First Order-Rewritability and Containment of Conjunctive Queries in Horn Description Logics

Meghyn Bienvenu, Peter Hansen, Carsten Lutz, Frank Wolter

First Order-Rewritability and Containment of Conjunctive Queries in Horn Description Logics

Meghyn Bienvenu
CNRS, Univ. Montpellier, Inria, France
meghyn@lirmm.fr

Peter Hansen and **Carsten Lutz**
University of Bremen, Germany
{hansen, clu}@informatik.uni-bremen.de

Frank Wolter
University of Liverpool, UK
frank@csc.liv.ac.uk

Abstract

We study FO-rewritability of conjunctive queries in the presence of ontologies formulated in a description logic between $\mathcal{EL}$ and Horn-$\mathcal{SHIF}$, along with related query containment problems. Apart from providing characterizations, we establish complexity results ranging from ExpTime via NExpTime to 2ExpTime, pointing out several interesting effects. In particular, FO-rewriting is more complex for conjunctive queries than for atomic queries when inverse roles are present, but not otherwise.

1 Introduction

When ontologies are used to enrich incomplete and heterogeneous data with a semantics and with background knowledge [Calvanese *et al.*, 2009; Kontchakov *et al.*, 2013; Bienvenu and Ortiz, 2015], efficient query anwering is a primary concern. Since classical database systems are unaware of ontologies and implementing new ontology-aware systems that can compete with these would be a huge effort, a main approach used today is *query rewriting*: the user query q and the ontology $\mathcal{O}$ are combined into a new query $q_{\mathcal{O}}$ that produces the same answers as q under $\mathcal{O}$ (over all inputs) and can be handed over to a database system for execution. Popular target languages for the query $q_{\mathcal{O}}$ include SQL and Datalog. In this paper, we concentrate on ontologies formulated in description logics (DLs) and on rewritability into SQL, which we equate with first-order logic (FO).

FO-rewritability in the context of query answering under DL ontologies was first studied in [Calvanese *et al.*, 2007]. Since FO-rewritings are not guaranteed to exist when ontologies are formulated in traditional DLs, the authors introduce the DL-Lite family of DLs specifically for the purpose of ontology-aware query answering using SQL database systems; in fact, the expressive power of DL-Lite is seriously restricted, in this way enabling existence guarantees for FO-rewritings. While DL-Lite is a successful family of DLs, there are many applications that require DLs with greater expressive power. The potential non-existence of FO-rewritings in this case is not necessarily a problem in practical applications. In fact, ontologies emerging from such applications typically use the available expressive means in a harmless way in the sense that efficient reasoning is often possible despite high worst-case complexity. One might thus hope that, in practice, FO-rewritings can often be constructed also beyond DL-Lite.

This hope was confirmed in [Bienvenu *et al.*, 2013; Hansen *et al.*, 2015], which consider the case where ontologies are formulated in a DL of the $\mathcal{EL}$ family [Baader *et al.*, 2005] and queries are atomic queries (AQs) of the form $A(x)$. To describe the obtained results in more detail, let an ontology-mediated query (OMQ) be a triple $(\mathcal{T}, \Sigma, q)$ with $\mathcal{T}$ a description logic TBox (representing an ontology), Σ an ABox signature (the set of concept and role names that can occur in the data), and q an actual query. Note that $\mathcal{T}$ and q might use symbols that do not occur in Σ; in this way, the TBox enriches the vocabulary available for formulating q. We use $(\mathcal{L}, \mathcal{Q})$ to denote the OMQ language that consists of all OMQs where $\mathcal{T}$ is formulated in the description logic $\mathcal{L}$ and q in the query language $\mathcal{Q}$. In [Bienvenu *et al.*, 2013], FO-rewritability is characterized in terms of the existence of certain tree-shaped ABoxes, covering a range of OMQ languages between $(\mathcal{EL}, \text{AQ})$ and (Horn-$\mathcal{SHI}$, AQ). On the one hand, this characterization is used to clarify the complexity of deciding whether a given OMQ is FO-rewritable, which turns out to be ExpTime-complete. On the other hand, it provides the foundations for developing practically efficient and complete algorithms for computing FO-rewritings. The latter was explored further in [Hansen *et al.*, 2015], where a novel type of algorithm for computing FO-rewritings of OMQs from $(\mathcal{EL}, \text{AQ})$ is introduced, crucially relying on the previous results from [Bienvenu *et al.*, 2013]. Its evaluation shows excellent performance and confirms the hope that, in practice, FO-rewritings almost always exist. In fact, rewriting fails in only 285 out of 10989 test cases.

A limitation of the discussed results is that they concern only AQs while in many applications, the more expressive conjunctive queries (CQs) are required. The aim of the current paper is thus to study FO-rewritability of OMQ languages based on CQs, considering ontology languages between $\mathcal{EL}$ and Horn-$\mathcal{SHIF}$. In particular, we provide characterizations of FO-rewritability in the required OMQ languages that are inspired by those in [Bienvenu *et al.*, 2013] (replacing tree-shaped ABoxes with a more general form of ABox), and we analyze the complexity of FO-rewritability using an automata-based approach. While practically efficient algorithms are out of the scope of this article, we believe that our work also lays important ground for the subsequent development of such

algorithms. Our approach actually *does* allow the construction of rewritings, but it is not tailored towards doing that in a practically efficient way. It turns out that the studied FO-rewritability problems are closely related to OMQ containment problems as considered in [Bienvenu *et al.*, 2012; Bourhis and Lutz, 2016]. In fact, being able to decide OMQ containment allows us to concentrate on connected CQs when deciding FO-rewritability, which simplifies technicalities considerably. For this reason, we also study characterizations and the complexity of query containment in the OMQ languages considered.

Our main complexity results are that FO-rewritability and containment are EXPTIME-complete for OMQ languages between $(\mathcal{EL}, \text{AQ})$ and $(\mathcal{ELHF}_\bot, \text{CQ})$ and 2EXPTIME-complete for OMQ languages between $(\mathcal{ELI}, \text{CQ})$ and $(\text{Horn-}\mathcal{SHIF}, \text{CQ})$. The lower bound for containment applies already when both OMQs share the same TBox. Replacing AQs with CQs thus results in an increase of complexity by one exponential in the presence of inverse roles (indicated by $\mathcal{I}$), but not otherwise. Note that the effect that inverse roles can increase the complexity of querying-related problems was known from expressive DLs of the $\mathcal{ALC}$ family [Lutz, 2008], but it has not previously been observed for Horn-DLs such as $\mathcal{ELI}$ and Horn-$\mathcal{SHIF}$. While 2EXPTIME might appear to be very high complexity, we are fortunately also able to show that the runtime is double exponential only in the size of the actual queries (which tends to be very small) while it is only single exponential in the size of the ontologies. We also show that the complexity drops to NEXPTIME when we restrict our attention to *rooted* CQs, that is, CQs which contain at least one answer variable and are connected. Practically relevant queries are typically of this kind.

