N_C}\}.$$

We then have $(\mathcal{T}_1, q_1) \subseteq_\Sigma (\mathcal{T}_2, q_2)$ iff $s_1(q_1) \subseteq_{\mathcal{T},\Sigma} s_2(q_2)$.

The general case, where Σ may contain role names, requires some technical tricks. Since an assertion $r(a, b)$ in the ABox cannot be 'copied' into two assertions $s_1(r)(a, b)$ and $s_2(r)(a, b)$ as in the last component of $\mathcal{T}$, we leave role names from Σ untouched and do not replace them with fresh symbols when defining the TBoxes $\mathcal{T}_i'$. This is unsound as it enables undesired interaction between $\mathcal{T}_1$ and $\mathcal{T}_2$; we rectify this problem by introducing additional concept names E_1 and E_2 that represent the 'active domains' of $\mathcal{T}_1$ and $\mathcal{T}_2$, and syntactically relativize all concepts in $\mathcal{T}_i$ to E_i in the definition of $\mathcal{T}_i'$. ❏

The following result shows that, in the presence of two TBoxes, we can eliminate the ABox signature.

Theorem 4. *For $\mathcal{L} \in \{\mathcal{EL}_\bot, \text{DL-Lite}_{\mathsf{core}}, \text{DL-Lite}_{\mathsf{horn}}, \mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCF}\}$ and $Q \in \{\text{CQ}, \text{IQ}\}$, Q-containment in $\mathcal{L}$ can be polynomially reduced to full signature Q-containment in $\mathcal{L}$.*

Proof. Here, we only illustrate the proof idea using an example. Assume that we are interested in deciding CQ containment in the presence of the TBox $\mathcal{T} = \{A \sqsubseteq \exists r.A\}$ and with the ABox signature restricted to $\Sigma = \{A\}$. Set

$$\mathcal{T}' = \{A \sqsubseteq \exists r'.A, \exists r.\top \sqsubseteq \bot\}$$

Then, for any two CQs q_1, q_2 of the same arity using the symbols A and r only, $q_1 \subseteq_{\mathcal{T},\Sigma} q_2$ iff $(\mathcal{T}, q_1) \subseteq (\mathcal{T}', q_2')$, where q_2' is obtained from q_2 by replacing any occurrence of r by r'. The equivalence holds since for any ABox $\mathcal{A}$ using the non-Σ-symbol r, $\mathcal{A}$ is not consistent w.r.t. $\mathcal{T}'$. ❏

The next example illustrates a central reason for why restricting the ABox signature (or admitting two distinct TBoxes) can make query containment harder.

Example 5. *Restricting the signature of ABoxes or admitting two TBoxes can introduce a form of disjunction. This is illustrated by the queries q_1 and q_2 in Figure 1, where all variables are quantified, together with the very simple DL-Lite$_{\mathsf{core}}$-TBox $\{A \sqsubseteq X, B \sqsubseteq X\}$ and ABox signature $\Sigma = \{A, B, r\}$. Note that, by definition of $\mathcal{T}$ and Σ, an ABox can only 'enforce' the concept name X by an assertion $A(a)$* or *an assertion $B(a)$, which is the mentioned disjunction and results in the fact that $q_1 \equiv_{\mathcal{T},\Sigma} q_2$.*

Some lower bounds in this paper are inherited from query emptiness, a reasoning problem that is defined as follows: given a query q, TBox $\mathcal{T}$, and ABox signature Σ, decide whether there exists a Σ-ABox $\mathcal{A}$ that is consistent w.r.t. $\mathcal{T}$ such that $\mathsf{cert}_\mathcal{T}(q, \mathcal{A}) \neq \emptyset$. The following lemma shows that query emptiness can be polynomially reduced to single TBox query containment, both for IQs and CQs and for any DL studied in this paper.

Lemma 6. *Let q be a CQ, $\mathcal{T}$ be an $\mathcal{ALCFI}$-TBox, Σ be an ABox signature, A be a concept name that does not occur in q, Σ, and $\mathcal{T}$, and q_A be any query with the same arity as q that uses A. Then there exists a Σ-ABox $\mathcal{A}$ that is consistent w.r.t. $\mathcal{T}$ with $\mathsf{cert}_\mathcal{T}(q, \mathcal{A}) \neq \emptyset$ iff $q \not\subseteq_{\mathcal{T},\Sigma} q_A$.*

3 Containment in DL-Lite

For query containment in the DL-Lite family, we find a difference in complexity depending on whether we consider IQs or CQs. We show IQ-containment to be tractable for DL-Lite$_{\mathsf{core}}$ and co-NP-complete for DL-Lite$_{\mathsf{horn}}$, whereas CQ-containment is Π_2^p-complete for both logics. The lower bounds are proven for single TBox containment, and hence also hold for full signature containment by Theorem 4 .

We begin by the tractability result for IQs in DL-Lite$_{\mathsf{core}}$.

Theorem 7. *IQ-containment in DL-Lite$_{\mathsf{core}}$ is in* PTIME.

Proof. It can be proved that $(\mathcal{T}_1, A(x)) \subseteq_\Sigma (\mathcal{T}_2, B(x))$ if and only if for every $C \in \mathsf{N_C} \cap \Sigma \cup \{\exists r, \exists r^- \mid r \in \mathsf{N_R} \cap \Sigma\}$ we have that $\mathcal{T}_1 \models C \sqsubseteq A$ implies $\mathcal{T}_2 \models C \sqsubseteq B$. The latter property can be verified in polynomial time (Calvanese et al. 2007). ❏

We now turn to DL-Lite$_{\mathsf{horn}}$, showing IQ-containment to be coNP-complete and giving a Π_2^p upper bound for CQ-containment.

Theorem 8. *In DL-Lite$_{\mathsf{horn}}$, IQ-containment is* coNP*-complete and CQ-containment is in Π_2^p.*

Proof. We start with instance queries. When $(\mathcal{T}_1, A(x)) \not\subseteq_\Sigma (\mathcal{T}_2, B(x))$, then there is a Σ-ABox $\mathcal{A}$ and an $a \in \mathsf{Ind}(\mathcal{A})$ with $\mathcal{T}_1, \mathcal{A} \models A(a)$ and $\mathcal{T}_2, \mathcal{A} \not\models B(a)$. For each role $r \in \Sigma$, choose an $a_r \in \mathsf{Ind}(\mathcal{A})$ with $r(a, a_r) \in \mathcal{A}$, if such exists. Let $\mathcal{A}'$ be the restriction of $\mathcal{A}$ to the individuals $\{a\} \cup \{a_r \mid r \in \Sigma\}$. Then $\mathcal{T}_1, \mathcal{A}' \models A(a)$ and $\mathcal{T}_2, \mathcal{A}' \not\models B(a)$. To decide instance query non-entailment, we may thus guess an ABox with $|\mathsf{Ind}(\mathcal{A})| \leq |\Sigma| + 1$ and an $a \in \mathsf{Ind}(\mathcal{A})$ and then check in polytime whether $\mathcal{T}_1, \mathcal{A} \models A(a)$ and $\mathcal{T}_2, \mathcal{A} \not\models B(a)$ (Artale et al. 2009). For the lower bound, we use the coNP-hardness of IQ-emptiness (Baader et al. 2010) together with Lemma 6.

