Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Ontology-Mediated Queries with Closed Predicates

Carsten Lutz[1], **İnanç Seylan**[1], and **Frank Wolter**[2]
[1]University of Bremen
{clu, seylan}@informatik.uni-bremen.de
[2]University of Liverpool, UK
wolter@liverpool.ac.uk

Abstract

In the context of ontology-based data access with description logics (DLs), we study ontology-mediated queries in which selected predicates can be closed (OMQCs). In particular, we contribute to the classification of the data complexity of such queries in several relevant DLs. For the case where only concept names can be closed, we tightly link this question to the complexity of surjective CSPs. When also role names can be closed, we show that a full complexity classification is equivalent to classifying the complexity of all problems in coNP, thus currently out of reach. We also identify a class of OMQCs based on ontologies formulated in DL-Lite$_R$ that are guaranteed to be tractable and even FO-rewritable.

1 Introduction

The aim of ontology-based data access (OBDA) is to facilitate querying of data that is significantly incomplete and heterogeneous. To account for the incompleteness, OBDA formalisms typically adopt the open world assumption (OWA). In some applications, though, there are selected parts of the data for which the closed world assumption (CWA) is more appropriate. As an example, consider a large-scale data integration application where parts of the data are extracted from the web and are significantly incomplete, thus justifying the OWA, while other parts of the data come from curated relational database systems and are known to be complete, thus justifying the CWA. Another example is given in [Lutz *et al.*, 2013], namely querying geo-databases such as OpenStreetMap in which the geo-data is typically assumed to be complete, thus justifying the CWA, while annotations are significantly incomplete and thus require the OWA.

In this paper, we consider OBDA formalisms where the ontology is formulated in a description logic (DL). Several approaches have been proposed to implement a partial CWA in OBDA and in other forms of DL reasoning [Calvanese *et al.*, 2007b; Donini *et al.*, 2002; Grimm and Motik, 2005; Motik and Rosati, 2010; Sengupta *et al.*, 2011]. A particularly simple and natural one is to distinguish between OWA predicates and CWA predicates (a predicate is a concept name or a role name) and to adopt the standard semantics from relational databases for the latter: the interpretation of CWA predicates is fixed to what is explicitly stated in the data while OWA predicates can be interpreted as any extension thereof [Lutz *et al.*, 2013]. This semantics generalizes both ABoxes as used in conventional OBDA (all predicates OWA) and so-called DBoxes (all predicates CWA) [Seylan *et al.*, 2009].

Closing predicates has a strong effect on the complexity of query answering. In this paper, we concentrate on data complexity, see [Ngo *et al.*, 2015] for an analysis of combined complexity in the presence of closed predicates. The (data) complexity of answering conjunctive queries (CQs) becomes coNP-hard already when ontologies are formulated in inexpressive DLs such as DL-Lite and $\mathcal{EL}$ [Franconi *et al.*, 2011] while CQ answering without closed predicates is in AC^0 for DL-Lite and in PTime for $\mathcal{EL}$ [Calvanese *et al.*, 2007a; Artale *et al.*, 2009; Hustadt *et al.*, 2005]. Since intractability comes so quickly, from a user's perspective it is not very helpful to analyze complexity on the level of logics, as in the complexity statements just made; instead, one would like to know whether adopting the CWA results in intractability for the *concrete ontology and query used in an application*. If it does not, there can be considerable benefit in adopting the CWA since it potentially results in additional (that is, more complete) answers and allows to use full first-order (FO) queries for the closed part of the vocabulary (which otherwise leads to undecidability). If adopting the CWA results in intractability, this is important information and the user can decide whether (s)he wants to resort to OWA as an approximation semantics or pay (in terms of complexity) for adopting the (partial) CWA.