A slight modification of our lower bounds yields new lower bounds for monadic Datalog containment. In fact, we close an open problem from [Chaudhuri and Vardi, 1994] by showing that containment of a monadic Datalog program in a rooted CQ is CONEXPTIME-complete. We also improve the 2EXPTIME lower bound for containment of a monadic Datalog program in a CQ from [Benedikt *et al.*, 2012] by showing that it already applies when the arity of EDB relations is bounded by two, rule bodies are tree-shaped, and there are no constants (which in this case correspond to nominals); the existing construction cannot achieve the latter two conditions simultaneously.

Full proofs are provided at http://www.informatik.uni-bremen.de/tdki/research/papers.html.

Related work. Pragmatic approaches to OMQ rewriting beyond DL-Lite often consider Datalog as a target language [Rosati, 2007; Pérez-Urbina *et al.*, 2010; Eiter *et al.*, 2012; Kaminski *et al.*, 2014; Trivela *et al.*, 2015]. These approaches might produce a non-recursive (thus FO) rewriting if it exists, but there are no guarantees. FO-rewritability of OMQs based on expressive DLs is considered in [Bienvenu *et al.*, 2014], and based on existential rules in [Baget *et al.*, 2011]. A problem related to ours is whether *all* queries are FO-rewritable when combined with a given TBox [Lutz and Wolter, 2012; Civili and Rosati, 2015]. There are several related works in the area of Datalog; recall that a Datalog program is bounded if and only if it is FO-rewritable [Ajtai and Gurevich, 1994]. For monadic Datalog programs, boundedness is known to be decidable [Cosmadakis *et al.*, 1988] and 2EXPTIME-complete [Benedikt *et al.*, 2015]; containment is also 2EXPTIME-complete [Cosmadakis *et al.*, 1988; Benedikt *et al.*, 2012]. OMQs from $(\text{Horn-}\mathcal{SHI}, \text{CQ})$ can be translated to monadic Datalog with an exponential blowup, functional roles (indicated by $\mathcal{F}$) are not expressible.

2 Preliminaries and Basic Observations

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be disjoint and countably infinite sets of *concept* and *role names*. A *role* is a role name r or an *inverse role* r^-, with r a role name. A *Horn-$\mathcal{SHIF}$ concept inclusion (CI)* is of the form $L \sqsubseteq R$, where L and R are concepts defined by the syntax rules

$$R, R' ::= \top \mid \bot \mid A \mid \neg A \mid R \sqcap R' \mid \neg L \sqcup R \mid \exists r.R \mid \forall r.R$$

$$L, L' ::= \top \mid \bot \mid A \mid L \sqcap L' \mid L \sqcup L' \mid \exists r.L$$

with A ranging over concept names and r over roles. In DLs, ontologies are formalized as TBoxes. A *Horn-$\mathcal{SHIF}$ TBox* $\mathcal{T}$ is a finite set of Horn-$\mathcal{SHIF}$ CIs, *functionality assertions* $\mathsf{func}(r)$, *transitivity assertions* $\mathsf{trans}(r)$, and *role inclusions (RIs)* $r \sqsubseteq s$, with r and s roles. It is standard to assume that functional roles are not transitive and neither are transitive roles included in them (directly or indirectly). We make the slighty stronger assumption that functional roles do not occur on the right-hand side of role inclusions at all. This assumption seems natural from a modeling perspective and mainly serves the purpose of simplifying constructions; all our results can be extended to the milder standard assumption. An *$\mathcal{ELIHF}_\bot$ TBox* is a *Horn-$\mathcal{SHIF}$ TBox* that contains neither transitivity assertions nor disjunctions in CIs, an *$\mathcal{ELI}$ TBox* is an $\mathcal{ELIHF}_\bot$ TBox that contains neither functionality assertions nor RIs, and an $\mathcal{ELHF}_\bot$ TBox is an $\mathcal{ELIHF}_\bot$ TBox that does not contain inverse roles.

An *ABox* is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ where A is a concept name, r a role name, and a, b individual names from a countably infinite set $\mathsf{N_I}$. We sometimes write $r^-(a, b)$ instead of $r(b, a)$ and use $\mathsf{Ind}(\mathcal{A})$ to denote the set of all individual names used in $\mathcal{A}$. A *signature* is a set of concept and role names. We will often assume that the ABox is formulated in a prescribed signature, which we then call an *ABox signature*. An ABox that only uses concept and role names from a signature Σ is called a *Σ-ABox*.

The semantics of DLs is given in terms of *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set (the *domain*) and $\cdot^\mathcal{I}$ is the *interpretation function*, assigning to each $A \in \mathsf{N_C}$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and to each $r \in \mathsf{N_R}$ a relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$ of a concept C in $\mathcal{I}$ is defined as usual, see [Baader *et al.*, 2003]. An interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a functionality assertion $\mathsf{func}(r)$ if $r^\mathcal{I}$ is a partial function, a transitivity assertion $\mathsf{trans}(r)$ if $r^\mathcal{I}$ is transitive, an RI $r \sqsubseteq s$ 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 $(a, b) \in r^\mathcal{I}$. We say that $\mathcal{I}$ is a *model of* a TBox or an ABox if it satisfies all inclusions and assertions in it. An ABox $\mathcal{A}$ is *consistent* with a TBox $\mathcal{T}$ if $\mathcal{A}$ and $\mathcal{T}$ have a common model. If α is a CI, RI, or functionality assertion, we write $\mathcal{T} \models \alpha$ if all models of $\mathcal{T}$ satisfy α.

A *conjunctive query (CQ)* takes the form $q = \exists \mathbf{x}\, \varphi(\mathbf{x}, \mathbf{y})$ with $\mathbf{x}, \mathbf{y}$ tuples of variables and φ a conjunction of atoms of

the form $A(x)$ and $r(x,y)$ that uses only variables from $\mathbf{x} \cup \mathbf{y}$. The variables in $\mathbf{y}$ are called *answer variables*, the *arity* of q is the length of $\mathbf{y}$, and q is *Boolean* if it has arity zero. An *atomic query (AQ)* is a conjunctive query of the form $A(x)$. A *union of conjunctive queries (UCQ)* is a disjunction of CQs that share the same answer variables. Ontology-mediated queries (OMQs) and the notation $(\mathcal{L}, \mathcal{Q})$ for OMQ languages were already defined in the introduction. We generally assume that if a role name r occurs in q and $\mathcal{T} \models s \sqsubseteq r$, then $\mathsf{trans}(s) \notin \mathcal{T}$. This is common since allowing transitive roles in the query poses serious additional complications, which are outside the scope of this paper; see e.g. [Bienvenu *et al.*, 2010; Gottlob *et al.*, 2013].