We only sketch the proof for CQs. A witness for $(\mathcal{T}_1, q_1) \not\subseteq_\Sigma (\mathcal{T}_2, q_2)$ consists of an ABox $\mathcal{A}$, a tuple of individuals $\vec{a}$ from $\mathsf{Ind}(\mathcal{A})$, and a part of the canonical model for $\mathcal{A}$ and $\mathcal{T}_1$ (see (Kontchakov et al. 2010)) such that q_1 has an $\vec{a}$-match in that part and $\mathcal{T}_2, \mathcal{A} \not\models q_2(\vec{a})$. Similarly

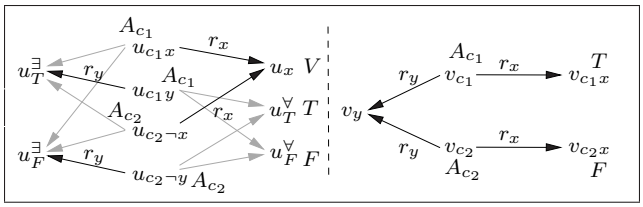

Figure 2: Queries q_1, q_2 for QBF $\forall x \exists y (x \vee y) \wedge (\neg x \vee \neg y)$.

to the case of IQs, we can suppose that the witness ABox $\mathcal{A}$ satisfies $|\mathsf{Ind}(\mathcal{A})| \leq |q| \cdot (|\Sigma| + 1)$. To decide CQ non-containment, we guess such a witness and then verify in coNP that $\mathcal{T}_2, \mathcal{A} \not\models q_2(\vec{a})$. ❑

To complete the picture, a Π_2^p-lower bound for single TBox CQ-containment in DL-Lite$_{\mathsf{core}}$ is proved in the following. Interestingly, it requires only TBox statements of the form $A \sqsubseteq B$ (no roles in the TBox, no disjointness). The construction is inspired by the Π_2^p-lower bound proof for containment of positive existential queries in (Sagiv and Yannakakis 1980).

The proof is by reduction from the validity of Π_2^p-QBFs, i.e., formulas of the form $\forall \vec{x} \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ with φ a propositional formula in CNF. We call the variables in $\vec{x}$ *universal variables* and the variables in $\vec{y}$ *existential variables* and aim at finding $\mathcal{T}, \Sigma, q_1$, and q_2 such that $\forall \vec{x} \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ is valid iff $q_1 \subseteq_{\mathcal{T},\Sigma} q_2$. Set

$$\begin{array}{rcl} \mathcal{T} & = & \{T \sqsubseteq V, F \sqsubseteq V\} \\ \Sigma & = & \{T, F\} \cup \{A_c \mid c \text{ clause in } \varphi\} \cup \{r_x \mid x \in \vec{x} \cup \vec{y}\}. \end{array}$$

The query q_1 consists of the following atoms, where all variables are quantified variables:

- $F(u_F^\forall)$, $T(u_T^\forall)$, and $V(u_x)$ for each universal variable x;
- for each clause c in φ and each literal ℓ in c:
  1. $A_c(u_{c\ell})$;
  2. $r_x(u_{c\ell}, u_x)$ if $\ell = x$ or $\ell = \neg x$, with x a universal variable;
  3. $r_y(u_{c\ell}, u_T^\exists)$ if $\ell = y$ is an existential variable;
  4. $r_y(u_{c\ell}, u_F^\exists)$ if $\ell = \neg y$, with y an existential variable;
  5. for each universal variable x different from the variable in ℓ, the atoms $r_x(u_{c\ell}, u_F^\forall)$ and $r_x(u_{c\ell}, u_T^\forall)$;
  6. for each existential variable y different from the variable in ℓ, the atoms $r_y(u_{c\ell}, u_F^\exists)$ and $r_y(u_{c\ell}, u_T^\exists)$.

The query q_2 consists of the following atoms, for each clause c in φ, where again all variables are quantified variables:

- $A_c(v_c)$;
- for each universal variable x that occurs positively in c, the atoms $r_x(v_c, v_{cx})$ and $T(v_{cx})$;
- for each universal variable x that occurs negatively in c, the atoms $r_x(v_c, v_{cx})$ and $F(v_{cx})$;
- for each existential variable y in c, the atom $r_y(v_c, v_y)$.

An example can be found in Figure 2, where the edges from Points 5 and 6 above are drawn in gray to improve readability. Intuitively, the atoms $V(u_x)$ in q_1, together with the $T(v_{cx})$ and $F(v_{cx})$ in q_2 and the fact that $V \notin \Sigma$ ensures all truth assignments to universal variables are considered (cf. Example 5). The truth assignment for the existential variables is selected via the variables v_y of q_2, which intuitively can either be mapped to where $u_T^\exists$ of q_1 is mapped or to where $u_F^\exists$ of q_1 is mapped. Note that each v_{c_i} of q_2 can be mapped to where any of the $u_{c_i\ell}$ of q_1 is mapped, which corresponds to selecting a literal in c_i that is made true.

Theorem 9. *Single TBox CQ-containment in DL-Lite$_{\mathsf{core}}$ is Π_2^p-hard.*

4 Containment in $\mathcal{EL}$, General TBoxes

We study the complexity of IQ- and CQ-containment in $\mathcal{EL}$, concentrating on the most general form of TBox as introduced in Section 2. Our main result is that query containment is ExpTime-complete both for CQs and IQs. The lower bound holds already for single TBox containment and full signature containment, and the upper bound is for $\mathcal{EL}_\bot$.

We start by establishing an ExpTime lower bound for IQ-containment in the single TBox case. By contrast, note that query emptiness can be checked in polynomial time for $\mathcal{EL}$-TBoxes.

Theorem 10. *Single-TBox IQ-containment in $\mathcal{EL}$ is* ExpTime*-hard.*

Proof. The proof is by reduction from instance query emptiness in $\mathcal{EL}_\bot$, shown to be ExpTime-hard in (Baader et al. 2010). Specifically, we establish the following.

Claim. $A(x)$ is Σ-empty w.r.t. $\mathcal{T}$ iff $A(x) \subseteq_{\mathcal{T}',\Sigma} B(x)$ where $B \in \mathsf{N_C}$ is fresh and $\mathcal{T}'$ is obtained from $\mathcal{T}$ by (a) replacing every assertion $C \sqsubseteq \bot$ in $\mathcal{T}$ with $C \sqsubseteq B$ and (b) adding $\exists r.B \sqsubseteq B$ for every role r in $\mathcal{T}$ and Σ. ❑

To obtain an ExpTime lower bound for full signature containment, we show that an analogue of Theorem 4 holds in $\mathcal{EL}$ for instance queries. The proof is similar to that of Theorem 4 and relies on the fact that we can force the second instance query to hold whenever the ABox contains a non-Σ symbol.

Theorem 11. *In $\mathcal{EL}$, IQ-containment can be polynomially reduced to full signature IQ-containment.*

In the remainder of the section, we aim to prove an ExpTime upper bound for CQ-containment in $\mathcal{EL}_\bot$. Because of Theorem 3, it is sufficient to consider single TBox containment. The key insight is that if $q_1 \not\subseteq_{\mathcal{T},\Sigma} q_2$, then this is witnessed by a *forest-shaped* ABox that consists of a small core whose relational structure is not restricted in any way and a (potentially infinite) tree below each core element. Our decision procedure is based upon a reduction of containment to the existence of a compact description of such a witness ABox.

Tree-shaped queries play an important role in what follows, so we begin by recalling their definition and introducing some relevant notation. We recall that each CQ q can be viewed as a directed graph $G_q = (V_q, E_q)$ with $V_q = \mathsf{term}(q)$ and $E_q = \{(t, t') \mid r(t, t') \in q \text{ for some } r \in \mathsf{N_R}\}$. We call q *tree-shaped* if G_q is a tree and $r(t, t'), s(t, t') \in q$

implies $r = s$. If q is tree-shaped and t is the root of G_q, we call t the *root of* q.

A query q' is *obtained from q by performing fork elimination* if q' is obtained from q by selecting two atoms $r(x, z)$ and $r(y, z)$ with x, y, z quantified variables and $x \neq y$ and then identifying x and y. A query which is obtained from q by repeatedly (but not necessarily exhaustively) performing fork elimination is called a *fork rewriting* of q.