Such a *non-uniform analysis* has been carried out in [Lutz and Wolter, 2012] and in [Bienvenu *et al.*, 2014] for classical OBDA (that is, no CWA predicates) with expressive DLs such as $\mathcal{ALC}$. The former reference aims to classify the complexity of ontologies, *quantifying over the actual query*: query answering for an ontology $\mathcal{O}$ is in PTime if every CQ can be answered in PTime w.r.t. $\mathcal{O}$ and it is coNP-hard if there is at least one Boolean CQ that is coNP-hard to answer w.r.t. $\mathcal{O}$. In the latter reference, an even more fine-grained approach is taken where the query is not quantified away. It aims to classify the complexity of *ontology-mediated queries (OMQs)*, that is, triples $(\mathcal{O}, \Sigma_{\mathsf{A}}, q)$ where $\mathcal{O}$ is an ontology, Σ_{A} a data vocabulary, and q an actual query. In both cases, a close connection to the complexity of (fixed-template) constraint satisfaction problems (CSPs) is identified, an active field of research that brings together algebra, graph theory, and logic [Feder and Vardi, 1993; Kun and Szegedy, 2009; Bulatov, 2011]. Such a connection is interesting for at least two reasons. First, it clarifies how difficult it is to attain a full complexity classification of relevant classes of ontologies/OMQs; in fact, there is a large body of literature on classifying the complexity of CSPs that revolves around the Feder-Vardi conjecture which states that every CSP is in PTIME or NP-hard [Feder and Vardi, 1993]. And second, it allows to transfer the technically deep results that have been obtained for CSPs in the last years to the world of OBDA.

For OBDA with closed predicates, the case of quantified queries has been analyzed in [Lutz *et al.*, 2013]. The main finding is that there are transparent and PTIME decidable syntactic conditions that separate the easy cases from the hard cases for ontologies formulated in DL-Lite and in $\mathcal{EL}$ (thus the complexity classification is much easier than for CSPs). However, it is also shown that the PTIME cases are exactly those where adopting the CWA does not result in returning additional answers and thus being able to use FO queries on the closed part of the vocabulary is the only benefit. This suggests that an analysis which quantifies over the queries is still too coarse to be practically useful. In the present paper, we therefore consider a complexity analysis on the level of *ontology-mediated queries with closed predicates (OMQCs)*, which are quadruples $(\mathcal{O}, \Sigma_A, \Sigma_C, q)$ where $\mathcal{O}$, Σ_A, and q are as above and $\Sigma_C \subseteq \Sigma_A$ is a set of closed predicates.

Our main finding is that while classifying the complexity of OMQs with expressive ontologies corresponds to classifying CSPs, classifying OMQCs is tightly linked to classifying *surjective CSPs*. The latter are defined exactly like standard CSPs (with fixed template) except that homomorphisms are required to be surjective. What might sound like a minor change actually makes complexity analyses dramatically more difficult. In fact, there are concrete surjective CSPs of size 6 whose complexity is not understood [Bodirsky *et al.*, 2012] while there are no such open cases for standard CSPs. Like standard CSPs, the complexity of surjective CSPs is currently subject to significant research activities [Bodirsky *et al.*, 2012; Chen, 2014]. Unlike for standard CSPs, though, we are not aware of dichotomy conjectures for surjective CSPs, this kind of question appears to be wide open.

Our connection to surjective CSPs concerns OMQCs where the ontology is formulated in any DL between DL-Lite$_{\text{core}}$ and $\mathcal{ALCHI}$ or between $\mathcal{EL}$ and $\mathcal{ALCHI}$, where only concept names (unary predicates) can be closed, and where the actual queries are tree-shaped unions of conjunctive queries (tUCQs). We find it remarkable that there is no difference between classifying OMQCs based on extremely simple DLs such as DL-Lite$_{\text{core}}$ and rather expressive ones such as $\mathcal{ALCHI}$. For the case where also role names (binary predicates) can be closed, we show that for every NP Turing machine M, there is an OMQC that is polynomially equivalent to the complement of M's word problem and where the ontology can either be formulated in DL-Lite or in $\mathcal{EL}$ (and queries are tUCQs). In the case of closed role names, there is thus no dichotomy between PTIME and CONP (unless PTIME = NP) and a full complexity classification does thus not appear feasible with today's knowledge in complexity theory.

We start in Sections 2 and 3 with formally introducing our framework and establishing some preliminary results. In Section 4, we identify a large and practically useful class of OMQCs that are tractable and even FO-rewritable; ontologies in these OMQCs are formulated in DL-Lite$_{\mathcal{R}}$, both concept and role names can be closed, and queries are quantifier-free UCQs. In Section 5, we establish the connection to surjective CSPs for the case where only concept names can be closed (and where quantifiers in the query are allowed) and in Section 6 we establish the connection to Turing machines when also role names can be closed.

Proof details can be found in the appendix which is available at http://informatik.uni-bremen.de/tdki/p.html.