Let $Q = (\mathcal{T}, \Sigma, q)$ be an OMQ, q of arity n, $\mathcal{A}$ a Σ-ABox and $\mathbf{a} \in \mathsf{Ind}(\mathcal{A})^n$. We write $\mathcal{A} \models Q(\mathbf{a})$ if $\mathcal{I} \models q(\mathbf{a})$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. In this case, $\mathbf{a}$ is a *certain answer* to Q on $\mathcal{A}$. We use $\mathsf{cert}(Q, \mathcal{A})$ to denote the set of all certain answers to Q on $\mathcal{A}$.

A first-order query (FOQ) is a first-order formula φ constructed from atoms $A(x)$, $r(x,y)$, and $x = y$; here, concept names are viewed as unary predicates, role names as binary predicates, and predicates of other arity, function symbols, and constant symbols are not permitted. We write $\varphi(\mathbf{x})$ to indicate that the free variables of φ are among $\mathbf{x}$ and call $\mathbf{x}$ the *answer variables* of φ. The number of answer variables is the *arity* of φ and φ is *Boolean* if it has arity zero. We use $\mathsf{ans}(\mathcal{I}, \varphi)$ to denote the set of answers to the FOQ φ on the interpretation $\mathcal{I}$; that is, if φ is n-ary, then $\mathsf{ans}(\mathcal{I}, \varphi)$ contains all tuples $\mathbf{d} \in (\Delta^{\mathcal{I}})^n$ with $\mathcal{I} \models \varphi(\mathbf{d})$. To bridge the gap between certain answers and answers to FOQs, we sometime view an ABox $\mathcal{A}$ as an interpretation $\mathcal{I}_{\mathcal{A}}$, defined in the obvious way.

For any syntactic object O (such as a TBox, a query, an OMQ), we use $|O|$ to denote the *size* of O, that is, the number of symbols needed to write it (concept and role names counted as a single symbol).

Definition 1 (FO-rewriting)**.** An FOQ φ is an *FO-rewriting* of an OMQ $Q = (\mathcal{T}, \Sigma, q)$ if $\mathsf{cert}(Q, \mathcal{A}) = \mathsf{ans}(\mathcal{I}_{\mathcal{A}}, \varphi)$ for all Σ-ABoxes $\mathcal{A}$ that are consistent with $\mathcal{T}$. If there is such a φ, then Q is *FO-rewritable*.

Example 2. *(1) Let $Q_0 = (\mathcal{T}_0, \Sigma_0, q_0(x,y))$, where $\mathcal{T}_0 = \{\exists r.A \sqsubseteq A, B \sqsubseteq \forall r.A\}$, $\Sigma_0 = \{r, A, B\}$ and $q_0(x,y) = B(x) \wedge r(x,y) \wedge A(y)$. Then $\varphi_0(x,y) = B(x) \wedge r(x,y)$ is an FO-rewriting of Q_0.*

We will see in Example 10 that the query Q_A obtained from Q_0 by replacing $q_0(x,y)$ with the AQ $A(x)$ is not FO-rewritable (due to the unbounded propagation of A via r-edges by $\mathcal{T}_0$). Thus, an FO-rewritable OMQ can give raise to AQ 'subqueries' that are not FO-rewritable.

(2) Let $Q_1 = (\mathcal{T}_1, \Sigma_1, q_1(x))$, where $\mathcal{T}_1 = \{\exists r.\exists r.A \sqsubseteq \exists r.A\}$, $\Sigma_1 = \{r, A\}$, and $q_1(x) = \exists y(r(x,y) \wedge A(y))$. Then Q_1 is not FO-rewritable (see again Example 10), but all AQ subqueries that Q_1 gives raise to are FO-rewritable.

The main reasoning problem studied in this paper is to decide whether a given OMQ $Q = (\mathcal{T}, \Sigma, q)$ is FO-rewritable. We assume without loss of generality that every symbol in Σ occurs in $\mathcal{T}$ or in q. We obtain different versions of this problem by varying the OMQ language used. Note that we have defined FO-rewritability relative to ABoxes that are consistent with the TBox. It is thus important for the user to know whether that is the case. Therefore, we also consider FO-rewritability of ABox inconsistency. More precisely, we say that *ABox inconsistency is FO-rewritable* relative to a TBox $\mathcal{T}$ and ABox signature Σ if there is a Boolean FOQ φ such that for every Σ-ABox $\mathcal{A}$, $\mathcal{A}$ is inconsistent with $\mathcal{T}$ iff $\mathcal{I}_{\mathcal{A}} \models \varphi()$.

Apart from FO-rewritability questions, we will also study OMQ containment. Let $Q_i = (\mathcal{T}_i, \Sigma, q_i)$ be two OMQs over the same ABox signature. We say that Q_1 *is contained in* Q_2, in symbols $Q_1 \subseteq Q_2$, if $\mathsf{cert}(Q_1, \mathcal{A}) \subseteq \mathsf{cert}(Q_2, \mathcal{A})$ holds for all Σ-ABoxes $\mathcal{A}$ that are consistent with $\mathcal{T}_1$ and $\mathcal{T}_2$.

We now make two basic observations that we use in an essential way in the remaining paper. We first observe that it suffices to concentrate on $\mathcal{ELIHF}_\bot$ TBoxes $\mathcal{T}$ in *normal form*, that is, all CIs are of one of the forms $A \sqsubseteq \bot, A \sqsubseteq \exists r.B, \top \sqsubseteq A, B_1 \sqcap B_2 \sqsubseteq A, \exists r.B \sqsubseteq A$ with A, B, B_1, B_2 concept names and r a role. We use $\mathsf{sig}(\mathcal{T})$ to denote the concept and role names that occur in $\mathcal{T}$.