For a CQ q and $t \in \mathsf{term}(q)$, let $\mathsf{Reach}_q(t)$ denote the set of all terms that are reachable from t in the directed graph G_q. For $T \subseteq \mathsf{term}(q)$, we write $q|_T$ to denote for the restriction of q to atoms that contain only terms from T. Define

$$\begin{aligned}
\mathsf{Trees}(q) &:= \{q|_{\mathsf{Reach}_q(x)} \mid x \in \mathsf{term}(q) \text{ a variable} \\
&\qquad \text{and } q|_{\mathsf{Reach}_q(x)} \text{ tree-shaped}\} \\
\mathsf{Trees}^+(q) &:= \{r(t,x) \wedge q' \mid r(t,x) \in q \\
&\qquad \text{and } q' = \emptyset \text{ or } q' \in \mathsf{Trees}(q) \text{ has root } x\} \\
\mathsf{Trees}^*(q) &:= \bigcup_{q' \text{ fork rewriting of } q} \mathsf{Trees}(q') \cup \mathsf{Trees}^+(q').
\end{aligned}$$

The cardinality of $\mathsf{Trees}(q)$ and $\mathsf{Trees}^+(q)$ is clearly polynomial in the size of q, and it follows from results from (Lutz 2008) that this is true of $\mathsf{Trees}^*(q)$ as well.

It is well-known that whenever $\mathcal{T}, \mathcal{A} \not\models q$ for an $\mathcal{EL}_\bot$-TBox $\mathcal{T}$, ABox $\mathcal{A}$, and CQ q, then this is witnessed by a forest-shaped model of $\mathcal{T}$ and $\mathcal{A}$. We now introduce the notion of a match candidate, which intuitively describes a possible match of a CQ in such a model. Let $\mathcal{A}$ be an ABox, q be a CQ with answer variables $x_1, \dots, x_\ell$, and $\vec{a} = a_1, \dots, a_\ell \in \mathsf{Ind}(\mathcal{A})^\ell$ be a candidate answer to q in $\mathcal{A}$. An *$\vec{a}$-match candidate for q in $\mathcal{A}$* is a tuple $\Pi = \langle p_0, p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m, f\rangle$ where $p_0, p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m$ is a partitioning of q and $f : \mathsf{term}(q) \to \mathsf{Ind}(\mathcal{A})$ maps each term in q to an individual name in $\mathcal{A}$. Let $p_i = r_i(t_i, y_i) \wedge p_i'$ for $1 \leq i \leq n$. We require that the following conditions are satisfied:

1. $p_1, \dots, p_n \in \mathsf{Trees}^+(q)$ and $\hat{p}_1, \dots, \hat{p}_m \in \mathsf{Trees}(q)$;
2. $f(x_i) = a_i$ for $1 \leq i \leq \ell$;
3. $f(a) = a$ for all $a \in \mathsf{term}(q) \cap \mathsf{N_I}$;
4. $A(t) \in p_0$ implies $A(f(t)) \in \mathcal{A}$;
5. $r(t, t') \in p_0$ implies $r(f(t), f(t')) \in \mathcal{A}$;
6. the p_i' and the $\hat{p}_i$ contain only quantified variables;
7. if $s \in \{p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m\}$ and $x, y \in \mathsf{term}(s)$, then $f(x) = f(y)$;
8. $\mathsf{term}(p_0) \cup \{t_1, \dots, t_n\}, \mathsf{term}(p_1'), \dots, \mathsf{term}(p_n'), \mathsf{term}(\hat{p}_1), \dots, \mathsf{term}(\hat{p}_m)$ are pairwise disjoint.

Intuitively, the function f is used to map (i) each term in p_0 and (ii) each subquery p_i and $\hat{p}_i$ ($i > 0$) to some individual name. To achieve this, we use the uniformity condition 7 and set $f(p_i) = f(x)$ (resp. $f(\hat{p}_i) = f(x)$) where x is any variable in p_i (resp. $\hat{p}_i$). Now, a match candidate describes a match of q in a forest-shaped model of $\mathcal{T}$ and $\mathcal{A}$ as follows: the atoms in p_0 are mapped to the core of the forest-shaped model with each $t \in \mathsf{term}(p_0)$ being mapped to $f(t)$. Each query p_i is mapped to the tree below $f(p_i)$ in a rooted way (the root is the term t_i), and each query $\hat{p}_i$ is mapped to the tree below $f(\hat{p}_i)$ in a non-rooted way.

We now relate query entailment to the existence of a match candidate. To do this, we represent tree-shaped CQs as concepts expressed in $\mathcal{EL}$ or its extension $\mathcal{EL}^u$. For our purposes, an $\mathcal{EL}^u$ concept takes the form C or $\exists u.C$ where C is an $\mathcal{EL}$-concept and u is the *universal role*, interpreted as $u^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. To every tree-shaped CQ q and term $t \in \mathsf{term}(q)$, we associate an $\mathcal{EL}$-concept $C_{q,t}$ as follows:

$$C_{q,t} = \bigsqcap_{A(t) \in q} A \sqcap \bigsqcap_{r(t,t') \in q} \exists r.C_{q,t'}.$$

We use C_q to abbreviate $C_{q,t}$ when t is the root of q and C_q^u to denote the $\mathcal{EL}^u$-concept $\exists u.C_q$.

Lemma 12. *Suppose $\mathcal{A}$ is consistent with $\mathcal{T}$, and let $\mathsf{close}(\mathcal{T}, \mathcal{A})$ denote the ABox*

$$\mathcal{A} \cup \{A(a) \mid \mathcal{T}, \mathcal{A} \models A(a), a \in \mathsf{Ind}(\mathcal{A}), A \in \mathsf{N_C}\}.$$

Then $\mathcal{T}, \mathcal{A} \models q(\vec{a})$ iff there exists a fork rewriting q' of q and an $\vec{a}$-match candidate $\Pi = \langle p_0, p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m, f\rangle$ for q' and $\mathsf{close}(\mathcal{T}, \mathcal{A})$ such that $\mathcal{T}, \mathcal{A} \models C(a)$ for all concepts C from $\{C_{p_i} \mid f(p_i) = a\} \cup \{C_{\hat{p}_i}^u \mid f(\hat{p}_i) = a\}$.

According to the preceding lemma, a query is not entailed just in the case that for each match candidate for a fork rewriting, one of the required concept assertions is not entailed. This leads us to define the notion of a spoiler, which describes a possible way of avoiding a query match. We say a map

$$\nu : \mathsf{Ind}(\mathcal{A}) \to 2^{\{C_{q'}, C_{q'}^u \mid q' \in \mathsf{Trees}^*(q)\}}$$

is an *$\vec{a}$-spoiler for q w.r.t. $\mathcal{A}$* if for every fork rewriting q' of q and $\vec{a}$-match candidate $\Pi = \langle p_0, p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m, f\rangle$ for q' in $\mathcal{A}$, there is a p_i such that $C_{p_i} \in \nu(f(p_i))$ or a $\hat{p}_i$ such that $C_{\hat{p}_i}^u \in \nu(f(\hat{p}_i))$.

We are now ready to define compact witnesses, which are the key ingredient to our upper bound. A *compact witness for $q_1 \not\subseteq_{\mathcal{T},\Sigma} q_2$* is a tuple $(\mathcal{A}_w, \vec{a}_w, q_1', \Pi_w, \nu_w)$ where $\mathcal{A}_w$ is a $\mathsf{sig}(\mathcal{T})$-ABox with $|\mathsf{Ind}(\mathcal{A}_w)|$ bounded by $|\mathsf{term}(q_1)|$, $\vec{a}_w$ a candidate answer to q_1 and q_2 in $\mathcal{A}_w$, q_1' a fork rewriting of q_1, $\Pi_w = \langle p_0, p_1, \dots, p_n, \hat{p}_1, \dots, \hat{p}_m, f\rangle$ an $\vec{a}$-match candidate for q_1' in $\mathcal{A}_w$, and ν_w an $\vec{a}$-spoiler for q_2 w.r.t. $\mathcal{A}_w$ such that the following conditions are satisfied:

1. $\mathcal{A}_w$ is consistent w.r.t. $\mathcal{T}$;
2. if $\mathcal{T}, \mathcal{A}_w \models A(a)$ with $a \in \mathsf{Ind}(\mathcal{A}_w)$ and $A \in \mathsf{N_C}$, then $A(a) \in \mathcal{A}_w$;
3. all role names in $\mathcal{A}_w$ are from Σ;
4. for each $a \in \mathsf{Ind}(\mathcal{A}_w)$, there is a tree-shaped Σ-ABox $\mathcal{A}_a$ with root a which is consistent with $\mathcal{T}$ and such that
   (a) $A(a) \in \mathcal{A}_w$ iff $A(a) \in \mathcal{A}_a$ for all $A \in \mathsf{N_C} \cap \Sigma$;
   (b) $\mathcal{T}, \mathcal{A}_a \models A(a)$ iff $A(a) \in \mathcal{A}_w$ for all $A \in \mathsf{N_C} \setminus \Sigma$;
   (c) $\mathcal{T}, \mathcal{A}_a \models C(a)$ for all concepts C from
   $$\{C_{p_i} \mid f(p_i) = a\} \cup \{C_{\hat{p}_i}^u \mid f(\hat{p}_i) = a\};$$
   (d) $\mathcal{T}, \mathcal{A}_a \not\models C(a)$, for every $C \in \nu_w(a)$.