2 Preliminaries

Let N_C, N_R, and N_I be countably infinite sets of *concept*, *role*, and *individual names*. A *DL-Lite concept* is either a concept name or a concept of the form $\exists r$ or $\exists r^-$ with $r \in N_R$. We call r^- an *inverse role* and set $s^- = r$ if $s = r^-$ and $r \in N_R$. A *role* is of the form r or r^-, with $r \in N_R$. A *DL-Lite concept inclusion* is of the form $B_1 \sqsubseteq B_2$ or $B_1 \sqsubseteq \neg B_2$, where B_1, B_2 are DL-Lite concepts. A *role inclusion* is of the form $r \sqsubseteq s$, where r, s are roles. A *DL-Lite$_{core}$ TBox* is a finite set of DL-Lite concept inclusions and a DL-Lite$_{\mathcal{R}}$ TBox might additionally contain role inclusions [Calvanese *et al.*, 2007a; Artale *et al.*, 2009]. As usual in DLs, we use the terms *TBox* and *ontology* interchangeably.

$\mathcal{EL}$ concepts are constructed according to the rule $C, D := \top \mid A \mid C \sqcap D \mid \exists r.C$, where $A \in N_C$ and $r \in N_R$. $\mathcal{ELI}$ concepts extend $\mathcal{EL}$ concepts by adding existential restrictions $\exists r^-.C$, where r^- is an inverse role, and $\mathcal{ALCI}$ further allows negation. For $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}, \mathcal{ALCI}\}$, an $\mathcal{L}$ *concept inclusion* is of the form $C \sqsubseteq D$, where C, D are $\mathcal{L}$ concepts. An *$\mathcal{EL}$ TBox* is a finite set of $\mathcal{EL}$ concept inclusions, and $\mathcal{ELI}$ TBoxes are defined accordingly. An $\mathcal{ALCHI}$ TBox consists of $\mathcal{ALCI}$ concept inclusions and role inclusions. An ABox is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ with $A \in N_C$, $r \in N_R$, and $a, b \in N_I$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individuals used in the ABox $\mathcal{A}$.

An interpretation $\mathcal{I}$ (defined as usual) *satisfies* a concept inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a role inclusion $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, a concept assertion $A(a)$ if $a \in A^{\mathcal{I}}$ and a role assertion $r(a, b)$ if $(a, b) \in r^{\mathcal{I}}$. Note that this interpretation of ABox assertions adopts the standard names assumption (SNA) which requires that $a^{\mathcal{I}} = a$ for all $a \in N_I$ and implies the unique name assumption (UNA). An interpretation is a *model* of a TBox $\mathcal{T}$ if it satisfies all inclusions in $\mathcal{T}$ and a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. As usual, we write $\mathcal{T} \models r \sqsubseteq s$ if every model of $\mathcal{T}$ satisfies the CI $r \sqsubseteq s$ (which can be checked in polytime).

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

In this paper, we combine ontologies (that is, TBoxes) with database queries. A *conjunctive query (CQ)* takes the form $q(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ where $\varphi(\vec{x}, \vec{y})$ is a conjunction of atoms of the form $A(x)$ and $r(x, y)$ with $A \in N_C$ and $r \in N_R$. We call $\vec{x}$ the *answer variables* of $q(\vec{x})$. A *tree CQ (tCQ)* is a CQ that is a tree when viewed as a directed graph (multi-edges disallowed) with the root the only answer variable. An *atomic query (AQ)* is a CQ of the form $A(x)$. A CQ is *Boolean* when it has no answer variables. We use *BtCQ* to refer to Boolean tCQs. A *Boolean AQ (BAQ)* has the form $\exists x\, A(x)$. A *union of conjunctive queries (UCQ)* is a disjunction of CQs. Additional query classes such as tUCQ and BtUCQ are then defined in the obvious way, demanding that every CQ in the UCQ is of the expected form. The *length* $|q|$ of a UCQ q is the number of its variables.

In this paper, the general type of query that we are interested in are *ontology-mediated queries with closed predicates (OMQCs)* that consist of a TBox $\mathcal{T}$, a set Σ_A of predicates that can occur in the ABox, a set of closed predicates $\Sigma_\mathsf{C} \subseteq \Sigma_\mathsf{A}$, and an actual query q (such as a UCQ) that is to be answered. The semantics of such queries is as follows. A model $\mathcal{I}$ of an ABox $\mathcal{A}$ *respects closed predicates* Σ_C if the extension of these predicates agrees with what is explicitly stated in the ABox, that is,

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

Let $Q = (\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ be an OMQC and $\mathcal{A}$ a Σ_A-ABox. A tuple $\vec{a} \in \mathsf{Ind}(\mathcal{A})$ is a *certain answer to Q on $\mathcal{A}$*, written $\mathcal{T}, \mathcal{A} \models_{c(\Sigma_\mathsf{C})} q(\vec{a})$, if $\mathcal{I} \models q[\vec{a}]$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ that respect Σ_C.