Proposition 3. *Given a Horn-$\mathcal{SHIF}$ (resp. $\mathcal{ELHF}_\bot$) TBox $\mathcal{T}_1$ and ABox signature Σ, one can construct in polynomial time an $\mathcal{ELIHF}_\bot$ (resp. $\mathcal{ELHF}_\bot$) TBox $\mathcal{T}_2$ in normal form such that for every Σ-ABox $\mathcal{A}$,*

1. *$\mathcal{A}$ is consistent with $\mathcal{T}_1$ iff $\mathcal{A}$ is consistent with $\mathcal{T}_2$;*
2. *if $\mathcal{A}$ is consistent with $\mathcal{T}_1$, then for any CQ q that does not use symbols from $\mathsf{sig}(\mathcal{T}_2) \setminus \mathsf{sig}(\mathcal{T}_1)$, we have $\mathsf{cert}(Q_1, \mathcal{A}) = \mathsf{cert}(Q_2, \mathcal{A})$ where $Q_i = (\mathcal{T}_i, \Sigma, q)$.*

Theorem 3 yields polytime reductions of FO-rewritability in (Horn-$\mathcal{SHIF}, \mathcal{Q}$) to FO-rewritability in ($\mathcal{ELIHF}_\bot, \mathcal{Q}$) for any query language $\mathcal{Q}$, and likewise for OMQ containment and FO-rewritability of ABox inconsistency. It also tells us that, when working with $\mathcal{ELHF}_\bot$ TBoxes, we can assume normal form. Note that transitioning from (Horn-$\mathcal{SHF}, \mathcal{Q}$) to ($\mathcal{ELHF}_\bot, \mathcal{Q}$) is not as easy as in the case with inverse roles since universal restrictions on the right-hand side of concept inclusions cannot easily be eliminated; for this reason, we do not consider (Horn-$\mathcal{SHF}, \mathcal{Q}$). From now on, we work with TBoxes formulated in $\mathcal{ELIHF}_\bot$ or $\mathcal{ELHF}_\bot$ and assume without further notice that they are in normal form.

Our second observation is that, when deciding FO-rewritability, we can restrict our attention to connected queries provided that we have a way of deciding containment (for potentially disconnected queries). We use conCQ to denote the class of all connected CQs.

Theorem 4. *Let $\mathcal{L} \in \{\mathcal{ELIHF}_\bot, \mathcal{ELHF}_\bot\}$. Then FO-rewritability in $(\mathcal{L}, CQ)$ can be solved in polynomial time when there is access to oracles for containment in $(\mathcal{L}, \mathcal{Q})$ and for FO-rewritability in $(\mathcal{L}, conCQ)$.*

To prove Theorem 4, we observe that FO-rewritability of an OMQ $Q = (\mathcal{T}, \Sigma, q)$ is equivalent to FO-rewritability of all OMQs $Q = (\mathcal{T}, \Sigma, q_c)$ with q_c a maximal connected component of q, excluding certain redundant such components (which can be identified using containment). Backed by Theorem 4, we generally assume connected queries when studying FO-rewritability, which allows to avoid unpleasant technical complications and is a main reason for studying FO-rewritability and containment in the same paper.

3 Main Results

In this section, we summarize the main results established in this paper. We start with the following theorem.

Theorem 5. *FO-rewritability and containment are*

1. 2EXPTIME*-complete for any OMQ language between* $(\mathcal{ELI}, CQ)$ *and* $(\textit{Horn-}\mathcal{SHIF}, CQ)$*, and*
2. EXPTIME*-complete for any OMQ language between* $(\mathcal{EL}, AQ)$ *and* $(\mathcal{ELHF}_\bot, CQ)$.

Moreover, given an OMQ from $(\textit{Horn-}\mathcal{SHIF}, CQ)$ *that is FO-rewritable, one can effectively construct a UCQ-rewriting.*

Like the subsequent results, Theorem 5 illustrates the strong relationship between FO-rewritability and containment. Note that inverse roles increase the complexity of both reasoning tasks. We stress that this increase takes place only when the actual queries are conjunctive queries, since FO-rewritability for OMQ languages with inverse roles and atomic queries is in EXPTIME [Bienvenu *et al.*, 2013].

The 2EXPTIME-completeness result stated in Point 1 of Theorem 5 might look discouraging. However, the situation is not quite as bad as it seems. To show this, we state the upper bound underlying Point 1 of Theorem 5 a bit more carefully.

Theorem 6. *Given OMQs* $Q_i = (\mathcal{T}_i, \Sigma_i, q_i)$*,* $i \in \{1, 2\}$*, from* $(\textit{Horn-}\mathcal{SHIF}, CQ)$*, it can be decided*

1. *in time* $2^{2^{p(|q_1| + \log(|\mathcal{T}_1|))}}$ *whether* Q_1 *is FO-rewritable and*
2. *in time* $2^{2^{p(|q_1| + |q_2| + \log(|\mathcal{T}_1| + |\mathcal{T}_2|))}}$ *whether* $Q_1 \subseteq Q_2$*,*

for some polynomial p*.*

Note that the runtime is double exponential only in the size of the actual queries q_1 and q_2, while it is only single exponential in the size of the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$. This is good news since the size of q_1 and q_2 is typically very small compared to the sizes of $\mathcal{T}_1$ and $\mathcal{T}_2$. For this reason, it can even be reasonable to assume that the sizes of q_1 and q_2 are constant, in the same way in which the size of the query is assumed to be constant in data complexity. Note that, under this assumption, Theorem 6 yields EXPTIME upper bounds.

One other way to relativize the seemingly very high complexity stated in Point 1 of Theorem 5 is to observe that the lower bound proofs require the actual query to be Boolean or disconnected. In practical applications, though, typical queries are connected and have at least one answer variable. We call such CQs *rooted* and use rCQ to denote the class of all rooted CQs. Our last main result states that, when we restrict our attention to rooted CQs, then the complexity drops to CONEXPTIME.

Theorem 7. *FO-rewritability and containment are* CONEXPTIME*-complete in any OMQ language between* $(\mathcal{ELI}, rCQ)$ *and* $(\textit{Horn-}\mathcal{SHIF}, rCQ)$.

4 Semantic Characterization

The upper bounds stated in Theorems 5 and 6 are established in two steps. We first give characterizations of FO-rewritability in terms of the existence of certain (almost) tree-shaped ABoxes, and then utilize this characterization to design decision procedures based on alternating tree automata. The semantic characterizations are of independent interest.