Given a compact witness $(\mathcal{A}_w, \vec{a}_w, q_1', \Pi_w, \nu_w)$ for $q_1 \not\sqsubseteq_{\mathcal{T},\Sigma} q_2$, the desired forest-shaped witness ABox, call it $\mathcal{A}_f$, can be constructed by taking the role assertions in $\mathcal{A}_w$ and attaching the tree-shaped Σ-ABoxes described by condition 4. Note that because of conditions 2 and 4(a-b), the concept assertions in $\mathcal{A}_w$ are precisely the atomic concept assertions concerning individuals in $\mathcal{A}_w$ which are entailed from $\mathcal{T}, \mathcal{A}_f$. Consistency of the whole ABox $\mathcal{A}_f$ follows from conditions 1 and 4. To show entailment of $q_1(\vec{a})$, we use the match candidate Π_w together with condition 4(c), and for non-entailment of $q_2(\vec{a})$, we use the spoiler ν_w together with condition 4(d).

The following theorem gives the reduction announced earlier. It holds only for *normalized TBoxes* whose axioms are of the forms $A \sqsubseteq B$, $A_1 \sqcap A_2 \sqsubseteq B$, $A_1 \sqsubseteq \exists r.A_2$, and $\exists r.A \sqsubseteq B$, where $A, A_1, A_2 \in \mathsf{N_C}$ and $B \in \mathsf{N_C} \cup \{\bot\}$. This is unproblematic since every $\mathcal{EL}_\bot$-TBox can be transformed into a normalized $\mathcal{EL}_\bot$-TBox that is a model conservative extension of $\mathcal{T}$ (Baader, Brandt, and Lutz 2005).

Theorem 13. *If $\mathcal{T}$ is normalized, $q_1 \not\sqsubseteq_{\mathcal{T},\Sigma} q_2$ iff there is a compact witness for $q_1 \not\sqsubseteq_{\mathcal{T},\Sigma} q_2$.*

Based on Theorem 13, our decision procedure is as follows: enumerate all tuples $(\mathcal{A}_w, \vec{a}_w, q_1', \Pi_w, \nu_w)$ that are candidates for compact witnesses for $q_1 \not\sqsubseteq_{\mathcal{T},\Sigma} q_2$, i.e., $\mathcal{A}_w$ is a $\mathsf{sig}(\mathcal{T})$-ABox with $|\mathsf{Ind}(\mathcal{A}_w)|$ bounded by $|\mathsf{term}(q_1)|$, $\vec{a}_w$ a candidate answer to q_1 and q_2 in $\mathcal{A}_w$, q_1' a fork rewriting of q_1, $\Pi_w = \langle p_0, p_1, \ldots, p_n, \hat{p}_1, \ldots, \hat{p}_m, f\rangle$ an $\vec{a}$-match candidate for q_1' in $\mathcal{A}_w$, and ν_w an $\vec{a}$-spoiler for q_2 w.r.t. $\mathcal{A}_w$. It is not hard to verify that there are only exponentially many such tuples. For each tuple, verify whether it satisfies conditions 1 to 4 of compact witnesses. This can clearly be done in polynomial time for conditions 1 to 3, and thus it remains to deal with condition 4. This is less straightforward, and we use an automaton construction to show the following.

Proposition 14. *Given an $\mathcal{EL}_\bot$-TBox $\mathcal{T}$, an ABox signature Σ, and sets of $\mathcal{EL}^u$-concepts Ψ_1 and Ψ_2, it is in* ExpTime *to decide whether there is a tree-shaped Σ-ABox $\mathcal{A}$ with root a such that*

1. *$\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$;*
2. *$\mathcal{T}, \mathcal{A} \models C(a)$ for all $C \in \Psi_1$;*
3. *$\mathcal{T}, \mathcal{A} \not\models C(a)$ for all $C \in \Psi_2$.*

We thus obtain the desired upper bound:

Theorem 15. *CQ-containment in $\mathcal{EL}_\bot$ is in* ExpTime.

5 Containment in $\mathcal{EL}$, Classical TBoxes

A *classical $\mathcal{EL}$-TBox* is a set $\mathcal{T}$ of concept definitions $A \equiv C$ such that each A is a concept name and no concept name appears on the left-hand side of multiple definitions. $\mathcal{T}$ is called *acyclic* if it is classical and no concept name depends on itself; that is, for any sequence $A_0 \equiv C_0, \ldots, A_n \equiv C_n$ in $\mathcal{T}$ with A_{i+1} a subconcept of C_i, $0 \leq i < n$, A_0 is not a subconcept of C_n. The length of a longest such sequence is called the *definitorial depth* $d(\mathcal{T})$ of an acyclic TBox $\mathcal{T}$.

We show that IQ-containment is tractable for classical $\mathcal{EL}$-TBoxes and CQ-containment is PSpace-complete for acyclic $\mathcal{EL}$-TBoxes. The PSpace lower bound holds already for single TBox CQ-containment and full signature CQ-containment. The exact complexity of CQ-containment for (possibly cyclic) classical $\mathcal{EL}$-TBoxes remains open between PSpace and ExpTime.

Theorem 16. *IQ-containment in $\mathcal{EL}$ with classical TBoxes is in* PTime.

The proof of Theorem 16 applies techniques introduced in (Konev, Walther, and Wolter 2008; Konev et al. 2011) for proving that conservative extensions, Σ-entailment and Σ-inseparability between classical $\mathcal{EL}$-TBoxes are tractable. Here, we consider in more detail CQ-containment for acyclic $\mathcal{EL}$-TBoxes.

Theorem 17. *CQ-containment in $\mathcal{EL}$ with acyclic TBoxes is in* PSpace.

The polynomial reduction of CQ-containment to single TBox CQ-containment given in Theorem 3 can be modified to prove that CQ-containment for acyclic $\mathcal{EL}$-TBoxes is polynomially reducible to single TBox CQ-containment for acyclic $\mathcal{EL}$-TBoxes. Thus, it suffices to prove the PSpace upper bound for single TBox containment. Note that we can w.l.o.g. assume that Σ contains a distinguished role name $r_@$ that does not occur in the TBox and the queries since adding such a name does not impact the result of deciding containment.

We use a variation of the algorithm described in Section 4, and in particular Theorem 13. The central observation is that a PSpace procedure for single TBox CQ-containment w.r.t. *acyclic $\mathcal{EL}$-TBoxes* is obtained by guessing a candidate tuple for a compact witness and then using a variant of Proposition 14 to verify that it is indeed a compact witness. The additional role name $r_@$ can be traced all the way through to Proposition 14, thus it suffices to prove the following.