Example 1. *Assume an automobile company (say Skoda) wants to monitor the model ranges of international automobile manufacturers. It integrates its company databases storing information about its own products with information about other manufacturers extracted from the web. An ontology is used to provide a unifying vocabulary to be used in queries and to add background knowledge such as*

$$\mathsf{SkodaModel} \sqsubseteq \exists \mathsf{has\_engine}.\mathsf{SkodaEngine}$$
$$\mathsf{TeslaModel} \sqsubseteq \exists \mathsf{has\_engine}.\mathsf{TeslaEngine}$$
$$\mathsf{DieselEngine} \sqsubseteq \mathsf{ICEngine} \qquad \mathsf{PetrolEngine} \sqsubseteq \mathsf{ICEngine}$$

where IC stands for internal combustion. Assume that all data is stored in an ABox, which contains information about Skoda models:

$$\begin{array}{l} \mathsf{SkodaModel}(sm_1), \mathsf{SkodaModel}(sm_2), \ldots \\ \mathsf{SkodaEngine}(se_1), \mathsf{SkodaEngine}(se_2) \ldots \\ \mathsf{has\_engine}(sm_1, se_1), \ldots \\ \mathsf{DieselEngine}(se_1), \mathsf{PetrolEngine}(se_2), \ldots \end{array}$$

and about models by other manufacturers:

$$\begin{array}{l} \mathsf{TeslaModel}(tm_1), \ldots \\ \mathsf{TeslaEngine}(te_1), \mathsf{ElectEngine}(te_1), \ldots \end{array}$$

Skoda is sure that its own models and engines are in the database, therefore the concept names SkodaModel *and* SkodaEngine *are closed. As information about other manufacturers is taken from the web, it is assumed to be incomplete. To illustrate the effect of closing these predicates, consider the following query $q_1(x)$:*

$$\exists y(\mathsf{SkodaModel}(x) \wedge \mathsf{has\_engine}(x,y) \wedge \mathsf{ICEngine}(y)).$$

Assume that sm_{17} is a new Skoda model for which an existing engine will be used, but it is not yet decided which one. Thus the data only contains $\mathsf{SkodaModel}(sm_{17})$*, but no other assertions mentioning sm_{17}. Note that Skoda offers only petrol and diesel engines and that Tesla offers only electric engines which is both reflected in the data (e.g, for every assertion* $\mathsf{TeslaEngine}(te_i)$*, there is an assertion* $\mathsf{ElecEngine}(te_i)$*). Due to the knowledge in the ontology and since* SkodaEngine *is closed, sm_{17} is returned as an answer to q_1. This is not the case without closed predicates and it is in this sense that closed predicates can result in more complete answers. In particular, the query*

$$\exists y(\mathsf{TeslaModel}(x) \wedge \mathsf{has\_engine}(x,y) \wedge \mathsf{ElectEngine}(y))$$

does not return tm_4 if the ABox only contains $\mathsf{TeslaModel}(tm_4)$*, but does not associate tm_4 with any specific engine.*

An *OBDA language* is constituted by a triple $(\mathcal{L}, \Sigma, \mathcal{Q})$ that consists of a TBox language (such as DL-Lite$_\text{core}$, $\mathcal{EL}$, or $\mathcal{ALCHI}$), a set of predicates Σ (such as $\mathsf{N_C} \cup \mathsf{N_R}$ or $\mathsf{N_C}$) and a query language $\mathcal{Q}$ (such as UCQ or CQ). It comprises all OMQCs $(\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ such that $\mathcal{T} \in \mathcal{L}$, $\Sigma_\mathsf{C} \subseteq \Sigma$, and $q \in \mathcal{Q}$. Examples of OBDA languages considered in this paper include, for example, (DL-Lite$_\mathcal{R}$, $\mathsf{N_C} \cup \mathsf{N_R}$, BtUCQ) and ($\mathcal{ALCHI}$, $\mathsf{N_C}$, BtUCQ).