An ABox $\mathcal{A}$ is *tree-shaped* if the undirected graph with nodes $\mathsf{Ind}(\mathcal{A})$ and edges $\{\{a, b\} \mid r(a, b) \in \mathcal{A}\}$ is acyclic and connected and $r(a, b) \in \mathcal{A}$ implies that (i) $s(a, b) \notin \mathcal{A}$ for all $s \neq r$ and (ii) $s(b, a) \notin \mathcal{A}$ for all role names s. For tree-shaped ABoxes $\mathcal{A}$, we often distinguish an individual used as the root, denoted with $\rho_\mathcal{A}$. $\mathcal{A}$ is *ditree-shaped* if the directed graph with nodes $\mathsf{Ind}(\mathcal{A})$ and edges $\{(a, b) \mid r(a, b) \in \mathcal{A}\}$ is a tree and $r(a, b) \in \mathcal{A}$ implies (i) and (ii). The (unique) root of a ditree-shaped ABox $\mathcal{A}$ is also denoted with $\rho_\mathcal{A}$.

An ABox $\mathcal{A}$ is a *pseudo tree* if it is the union of ABoxes $\mathcal{A}_0, \dots, \mathcal{A}_k$ that satisfy the following conditions:

1. $\mathcal{A}_1, \dots, \mathcal{A}_k$ are tree-shaped;
2. $k \leq |\mathsf{Ind}(\mathcal{A}_0)|$;
3. $\mathcal{A}_i \cap \mathcal{A}_0 = \{\rho_{\mathcal{A}_i}\}$ and $\mathsf{Ind}(\mathcal{A}_i) \cap \mathsf{Ind}(\mathcal{A}_j) = \emptyset$, for $1 \leq i < j \leq k$.

We call $\mathcal{A}_0$ the *core* of $\mathcal{A}$ and $\mathcal{A}_1, \dots, \mathcal{A}_k$ the *trees* of $\mathcal{A}$. The *width* of $\mathcal{A}$ is $|\mathsf{Ind}(\mathcal{A}_0)|$, its *depth* is the depth of the deepest tree of $\mathcal{A}$, and its *outdegree* is the maximum outdegree of the ABoxes $\mathcal{A}_1, \dots, \mathcal{A}_k$. For a pseudo tree ABox $\mathcal{A}$ and $\ell \geq 0$, we write $\mathcal{A}|_{\leq \ell}$ to denote the restriction of $\mathcal{A}$ to the individuals whose minimal distance from a core individual is at most ℓ, and analogously for $\mathcal{A}|_{>\ell}$. A *pseudo ditree ABox* is defined analogously to a pseudo tree ABox, except that $\mathcal{A}_1, \dots, \mathcal{A}_k$ must be ditree-shaped.

When studying FO-rewritability and containment, we can restrict our attention to pseudo tree ABoxes, and even to pseudo ditree ABoxes when the TBox does not contain inverse roles. The following statement makes this precise for the case of containment. Its proof uses unraveling and compactness.

Proposition 8. *Let* $Q_i = (\mathcal{T}_i, \Sigma, q_i)$*,* $i \in \{1, 2\}$*, be OMQs from* $(\mathcal{ELIHF}_\bot, CQ)$*. Then* $Q_1 \not\subseteq Q_2$ *iff there is a pseudo tree* Σ*-ABox* $\mathcal{A}$ *of outdegree at most* $|\mathcal{T}_1|$ *and width at most* $|q_1|$ *that is consistent with both* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *and a tuple* $\mathbf{a}$ *from the core of* $\mathcal{A}$ *such that* $\mathcal{A} \models Q_1(\mathbf{a})$ *and* $\mathcal{A} \not\models Q_2(\mathbf{a})$.

If Q_1, Q_2 *are from* $(\mathcal{ELHF}_\bot, CQ)$*, then we can find a pseudo ditree ABox with these properties.*

We now establish a first version of the announced characterizations of FO-rewritability. Like Proposition 8, they are based on pseudo tree ABoxes.

Theorem 9. *Let* $Q = (\mathcal{T}, \Sigma, q)$ *be an OMQ from* $(\mathcal{ELIHF}_\bot, conCQ)$*. If the arity of* q *is at least one, then the following conditions are equivalent:*

1. Q *is FO-rewritable;*
2. *there is a* $k \geq 0$ *such that for all pseudo tree* Σ*-ABoxes* $\mathcal{A}$ *that are consistent with* $\mathcal{T}$ *and of outdegree at most* $|\mathcal{T}|$ *and width at most* $|q|$*: if* $\mathcal{A} \models Q(\mathbf{a})$ *with* $\mathbf{a}$ *from the core of* $\mathcal{A}$*, then* $\mathcal{A}|_{\leq k} \models Q(\mathbf{a})$*;*

If q *is Boolean, this equivalence holds with (2.) replaced by*

2′. *there is a* $k \geq 0$ *such that for all pseudo tree* Σ*-ABoxes* $\mathcal{A}$ *that are consistent with* $\mathcal{T}$ *and of outdegree at most* $|\mathcal{T}|$ *and of width at most* $|q|$*: if* $\mathcal{A} \models Q$*, then* $\mathcal{A}|_{>0} \models Q$ *or* $\mathcal{A}|_{\leq k} \models Q$.

If Q *is from* $(\mathcal{ELHF}_\bot, conCQ)$*, then the above equivalences hold also when pseudo tree* Σ*-ABoxes are replaced with pseudo ditree* Σ*-ABoxes.*

The proof of Proposition 8 gives a good intuition of why FO-rewritability can be characterized in terms of ABoxes that are pseudo trees. In fact, the proof of "2 $\Rightarrow$ 1" of Theorem 9 is similar to the proof of Proposition 8. The proof of "1 $\Rightarrow$ 2" uses locality arguments in the form of Ehrenfeucht-Fraïssé games. The following examples further illustrate Theorem 9.

Example 10. *(1) Non FO-rewritability of the OMQs Q_A and Q_1 from Example 2 is shown by refuting Condition 2 in Theorem 9: let $\mathcal{A}_k = \{r(a_0, a_1), \dots, r(a_k, a_{k+1}), A(a_{k+1})\}$, for all $k \geq 0$. Then $\mathcal{A}_k \models Q(a_0)$ but $\mathcal{A}_k|_{\leq k} \not\models Q(a_0)$ for $Q \in \{Q_A, Q_1\}$.*

(2) Theorem 9 only holds for connected CQs: consider $Q_2 = (\mathcal{T}_2, \Sigma_2, q_2)$, where $\mathcal{T}_2$ is the empty TBox, $\Sigma_2 = \{A, B\}$, and $q_2 = \exists x \exists y (A(x) \wedge B(y))$. Q_2 is FO-rewritable (q_2 itself is a rewriting), but Condition $2'$ does not hold: for $\mathcal{B}_k = \{A(a_0), R(a_0, a_1, \dots, R(a_k, a_{k+1}), B(a_{k+1})\}$ we have $\mathcal{B}_k \models Q_2$ but $\mathcal{B}_k|_{>0} \not\models Q_2$ and $\mathcal{B}_k|_{\leq k} \not\models Q_2$.