Proposition 18. *Given an acyclic $\mathcal{EL}$-TBox $\mathcal{T}$, an ABox signature Σ, and sets of $\mathcal{EL}^u$-concepts Ψ_1 and Ψ_2 such that the role name $r_@$ occurs in Σ, but not in $\mathcal{T}$, Ψ_1, and Ψ_2, it is in* PSpace *to decide whether there is a tree-shaped Σ-ABox $\mathcal{A}$ with root a such that Points 1 to 3 of Proposition 14 hold.*

Proof. (sketch) We assume that $\mathcal{T}$ is in a certain normal form in which each $A \equiv C$ is of the form $A \equiv \exists r.B$ or $A \equiv B_1 \sqcap \cdots \sqcap B_n$, where $B, B_1, \ldots, B_n$ are concept names; details are given in the long version. Call a concept name A *primitive in $\mathcal{T}$* if it does not occur on the left hand side of a concept definition in $\mathcal{T}$. Let $\Psi_i = \Psi_i^l \cup \{\exists u.C \mid C \in \Psi_i^u\}$, where Ψ_i^l and Ψ_i^u are sets of $\mathcal{EL}$-concepts, for $i = 1, 2$. Let S be the set of subconcepts of $\mathcal{T} \cup \Psi_1^l \cup \Psi_1^u \cup \Psi_2^l \cup \Psi_2^u$ and w.l.o.g. $\top \in S$.

We define a recursive polynomial space procedure computing the predicate $\mathsf{ABox}(\Gamma, k)$ for $\Gamma \subseteq S$ such that $\mathsf{ABox}(\Gamma, k) = 1$ iff there is a tree-shaped Σ-ABox $\mathcal{A}$ with root a of depth $\leq k$ and such that $\mathcal{T}, \mathcal{A} \models C(a)$ iff $C \in \Gamma$ (for all $C \in S$), and $\mathcal{T}, \mathcal{A} \not\models \exists u.C(a)$ for all $C \in \Psi_2^u$. $\mathsf{ABox}(\Gamma, 0) = 1$ can be checked in polyspace. For $n > 0$, one can show that $\mathsf{ABox}(\Gamma, n) = 1$ if and only if the following conditions are satisfied:

(b1) $\top \in \Gamma$ and $\Psi_2^u \subseteq S \setminus \Gamma$;

(b2) for all $C_1, C_2 \in S$: if $C_1 \in \Gamma$ and $\mathcal{T} \models C_1 \sqsubseteq C_2$, then $C_2 \in \Gamma$;

(b3) for all $A \in \Gamma \cap \Sigma$ and $C \in \Psi_2^u$: $\mathcal{T} \not\models A \sqsubseteq \exists u.C$;

(b4) for all A primitive in $\mathcal{T}$: $A \in \Gamma$ iff there exists $B \in \Sigma \cap \Gamma$ such that $\mathcal{T} \models B \sqsubseteq A$;

(b5) for all $\exists r.C \in S$: $\exists r.C \in \Gamma$ iff there exists $B \in \Sigma \cap \Gamma$ such that $\mathcal{T} \models B \sqsubseteq \exists r.C$ or there exist $\Gamma' \subseteq S$ such that, recursively: (1) $C \in \Gamma'$, (2) for all $D \in S$: if $\mathcal{T} \models \exists r.(\bigsqcap \Gamma') \sqsubseteq D$, then $D \in \Gamma$, (3) $\mathsf{ABox}(\Gamma', n-1) = 1$.

Conditions (b1) to (b5) can be checked in polyspace. Let $d(S)$ denote the maximal number of nestings of existential restriction in concepts from S. Now one can show that an ABox with the properties from Proposition 18 exists if

1. there exists $\Gamma \subseteq S$ such that $\mathsf{ABox}(\Gamma, d(\mathcal{T}) + d(S)) = 1$, $\Psi_1^l \subseteq \Gamma$ and $\Psi_2^l \cap \Gamma = \emptyset$;
2. for all $C \in \Psi_1^u$: there exists $\Gamma \subseteq S$ such that $\mathsf{ABox}(\Gamma, d(\mathcal{T}) + d(S)) = 1$ and $C \in \Gamma$ or $\mathcal{T} \models A \sqsubseteq \exists u.C$ for some $A \in \Gamma \cap \Sigma$.

One obtains the required tree-shaped Σ-ABox by adding the roots of the tree-shaped Σ-ABoxes obtained in Point 2 for $C \in \Psi_1^u$ as $r_@$-successors to the root of the Σ-ABox obtained in Point 1. ❑

We now state the matching lower bound.

Theorem 19. *CQ-containment in $\mathcal{EL}$ with acyclic TBoxes is* PSPACE*-hard. This is true already for single TBox and full signature containment.*

We consider single TBox containment and give a polytime reduction of the validity of QBF formulas $\varphi = Q_1 p_1 \dots Q_n p_n.\vartheta$ with $\vartheta = \bigsqcap_{c \in C} c$ a propositional formula in CNF, C its clauses, and $Q_i \in \{\forall, \exists\}$, for $1 \leq i \leq n$. Recall that a *validation tree* for φ is a tree of depth n in which every level (except the leaves) corresponds to one of the quantifiers in φ. In $\forall p_i$-levels, each node has two successors, one for $p_i = \top$ and one for $p_i = \bot$. In $\exists p_i$-levels, each node has one successor, either for $p_i = \top$ or for $p_i = \bot$. Thus, every branch of a validation tree corresponds to a truth assignment to the variables $p_1, \dots, p_n$, and it is required that the propositional formula ϑ evaluates to true on every branch. We say φ is *valid* iff there exists a validation tree for φ.

Now let $\varphi = Q_1 p_1 \dots Q_n p_n.\vartheta$ be of the form above. In the reduction, the existence of a validation tree is encoded in the existence of a Σ-ABox $\mathcal{A}$ that witnesses $q_1 \not\subseteq_{\mathcal{T},\Sigma} q_2$. To encode the tree structure, we use a role name r to represent the edges of the validation tree, and the concept names $L_0, \dots, L_n$ to identify the n levels. The truth values of the variables $p_1, \dots, p_n$ are represented via the concept names $P_1, \dots, P_n$ (indicating truth) and $\bar{P}_1, \dots, \bar{P}_n$ (indicating falsity). Concept names $V_1, \dots, V_n$ are used to indicate that either P_i or $\bar{P}_i$ holds. Finally concept names P_c, $c \in C$, indicate that the clause c of ϑ is true. We set $\Sigma = \{r\} \cup \{P_i, \bar{P}_i \mid i \leq n\}$ and let $\mathcal{T}$ consist of, for $i \leq n$:

$$P_i \sqsubseteq V_i \sqcap \bigsqcap_{P_i \in c, c \in C} P_c, \quad \bar{P}_i \sqsubseteq V_i \sqcap \bigsqcap_{\bar{P}_i \in C, c \in C} P_c,$$

for $i < n$ and $Q_i = \forall$:

$$L_i \equiv \exists r.(P_{i+1} \sqcap L_{i+1} \sqcap \bigsqcap_{j \leq i} V_j) \sqcap \exists r.(\bar{P}_{i+1} \sqcap L_{i+1} \sqcap \bigsqcap_{j \leq i} V_j),$$

for $i < n$ and $Q_{i+1} = \exists$: $\quad L_i \equiv \exists r.(L_{i+1} \sqcap \bigsqcap_{j \leq i+1} V_j)$,

and, finally, $L_n \equiv \bigsqcap_{c \in C} P_c$.