(3) The modification $2'$ of Condition 2 is needed to characterize FO-rewritability of Boolean OMQs: obtain Q_B from Q_2 by replacing q_2 with $\exists x B(x)$. Then Q_B is FO-rewritable, but the ABoxes $\mathcal{B}_k$ show that Condition 2 does not hold.

Theorem 9 does not immediately suggest a decision procedure for FO-rewritability since there is no bound on the depth of the pseudo tree ABoxes $\mathcal{A}$ used. The next result establishes such a bound.

Theorem 11. *Let $\mathcal{T}$ be an $\mathcal{ELIHF}_\bot$ TBox. Then Theorem 9 still holds with the following modifications:*

1. *if q is not Boolean or $\mathcal{T}$ is an $\mathcal{ELHF}_\bot$ TBox, "there is a $k \geq 0$" is replaced with "for $k = |q| + 2^{4(|\mathcal{T}|+|q|)^2}$";*
2. *if q is Boolean, "there is a $k \geq 0$" is replaced with "for $k = |q| + 2^{4(|\mathcal{T}|+2^{|q|})^2}$".*

The proof of Theorem 11 uses a pumping argument based on derivations of concept names in the pumped ABox by $\mathcal{T}$. Due to the presence of inverse roles, this is not entirely trivial and uses what we call *transfer sequences*, describing the derivation history at a point of an ABox. Together with the proof of Theorem 9, Theorem 11 gives rise to an algorithm that constructs actual rewritings when they exist.

5 Constructing Automata

We show that Proposition 8 and Theorem 11 give rise to automata-based decision procedures for containment and FO-rewritability that establish the upper bounds stated in Theorems 5 and 6. By Theorem 4, it suffices to consider connected queries in the case of FO-rewritability. We now observe that we can further restrict our attention to Boolean queries. We use BCQ (resp. conBCQ) to denote the class of all Boolean CQs (resp. connected Boolean CQs).

Lemma 12. *Let $\mathcal{L} \in \{\mathcal{ELIHF}_\bot, \mathcal{ELHF}_\bot\}$. Then*

1. *FO-rewritability in $(\mathcal{L}, \text{conCQ})$ can be reduced in polytime to FO-rewritability in $(\mathcal{L}, \text{conBCQ})$;*
2. *Containment in $(\mathcal{L}, \text{CQ})$ can be reduced in polytime to containment in $(\mathcal{L}, \text{BCQ})$.*

The decision procedures rely on building automata that accept pseudo tree ABoxes which witness non-containment and non-FO-rewritability as stipulated by Proposition 8 and Theorem 11, respectively. We first have to encode pseudo tree ABoxes in a suitable way.

A *tree* is a non-empty (and potentially infinite) set $T \subseteq \mathbb{N}^*$ closed under prefixes. We say that T is m*-ary* if for every $x \in T$, the set $\{i \mid x \cdot i \in T\}$ is of cardinality at most m. For an alphabet Γ, a Γ*-labeled tree* is a pair (T, L) with T a tree and $L : T \to \Gamma$ a node labeling function. Let $Q = (\mathcal{T}, \Sigma, q)$ be an OMQ from $(\mathcal{ELIHF}_\bot, \text{conBCQ})$. We encode pseudo tree ABoxes of width at most $|q|$ and outdegree at most $|\mathcal{T}|$ by $(|\mathcal{T}| \cdot |q|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled trees, where Σ_ε is an alphabet used for labeling root nodes and Σ_N is for non-root nodes.

The alphabet Σ_ε consists of all Σ-ABoxes $\mathcal{A}$ such that $\mathsf{Ind}(\mathcal{A})$ only contains individual names from a fixed set $\mathsf{Ind}_{\mathsf{core}}$ of size $|q|$ and $\mathcal{A}$ satisfies all functionality statements in $\mathcal{T}$. The alphabet Σ_N consists of all subsets $\Theta \subseteq (\mathsf{N_C} \cap \Sigma) \uplus \{r, r^- \mid r \in \mathsf{N_R} \cap \Sigma\} \uplus \mathsf{Ind}_{\mathsf{core}}$ that contain exactly one (potentially inverse) role and at most one element of $\mathsf{Ind}_{\mathsf{core}}$. A $(|\mathcal{T}| \cdot |q|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled tree is *proper* if (i) the root node is labeled with a symbol from Σ_ε, (ii) each child of the root is labeled with a symbol from Σ_N that contains an element of $\mathsf{Ind}_{\mathsf{core}}$, (iii) every other non-root node is labeled with a symbol from Σ_N that contains no individual name, and (iv) every non-root node has at most $|q|$ successors and (v) for every $a \in \mathsf{Ind}_{\mathsf{core}}$, the root node has at most $|q|$ successors whose label includes a.

A proper $\Sigma_\varepsilon \cup \Sigma_N$-labeled tree (T, L) represents a pseudo tree ABox $\mathcal{A}_{(T,L)}$ whose individuals are those in the ABox $\mathcal{A}$ that labels the root of T plus all non-root nodes of T, and whose assertions are

$$\begin{aligned}
&\mathcal{A} \cup \{A(x) \mid A \in L(x)\} \\
&\cup \{r(b, x) \mid \{b, r\} \subseteq L(x)\} \cup \{r(x, b) \mid \{b, r^-\} \subseteq L(x)\} \\
&\cup \{r(x, y) \mid r \in L(y), y \text{ is a child of } x, L(x) \in \Sigma_N\} \\
&\cup \{r(y, x) \mid r^- \in L(y), y \text{ is a child of } x, L(x) \in \Sigma_N\}.
\end{aligned}$$

As the automaton model, we use two-way alternating parity automata on finite trees (TWAPAs). As usual, $L(\mathfrak{A})$ denotes the tree language accepted by the TWAPA $\mathfrak{A}$. Our central observation is the following.

Proposition 13. *For every OMQ $Q = (\mathcal{T}, \Sigma, q)$ from $(\mathcal{ELIHF}_\bot, \text{BCQ})$, there is a TWAPA*

1. *$\mathfrak{A}_Q$ that accepts a $(|\mathcal{T}| \cdot |q|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled tree (T, L) iff it is proper, $\mathcal{A}_{(T,L)}$ is consistent with $\mathcal{T}$, and $\mathcal{A}_{(T,L)} \models Q$;*

   $\mathfrak{A}_Q$ has at most $2^{p(|q| + \mathsf{log}(|\mathcal{T}|))}$ states, and at most $p(|q| + |\mathcal{T}|)$ states if $\mathcal{T}$ is an $\mathcal{ELHF}_\bot$ TBox, p a polynomial.
2. *$\mathfrak{A}_\mathcal{T}$ that accepts a $(|\mathcal{T}| \cdot |q|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled tree (T, L) iff it is proper and $\mathcal{A}_{(T,L)}$ is consistent with $\mathcal{T}$.*

   $\mathfrak{A}_\mathcal{T}$ has at most $p(|\mathcal{T}|)$ states, p a polynomial.

We can construct $\mathfrak{A}_Q$ and $\mathfrak{A}_\mathcal{T}$ in time polynomial in their size.

The construction of the automata in Proposition 13 uses forest decompositions of the CQ q as known for example from [Lutz, 2008]. The difference in automata size between $\mathcal{ELIHF}_\bot$ and $\mathcal{ELHF}_\bot$ is due to the different number of tree-shaped subqueries that can arise in these decompositions.

To decide $Q_1 \subseteq Q_2$ for OMQs $Q_i = (\mathcal{T}_i, \Sigma, q_i)$, $i \in \{1, 2\}$, from $(\mathcal{ELIHF}_\bot, \text{BCQ})$, by Proposition 8 it suffices to decide whether $L(\mathfrak{A}_{Q_1}) \cap L(\mathfrak{A}_{\mathcal{T}_2}) \subseteq L(\mathfrak{A}_{Q_2})$. Since this question can be polynomially reduced to a TWAPA emptiness check and the latter can be executed in time single exponential in the number of states, this yields the upper bounds for containment stated in Theorems 5 and 6.

To decide non-FO-rewritability of an OMQ $Q = (\mathcal{T}, \Sigma, q)$ from $(\mathcal{ELIHF}_\bot, \text{conBCQ})$, by Theorem 11 we need to decide whether there is a pseudo tree Σ-ABox $\mathcal{A}$ of outdegree at most $|\mathcal{T}|$ and width at most $|q|$ that is consistent with $\mathcal{T}$ and satisfies (i) $\mathcal{A} \models Q$, (ii) $\mathcal{A}|_{>0} \not\models Q$, and (iii) $\mathcal{A}|_{\leq k} \not\models Q$ where $k = |q| + 2^{4(|\mathcal{T}|+2^{|q|})^2}$. For consistency with $\mathcal{T}$ and for (i), we use the automaton $\mathfrak{A}_Q$ from Proposition 13. To achieve (ii) and (iii), we amend the tree alphabet $\Sigma_\varepsilon \cup \Sigma_n$ with additional labels that implement a counter which counts up to k and annotate each node in the tree with its depth (up to k). We then complement $\mathfrak{A}_Q$ (which for TWAPAs can be done in polynomial time), relativize the resulting automaton to all but the first level of the input ABox for (ii) and to the first k levels for (iii), and finally intersect all automata and check emptiness. This yields the upper bounds for FO-rewritability stated in Theorems 5 and 6.

As remarked in the introduction, apart from FO-rewritability of an OMQ $(\mathcal{T}, \Sigma, q)$ we should also be interested in FO-rewritability of ABox inconsistency relative to $\mathcal{T}$ and Σ. We close this section with noting that an upper bound for this problem can be obtained from Point 2 of Proposition 13 since TWAPAs can be complemented in polynomial time. A matching lower bound can be found in [Bienvenu *et al.*, 2013].

Theorem 14. *In $\mathcal{ELIHF}_\bot$, FO-rewritability of ABox inconsistency is* ExpTime*-complete.*

6 Rooted Queries and Lower Bounds

We first consider the case of rooted queries and establish the upper bound in Theorem 7.

Theorem 15. *FO-rewritability and containment in $(\mathcal{ELIHF}_\bot, rCQ)$ are in* coNExpTime*.*

Because of space limitations, we confine ourselves to a brief sketch, concentrating on FO-rewritability. By Point 1 of Theorem 11, deciding non-FO-rewritability of an OMQ $Q = (\mathcal{T}, \Sigma, q)$ from $(\mathcal{ELIHF}_\bot, \text{rCQ})$ comes down to checking the existence of a pseudo tree Σ-ABox $\mathcal{A}$ that is consistent with $\mathcal{T}$ and such that $\mathcal{A} \models Q(\mathbf{a})$ and $\mathcal{A}|_{\leq k} \not\models Q(\mathbf{a})$ for some tuple of individuals $\mathbf{a}$ from the core of $\mathcal{A}$, for some suitable k. Recall that $\mathcal{A} \models Q(\mathbf{a})$ if and only if there is a homomorphism h from q to the pseudo tree-shaped canonical model of $\mathcal{T}$ and $\mathcal{A}$ that takes the answer variables to $\mathbf{a}$. Because $\mathbf{a}$ is from the core of $\mathcal{A}$ and q is rooted, h can map existential variables in q only to individuals from $\mathcal{A}|_{|q|}$ and to the anonymous elements in the subtrees below them. To decide the existence of $\mathcal{A}$, we can thus guess $\mathcal{A}|_{|q|}$ together with sets of concept assertions about individuals in $\mathcal{A}|_{|q|}$ that can be inferred from $\mathcal{A}$ and $\mathcal{T}$, and from $\mathcal{A}|_{\leq k}$ and $\mathcal{T}$. We can then check whether there is a homomorphism h as described, without access to the full ABoxes $\mathcal{A}$ and $\mathcal{A}|_{\leq k}$. It remains to ensure that the guessed initial part $\mathcal{A}_{|q|}$ can be extended to $\mathcal{A}$ such that the entailed concept assertions are precisely those that were guessed, by attaching tree-shaped ABoxes to individuals on level $|q|$. This can be done by a mix of guessing and automata techniques.

We next establish the lower bounds stated in Theorems 5 and 7. For Theorem 5, we only prove a lower bound for Point 1 as the one in Point 2 follows from [Bienvenu *et al.*, 2013].

Theorem 16. *Containment and FO-rewritability are*

1. coNExpTime*-hard in $(\mathcal{ELI}, rCQ)$ and*
2. 2ExpTime*-hard in $(\mathcal{ELI}, CQ)$.*

The results for containment apply already when both OMQs share the same TBox.

Point 1 is proved by reduction of the problem of tiling a torus of exponential size, and Point 2 is proved by reduction of the word problem of exponentially space-bounded alternating Turing machines (ATMs). The proofs use queries similar to those introduced in [Lutz, 2008] to establish lower bounds on the complexity of query answering in the expressive OMQ languages $(\mathcal{ALCI}, \text{rCQ})$ and $(\mathcal{ALCI}, \text{CQ})$. A major difference to the proofs in [Lutz, 2008] is that we represent torus tilings / ATM computations in the ABox that witnesses non-containment or non-FO-rewritability, instead of in the 'anonymous part' of the model created by existential quantifiers.