Observe that if $\mathcal{T}, \mathcal{A} \models L_0(a)$ for a Σ-ABox $\mathcal{A}$, then one can show by induction that there exists a tree in $\mathcal{A}$ with root a satisfying L_i at every node of level i and satisfying $P_i \vee \bar{P}_i$ at all nodes of level j with $n \geq j \geq i$ (since V_i is satisfied in all those nodes and this can only be enforced by $P_i \vee \bar{P}_i$). Moreover, in $\forall p_i$-levels we have a successor node in which P_i holds and a successor node in which $\bar{P}_i$ holds. Finally, P_c is true in all leaf nodes for all $c \in C$. We, therefore, have a validation tree in which ϑ evaluates to true in every leaf iff there exists a Σ-ABox $\mathcal{A}$ with $\mathcal{T}, \mathcal{A} \models L_0(a)$ and $\mathcal{T}, \mathcal{A} \not\models C(a)$, for every $C \in \mathcal{C}$, where

$$\begin{aligned} \mathcal{C} \quad := \quad & \{\exists r^j.(P_i \sqcap \bar{P}_i) \mid 1 \leq j \leq n, 1 \leq i \leq n\} \cup \\ & \{\exists r^j.(\bar{P}_i \sqcap \exists r.P_i) \mid 1 \leq j \leq n-1, 1 \leq i < n\} \cup \\ & \{\exists r^j.(P_i \sqcap \exists r.\bar{P}_i) \mid 1 \leq j \leq n-1, 1 \leq i \leq n\} \end{aligned}$$

It follows that φ is valid iff $L_0(x) \not\subseteq_{\mathcal{T},\Sigma} \bigsqcup_{C \in \mathcal{C}} C(x)$. It now remains to encode the disjunction on the right-hand-side of this containment problem into an extension of $L_0(x)$ to a CQ. This is done in the long version by modifying a construction from a lower bound proof for ABox updates in $\mathcal{EL}$ (Liu, Lutz, and Milicic 2008). The proof for full signature containment is similar.

6 Containment in Expressive DLs

For $\mathcal{ALC}$ and $\mathcal{ALCI}$, we establish a $\mathrm{P}^{\mathrm{NExp}}$ upper bound on IQ-containment using the same technique that was used in (Baader et al. 2010) to prove such an upper bound for query emptiness. Embarrassingly enough, we do not know whether this bound is tight, neither for emptiness nor for containment, and we do not even know whether CQ-containment is decidable in $\mathcal{ALC}$ and $\mathcal{ALCI}$. In $\mathcal{ALCF}$, both IQ-containment and CQ-containment are undecidable as a direct consequence of the corresponding results for query emptiness, shown in (Baader et al. 2010).

Fix IQs $A_1(x)$, $A_2(x)$, a consistent $\mathcal{ALCI}$-TBox $\mathcal{T}$, and an ABox signature Σ such that it is to be decided whether $A_1(x) \subseteq_{\mathcal{T},\Sigma} A_2(x)$. The central observation is that one can construct, in exponential time, a single candidate Σ-ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ such that $A_1(x) \not\subseteq_{\mathcal{T},\Sigma} A_2(x)$ iff $\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma} \models q_1[a]$ and $\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma} \not\models q_2[a]$ for some $a \in \mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma})$. $\mathcal{A}_{\mathcal{T},\Sigma}$ is called the *canonical Σ-ABox for $\mathcal{T}$* and constructed as follows. The *closure* $\mathsf{cl}(\mathcal{T}, \Sigma)$ is the smallest set that contains $\Sigma \cap \mathsf{N_C}$ as well as all concepts that occur (potentially as a subconcept) in $\mathcal{T}$ and is closed under single negations. A *type* for $\mathcal{T}$ and Σ is a set $t \subseteq \mathsf{cl}(\mathcal{T}, \Sigma)$ such that for some model $\mathcal{I}$ of $\mathcal{T}$ and some $d \in \Delta^{\mathcal{I}}$, we have $t = \{C \in \mathsf{cl}(\mathcal{T}, \Sigma) \mid d \in C^{\mathcal{I}}\}$. Let $\mathfrak{T}_{\mathcal{T},\Sigma}$ denote the set of all types for $\mathcal{T}$ and Σ. Let a_t be mutually distinct individual names for $t \in \mathfrak{T}_{\mathcal{T},\Sigma}$ and define $\mathcal{A}_{\mathcal{T},\Sigma}$ as follows:

$$\begin{aligned} \mathcal{A}_{\mathcal{T},\Sigma} = \{ & A(a_t) \mid A \in t \cap \Sigma \text{ and } t \in \mathfrak{T}_{\mathcal{T},\Sigma}\} \cup \\ & \{r(a_t, a_{t'}) \mid t, t' \in \mathfrak{T}_{\mathcal{T},\Sigma} \text{ and } r \in \Sigma \text{ and} \\ & \text{for all } \exists r.C \in \mathsf{cl}(\mathcal{T}, \Sigma) : C \in t' \Rightarrow \exists r.C \in t\}. \end{aligned}$$

The set $\mathfrak{T}_{\mathcal{T},\Sigma}$ can be computed in exponential time by making use of well-known EXPTIME procedures for concept satisfiability w.r.t. TBoxes in $\mathcal{ALCI}$ (Baader et al. 2003). Thus, $\mathcal{A}_{\mathcal{T},\Sigma}$ can be computed in exponential time.

The main property of $\mathcal{A}_{\mathcal{T},\Sigma}$ is that $A_1(x) \subseteq_{\mathcal{T},\Sigma} A_2(x)$ iff $\mathsf{cert}_{\mathcal{T}}(A_1(x), \mathcal{A}_{\mathcal{T},\Sigma}) \subseteq \mathsf{cert}_{\mathcal{T}}(A_2(x), \mathcal{A}_{\mathcal{T},\Sigma})$, which is proved using homomorphisms between ABoxes. Based on $\mathcal{A}_{\mathcal{T},\Sigma}$, we can now prove the main result of this section.

Theorem 20. *IQ-containment in $\mathcal{ALCI}$ is in* $\mathrm{P}^{\mathrm{NEXP}}$.

Proof. By Theorem 3, it is sufficient to consider single TBox containment. We show that non-containment is in $\mathrm{NP}^{\mathrm{NEXP}}$ and derive from $\mathrm{NP}^{\mathrm{NEXP}} \subseteq \mathrm{P}^{\mathrm{NEXP}}$ the desired result (Hemachandra 1987). Let $\mathcal{T}$ be a consistent $\mathcal{ALCI}$-TBox, Σ be an ABox signature, and $A_1(x)$, $A_2(x)$ be IQs for which it is to be decided whether $A_1(x) \subseteq_{\mathcal{T},\Sigma} A_2(x)$ is *not* the case. The algorithm guesses $a \in \mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma})$, and then checks (1) $a \in \mathsf{cert}_{\mathcal{T}}(A_1(x), \mathcal{A}_{\mathcal{T},\Sigma})$ and (2) $a \notin \mathsf{cert}_{\mathcal{T}}(A_2(x), \mathcal{A}_{\mathcal{T},\Sigma})$, by calling a NEXPTIME oracle that decides the following problem: given an $\mathcal{ALCI}$-TBox $\mathcal{T}'$, signature Σ', individual $a' \in \mathsf{Ind}(\mathcal{A}_{\mathcal{T}',\Sigma'})$ and IQ $A(x)$, does $a' \notin \mathsf{cert}_{\mathcal{T}'}(A(x), \mathcal{A}_{\mathcal{T}',\Sigma'})$ hold? Such an oracle exists: it computes the canonical ABox $\mathcal{A}_{\mathcal{T}',\Sigma'}$ and guesses a map $\pi : \mathsf{Ind}(\mathcal{A}_{\mathcal{T}',\Sigma'}) \to \mathfrak{T}_{\mathcal{T}',\Sigma'}$ and accepts if (i) $A \notin \pi(a')$, (ii) $C(c) \in \mathcal{A}_{\mathcal{T}',\Sigma'}$ implies $C \in \pi(c)$, and (iii) $r(b,c) \in \mathcal{A}_{\mathcal{T}',\Sigma'}$, $C \in \pi(c)$, and $\exists r.C \in \mathsf{cl}(\mathcal{T}', \Sigma')$ implies $\exists r.C \in \pi(b)$. ❑

The best known lower bound for IQ-containment in $\mathcal{ALC}$ and $\mathcal{ALCI}$ is EXPTIME. It stems from an easy reduction of unsatisfiability in $\mathcal{ALC}$: a concept C is unsatisfiable w.r.t. $\mathcal{T}$ iff $A(x) \subseteq_{\mathcal{T},\Sigma} B(x)$ where $\Sigma = \{A\}$ and $\mathcal{T} = \{A \sqsubseteq C\}$.

We close with stating undecidability for $\mathcal{ALCF}$.

Theorem 21. *In $\mathcal{ALCF}$, IQ-containment and CQ-containment are undecidable.*

7 Query Optimization

A classical result by Chandra and Merlin states that, in relational databases, any two CQs that are equivalent and minimal w.r.t. set inclusion must be isomorphic (Chandra and Merlin 1977). Given a CQ q, one can thus find the *unique minimal CQ* that is equivalent to q by applying lazy minimization: repeatedly remove atoms that are *redundant* in the sense that dropping them preserves equivalence. The order in which atoms are dropped is irrelevant, and thus the only non-trivial part is checking equivalence (i.e., two query containment checks).

It is not hard to see that, in the presence of TBoxes, Chandra-Merlin uniqueness of minimal CQs fails. This is true even for the classical version of query containment, the full signature single TBox case, which we will generally assume throughout this section. Given a TBox $\mathcal{T}$, we call a CQ q *$\mathcal{T}$-minimal w.r.t. set inclusion* if there is no $q' \subsetneq q$ with $q' \equiv_{\mathcal{T}} q$.

Example 22. *(1) Let $\mathcal{T} = \{A \equiv B\}$ and $q(x) = A(x) \wedge B(x)$. Then $q(x) \equiv_{\mathcal{T}} A(x)$ and $q(x) \equiv_{\mathcal{T}} B(x)$, and both $A(x)$ and $B(x)$ are $\mathcal{T}$-minimal w.r.t. set inclusion, but not isomorphic.*

(2)
$$\mathcal{T} = \{A \sqsupseteq \exists r.(B_1 \sqcap \exists s.\top) \sqcap \exists r.(B_2 \sqcap \exists s.\top),\ A \sqsubseteq \exists r.(B_1 \sqcap B_2 \sqcap \exists s.\top)\ \}$$
$$q(x) = \exists y_1, y_2, y_3.B_1(y_1) \wedge B_2(y_2) \wedge r(x,y_1) \wedge r(x,y_2) \wedge s(y_1,y_3) \wedge s(y_2,x_3).$$

Then $q(x)$ is $\mathcal{T}$-minimal w.r.t. set inclusion, but equivalent to the smaller (and non-isomorphic) query $A(x)$.

Note that Example (1) above uses only CIs of the very simple form $A \sqsubseteq B$ and thus applies to any reasonable DL. The second example, where the structural differences between the original and minimized query is large, is formulated in $\mathcal{EL}$. While these examples demonstrate that naive lazy minimization does not yield the desired result in the presence of DL TBoxes, we show in the following that strong guarantees on minimized queries can be recovered when applying more careful minimization strategies.

Optimization in DL-Lite

We consider the basic member DL-Lite$_{\mathsf{core}}$ of the DL-Lite family. A first observation is that, although the result of standard lazy minimization is not unique, it still yields queries of *minimum cardinality*. In the following, we use $\#q$ to denote the number of atoms in the CQ q. Clearly, one can view a CQ q as an ABox $\mathcal{A}_q$ by treating concept atoms as concept assertions and role atoms as role assertions. We say that q is *consistent with $\mathcal{T}$* if $\mathcal{A}_q$ and $\mathcal{T}$ have a common model.

Theorem 23. *Let $\mathcal{T}$ be a DL-Lite$_{\mathsf{core}}$-TBox and q_1, q_2 CQs such that $q_1 \equiv_{\mathcal{T}} q_2$ and q_1, q_2 are $\mathcal{T}$-minimal w.r.t. set inclusion and consistent with $\mathcal{T}$. Then $\#q_1 = \#q_2$.*

We now show that even stronger guarantees can be obtained, concentrating on the class of *rooted CQs*, which are connected CQs that contain at least one answer variable or individual name. For these, we show that by adopting a suitable minimization strategy, we can find an equivalent query that is $\mathcal{T}$-minimal w.r.t. set inclusion and whose relational structure (restriction to role atoms) can be found in *any* equivalent query as a subquery. The strategy also guarantees a *unique result* of query minimization, similarly to Chandra and Merlin. We slightly generalize CQs and ABoxes by also admitting concept atoms of the form $\exists r(t)$ and ABox assertions of the form $\exists r(a)$.

Let $\mathcal{T}$ be a DL-Lite$_{\mathsf{core}}$-TBox. To break ties between equivalent concepts during minimization (see Part (1) of Example 22), we fix a strict partial order $<$ on the concepts that occur in $\mathcal{T}$ such that $\mathcal{T} \models C \equiv D$ implies $C < D$ or $D < C$ and, conversely, $C < D$ implies $\mathcal{T} \not\models C \equiv D$. Now, the minimization of a rooted CQ q that is consistent with $\mathcal{T}$ proceeds in three phases:

1. maximally extend q w.r.t. concept atoms: add $C(t)$ to q whenever $\mathcal{T}, \mathcal{A}_q \models C(t)$ and $t \in \mathsf{term}(q)$;
2. exhaustively remove redundant role atoms, in any order;
3. exhaustively remove redundant concept atoms; when two atoms $C(t)$ and $D(t)$ are both redundant, drop $D(t)$ rather than $C(t)$ whenever (i) $\mathcal{T} \models C \sqsubseteq D$, but not $\mathcal{T} \models D \sqsubseteq C$ or (ii) $\mathcal{T} \models C \equiv D$ and $C < D$.

It can be shown that applying this minimization strategy to a rooted CQ again yields a rooted CQ (or a CQ without any atoms, a special case that is easy to handle). It is easy to see that any query produced according to this strategy is $\mathcal{T}$-minimal w.r.t. set inclusion (thus Theorem 23 applies).

Example 24. *Consider $\mathcal{T} = \{A \equiv \exists r, \exists r^- \sqsubseteq B\}$, the rooted CQ $\exists y\, r(x,y) \wedge B(y)$, and $<$ with $A < \exists r$. In Step 1, we add the atoms $A(x), \exists r(x), \exists r^-(y)$. In Step 2, we drop $r(x,y)$, since this preserves equivalence under $\equiv_\mathcal{T}$. In Step 3, we remove $B(y)$ and $\exists r^-(y)$ since this again preserves equivalence under $\equiv_\mathcal{T}$. We also remove $\exists r(x)$ since $\exists r(x) \equiv_\mathcal{T} A(x)$ but $A < \exists r$. We thus obtain $A(x)$.*

We can show that with the proposed strategy, we achieve uniqueness of minimized queries as in the relational case.

Theorem 25. *Let $\mathcal{T}$ be a DL-Lite$_{\mathsf{core}}$-TBox, q_1, q_2 rooted CQs such that $q_1 \equiv_\mathcal{T} q_2$ and q_1, q_2 are consistent with $\mathcal{T}$, and $\widehat{q}_1, \widehat{q}_2$ CQs obtained from q_1 and q_2 by applying the minimization strategy. Then $\widehat{q}_1$ and $\widehat{q}_2$ are isomorphic.*

To see why we need rooted CQs, consider the unrooted CQs $q_1 = \exists x\, A(x)$ and $q_2 = \exists x\, B(x)$, the TBox $\mathcal{T} = \{A \equiv \exists r, B \equiv \exists r^-\}$, and $<$ such that $A < \exists r$ and $B < \exists r^-$. Then $q_1 \equiv_\mathcal{T} q_2$, $\widehat{q}_1 = q_1$, $\widehat{q}_2 = q_2$, but q_1 and q_2 are not isomorphic.

We write q^{r} to denote the CQ obtained from q by dropping all concept atoms, i.e., only the role atoms are kept. The queries produced by the strategy are optimal regarding their relational structure in following sense.