The proof of Point 2 of Theorem 16 can be modified to yield new lower bounds for monadic Datalog containment. Recall that the rule body of a Datalog program is a CQ. *Tree-shapedness* of a CQ q is defined in the same way as for an ABox in Section 4, that is, q viewed as an undirected graph must be a tree without multi-edges.

Theorem 17. *For monadic Datalog programs which contain no EDB relations of arity larger than two and no constants, containment*

1. *in a rooted CQ is* coNExpTime*-hard;*
2. *in a CQ is* 2ExpTime*-hard, even when all rule bodies are tree-shaped.*

Point 1 closes an open problem from [Chaudhuri and Vardi, 1994], where a coNExpTime upper bound for containment of a monadic Datalog program in a rooted UCQ was proved and the lower bound was left open. Point 2 further improves a lower bound from [Benedikt *et al.*, 2012] which also does not rely on EDB relations of arity larger than two, but requires that rule bodies are not tree-shaped or constants are present (which, in this case, correspond to nominals in the DL world).

7 Conclusion

A natural next step for future work is to use the techniques developed here for devising practically efficient algorithms that construct actual rewritings, which was very successful in the AQ case [Hansen *et al.*, 2015].

An interesting open theoretical question is the complexity of FO-rewritability and containment for the OMQ languages considered in this paper in the special case when the ABox signature contains all concept and role names.

Acknowledgements. Bienvenu was supported by ANR project PAGODA (12-JS02-007-01), Hansen and Lutz by ERC grant 647289, Wolter by EPSRC UK grant EP/M012646/1.

References

[Ajtai and Gurevich, 1994] Miklós Ajtai and Yuri Gurevich. Datalog vs First-Order Logic. J. Comput. Syst. Sci.,49(3): 562–588, 1994.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *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 *Proc. of IJCAI*, pages 364–369, 2005.

[Baget *et al.*, 2011] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

[Benedikt *et al.*, 2012] Michael Benedikt, Pierre Bourhis, and Pierre Senellart. Monadic Datalog Containment. In *Proc. of ICALP*, pages 79–91, 2012.

[Benedikt *et al.*, 2015] Michael Benedikt, Balder ten Cate, Thomas Colcombet, and Michael Vanden Boom. The Complexity of Boundedness for Guarded Logics. In *Proc. of LICS*, pages 293–304, 2015.

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Proc. of Reasoning Web*, volume 9203 of *LNCS*, pages 218–307, 2015.

[Bienvenu *et al.*, 2010] Meghyn Bienvenu, Thomas Eiter, Carsten Lutz, Magdalena Ortiz, and Mantas Simkus. Query answering in the description logic S. In *Proc. of DL*, volume 573 of *CEUR-WS,* 2010.

[Bienvenu *et al.*, 2012] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query containment in description logics reconsidered. In *Proc of KR*, pages 221–231, 2012.

[Bienvenu *et al.*, 2013] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. First order-rewritability of atomic queries in Horn description logics. In *Proc. of IJCAI*, pages 754–760, 2013.

[Bienvenu *et al.*, 2014] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: a study through disjunctive datalog, CSP, and MMSNP. *Proc. of TODS*, 39, 2014.

[Bourhis and Lutz, 2016] Pierre Bourhis and Carsten Lutz. Containment in monadic disjunctive datalog, MMSNP, and expressive description logics. In *Proc. of KR*, 2016.

[Calvanese *et al.*, 2007] 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. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases: The DL-Lite approach. In *Proc. of Reasoning Web*, volume 5689 of *LNCS*, pages 255–356, 2009.

[Chaudhuri and Vardi, 1994] Surajit Chaudhuri and Moshe Y. Vardi. On the complexity of equivalence between recursive and nonrecursive datalog programs In *Proc. of PODS*, pages 107–116, 1994.

[Civili and Rosati, 2015] Cristina Civili and Riccardo Rosati. On the first-order rewritability of conjunctive queries over binary guarded existential rules. In *Proc. of CILC*, volume 1459 of *CEUR-WS*, pages 25–30, 2015.

[Cosmadakis *et al.*, 1988] Stavros S. Cosmadakis, Haim Gaifman, Paris C. Kanellakis, and Moshe Y. Vardi. Decidable optimization problems for database logic programs (preliminary report). In *Proc. of STOC*, pages 477–490, 1988.

[Eiter *et al.*, 2012] Thomas Eiter, Magdalena Ortiz, Mantas Simkus, Trung-Kien Tran, and Guohui Xiao. Query rewriting for Horn-SHIQ plus rules. In *Proc. of AAAI*, 2012.

[Gottlob *et al.*, 2013] Georg Gottlob, Andreas Pieris, and Lidia Tendera. Querying the guarded fragment with transitivity. In *Proc. of ICALP*, volume 7966 of *LNCS*, pages 287–298, 2013.

[Hansen *et al.*, 2015] Peter Hansen, Carsten Lutz, Inanç Seylan, and Frank Wolter. Efficient query rewriting in the description logic EL and beyond. In *Proc. of IJCAI*, pages 3034–3040, 2015.

[Kaminski *et al.*, 2014] Mark Kaminski, Yavor Nenov, and Bernardo Cuenca Grau. Computing datalog rewritings for disjunctive datalog programs and description logic ontologies. In *Proc. of RR*, pages 76–91, 2014.

[Kontchakov *et al.*, 2013] Roman Kontchakov, Mariano Rodriguez-Muro, and Michael Zakharyaschev. Ontology-based data access with databases: A short course. In *Proc. of Reasoning Web*, pages 194–229, 2013.

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

[Lutz, 2008] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR*, volume 5195 of *LNCS*, pages 179–193, 2008.

[Pérez-Urbina *et al.*, 2010] Héctor Pérez-Urbina, Boris Motik, and Ian Horrocks. Tractable query answering and rewriting under description logic constraints. *J. Applied Logic*, 8(2):186–209, 2010.

[Rosati, 2007] Riccardo Rosati. On conjunctive query answering in EL. In *Proc. of DL*, pages 451–458, 2007.

[Trivela *et al.*, 2015] Despoina Trivela, Giorgos Stoilos, Alexandros Chortaras, and Giorgos B. Stamou. Optimising resolution-based rewriting algorithms for OWL ontologies. *J. Web Sem.*, 33:30–49, 2015.