Theorem 26. *Let $\mathcal{T}$ be a DL-Lite$_{\mathsf{core}}$-TBox, q_1, q_2 rooted CQs with $q_1 \equiv_\mathcal{T} q_2$ that are consistent with $\mathcal{T}$, and let $\widehat{q}_1$ be obtained from q_1 by the minimization strategy. Then $\widehat{q}_1^{\mathsf{r}}$ is isomorphic to a subquery of q_2.*

As a consequence, the minimization strategy yields an acyclic CQ iff *any* acyclic CQ is equivalent to q w.r.t. $\mathcal{T}$. The same holds for queries of bounded treewidth or with other desirable relational properties.

Optimization in $\mathcal{EL}$

We show that, in $\mathcal{EL}$, there is a minimization strategy which is guaranteed to produce an acyclic CQ whenever the original CQ is equivalent to *any* acyclic CQ.

Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox. We introduce a slightly different notion of fork elimination than was used in Section 4. A t_1, t_2-*fork* in a CQ q consists of atoms $r(t_1, t), r(t_2, t) \in q$ or $r(t, t_1), r(t, t_2) \in q$ such that $t_1 \neq t_2$. A t_1, t_2-fork is *eliminated* by identifying t_1 and t_2. Starting with an input CQ q, our minimization strategy is to exhaustively eliminate forks such that $\mathcal{T}$-equivalence is preserved, in any order. By the following lemma and as any given CQ admits only finitely many consecutive fork eliminations, we are guaranteed to find an acyclic CQ if there is one.

Lemma 27. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and q a CQ that is not acyclic, but such that $q \equiv_\mathcal{T} p$ for some acyclic CQ p. Then q contains a fork whose elimination yields q' with $q \equiv_\mathcal{T} q'$.*

In the long version, we show that, in a second step, we can further minimize the query such that it has a minimum number of variables among all equivalent queries while preserving acyclicity. This is at the expense of introducing query atoms $C(t)$ with C a subconcept of the TBox.

8 Related Work in Database Theory

Query containment is a central notion in relational database theory, due to its importance in query optimization (cf. (Abiteboul, Hull, and Vianu 1995) and references therein). Of greater relevance to the present paper is work on containment of datalog programs. Formally, a datalog program P is contained in another program P' with respect to a target predicate q if for every input database D over the (shared) input signature, the output tuples for q w.r.t. (P, D) are a subset of those obtained for (P', D). This is broadly similar to our setting, with the datalog rules playing the role of the TBox. Datalog containment was studied in the context of optimizing datalog programs by removing atoms or rules while preserving equivalence. Because equivalence of datalog programs is undecidable (Shmueli 1987), the emphasis is on sound, but incomplete approaches (Sagiv 1987). It is relevant to note that containment of *monadic* datalog programs is decidable (Cosmadakis et al. 1988), which might conceivably be used to obtain an alternative proof of decidability of some of our problems. Note, however, that any such proof would be rather involved (due to the necessity of compiling away existential restrictions), and would not yield optimal complexity results since the best-known upper bound for monadic datalog containment is 2-ExpTime (Cosmadakis et al. 1988). Finally, we note that in both the relational and deductive database settings, semantic query optimization (Chakravarthy, Grant, and Minker 1990) is based on a variant of containment that is relativized to the class of databases satisfying a given set of integrity constraints. This can be compared to our setting in which both the restricted ABox signature and the required consistency with the TBox act as constraints on the possible ABoxes.

9 Future Work

As noted earlier, using the ABox signature during optimization can lead to shorter equivalent queries. Unfortunately, the guarantees we obtained in Section 7 regarding the outcome of query minimization no longer hold in the presence of an ABox signature or two TBoxes. Thus, a relevant question for future research is the design of minimization strategies for the generalized versions of query containment.

It would be worthwhile to extend our investigation to other query languages, like unions of conjunctive queries (UCQs) or positive existential queries (PEQs). This enables further applications, such as verifying query rewritings that allow to implement CQ answering in the presence of DL TBoxes using relational database systems (Calvanese et al. 2007): given a CQ q and TBox $\mathcal{T}$ and a rewriting of q and $\mathcal{T}$ into a UCQ or PEQ q', check whether $(q, \mathcal{T}) \equiv (q', \emptyset)$, i.e., whether q' is an 'FO-rewriting' of q relative to $\mathcal{T}$.

Acknowledgements Carsten Lutz was supported by the DFG SFB/TR 8 "Spatial Cognition".

References

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

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.; McGuiness, D. L.; Nardi, D.; and Patel-Schneider, P., eds. 2003. *The Description Logic Handbook*. Cambridge University Press.

Baader, F.; Bienvenu, M.; Lutz, C.; and Wolter, F. 2010. Query and predicate emptiness in description logics. In *Proc. of KR*, 192–202.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*, 364–369.

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.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. The MASTRO system for ontology-based data access. *Semantic Web* 2(1):43–53.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1998. On the decidability of query containment under constraints. In *Proc. of PODS*, 149–158.

Calvanese, D.; Ortiz, M.; and Simkus, M. 2011. Containment of regular path queries under description logic constraints. In *Proc. of IJCAI*, 805–812.

Chakravarthy, U. S.; Grant, J.; and Minker, J. 1990. Logic-based approach to semantic query optimization. *ACM Transactions on Database Systems* 15(2):162–207.

Chandra, A. K., and Merlin, P. M. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Proc. of STOC*, 77–90.

Cosmadakis, S. S.; Gaifman, H.; Kanellakis, P. C.; and Vardi, M. Y. 1988. Decidable optimization problems for database logic programs (preliminary report). In *Proc.of STOC*, 477–490.

Glimm, B.; Lutz, C.; Horrocks, I.; and Sattler, U. 2008. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *J. of Artificial Intelligence Research* 31:157–204.

Gonçalves, R. S.; Parsia, B.; and Sattler, U. 2011. Analysing the evolution of the NCI thesaurus. In *Proc. of CBMS*, 1–6.

Hemachandra, L. A. 1987. The strong exponential hierarchy collapses. In *Proc. of STOC*, 110–122.

Hustadt, U.; Motik, B.; and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proc. of IJCAI*, 466–471.

IHTSDO. 2008. *SNOMED Clinical Terms User Guide*. The International Health Terminology Standards Development Organisation (IHTSDO). Available from http://www.ihtsdo.org/publications/snomed-docs/.

Jimnez-Ruiz, E.; Grau, B. C.; Horrocks, I.; and Llavori, R. B. 2011. Supporting concurrent ontology development: Framework, algorithms and tool. *Data & Knowledge Engineering* 70(1):146–164.

Klein, M. C. A.; Fensel, D.; Kiryakov, A.; and Ognyanov, D. 2002. Ontology versioning and change detection on the web. In *Proc. of EKAW*, 247–259.

Konev, B.; Ludwig, M.; Walther, D.; and Wolter, F. 2011. The logical diff for the lightweight description logic $\mathcal{EL}$. Technical report, University of Liverpool, http://www.liv.ac.uk/~frank/publ/.

Konev, B.; Walther, D.; and Wolter, F. 2008. The logical difference problem for description logic terminologies. In *Proc. of IJCAR*, 259–274.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2010. The combined approach to query answering in DL-Lite. In *Proc. of KR*, 247–257.

Liu, H.; Lutz, C.; and Milicic, M. 2008. The projection problem for $\mathcal{EL}$ actions. In *Proc. of DL*.

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.

Lutz, C. 2008. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR*, 179–193.

Noy, N. F., and Musen, M. A. 2002. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *Proc. of AAAI*, 744–750.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2011. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proc. of IJCAI*, 1039–1044.

Pérez-Urbina, H.; Horrocks, I.; and Motik, B. 2009. Efficient query answering for OWL 2. In *Proc. of ISWC*, 489–504.

Sagiv, Y., and Yannakakis, M. 1980. Equivalences among relational expressions with the union and difference operators. *J. of the ACM* 27(4):633–655.

Sagiv, Y. 1987. Optimizing datalog programs. In *Proc. of PODS*, 349–362.

Shmueli, O. 1987. Decidability and expressiveness of logic queries. In *Proc. of PODS*, 237–249.

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