For an ABox $\mathcal{A}$, we denote by $\mathcal{I}_\mathcal{A}$ the interpretation corresponding to $\mathcal{A}$, which satisfies $\Delta^{\mathcal{I}_\mathcal{A}} = \mathsf{Ind}(\mathcal{A})$ and is defined in the obvious way. An OMQC $Q = (\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ is *FO-rewritable* if there is an FO-query $\varphi(\vec{x})$ such that for all Σ_A-ABoxes $\mathcal{A}$ and all tuples $\vec{a}$ of individuals from $\mathsf{Ind}(\mathcal{A})$, we have $\mathcal{I}_\mathcal{A} \models \varphi(\vec{a})$ iff $\mathcal{T}, \mathcal{A} \models_{c(\Sigma_\mathsf{C})} q(\vec{a})$. Here and throughout the paper, we assume that FO-queries use only atoms of the form $A(x)$, $r(x,y)$, and $x = y$ where A is a concept name and r a role name.

In many applications, it is useful to identify and report inconsistencies of the data with the ontology. An ABox $\mathcal{A}$ is *consistent w.r.t. $\mathcal{T}$ and closed* Σ_C if there is a model of $\mathcal{T}$ and $\mathcal{A}$ that respects Σ_C. We say that *ABox consistency is FO-rewritable for an OMQC* $(\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ if there is a Boolean FO-query φ such that for all Σ_A-ABoxes $\mathcal{A}$, we have $\mathcal{I}_\mathcal{A} \models \varphi$ iff $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ and closed Σ_C.

3 Basic Results

We establish some basic results that set the stage for the rest of the paper. The first one is a general coNP upper bound (in data complexity) that encompasses all OMQ languages studied in this paper. Note that this bound is not a consequence of results on OBDA querying with nominals [Ortiz *et al.*, 2008] because nominals are part of the TBox and thus their number is a constant while closing a predicate corresponds to considering a set of individuals whose number is bounded by the size of the ABox (the input size). The proof uses a decomposition of countermodels (models that demonstrate query non-entailment) into mosaics and then relies on a guess-and-check algorithm for finding such decompositions.

Theorem 2. *Every OMQC in $(\mathcal{ALCHI}, \mathsf{N_C} \cup \mathsf{N_R}, UCQ)$ is in* coNP.

Our next result concerns the relationship between ABox signatures and closed predicates. In the cases relevant for us, we can assume w.l.o.g. that the ABox signature and the set of closed predicates coincide. This setup was called *DBoxes* in [Seylan *et al.*, 2009; Franconi *et al.*, 2011]. In the remainder of the paper, we are thus free to assume that OMQCs $(\mathcal{T}, \Sigma_A, \Sigma_C, q)$ satisfy $\Sigma_A = \Sigma_C$ whenever convenient. We denote such OMQCs as a triple $(\mathcal{T}, \Sigma, q)$ meaning that Σ serves both as Σ_A and Σ_C. Two Σ-queries Q_1 and Q_2 are *equivalent* if $Q_1(\mathcal{A}) = Q_2(\mathcal{A})$ for all Σ-ABoxes $\mathcal{A}$. A class $\mathcal{Q}$ of queries is called *canonical* if it is closed under replacing a concept or role atom in a query with an atom of the same kind. All classes of queries considered in this paper are canonical.

Theorem 3. *Let $\mathcal{L} \in \{$DL-Lite$_\mathcal{R}, \mathcal{ALCHI}\}$ and $\mathcal{Q}$ be a canonical class of UCQs, or let $\mathcal{L} \in \{$DL-Lite$_{core}, \mathcal{EL}\}$ and $\mathcal{Q}$ be a canonical class of UCQs closed under forming unions of queries. Then for every OMQC $Q = (\mathcal{T}, \Sigma_A, \Sigma_C, q)$ from $(\mathcal{L}, N_C \cup N_R, \mathcal{Q})$, one can construct in polynomial time an equivalent query OMQC $Q' = (\mathcal{T}', \Sigma_A, \Sigma_A, q')$ with $\mathcal{T}' \in \mathcal{L}$ and $q' \in \mathcal{Q}$.*

As noted in the introduction, full first-order queries can be used for the closed predicates. This simple observation was already made in [Lutz *et al.*, 2013] in a related but slightly different setup, and we repeat it here for the setup considered in the present paper. Let Σ_C be a signature that declares closed predicates and let $q = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$ be a CQ. An *FO(Σ_C)-extension of* q is a query of the form $\exists \vec{y} \exists \vec{z}\, \varphi(\vec{x}, \vec{y}) \wedge \psi_1(\vec{z}_1), \ldots, \psi_n(\vec{z}_n)$ where $\psi_1(\vec{z}_1), \ldots, \psi_n(\vec{z}_n)$ are FO-queries with $\mathsf{sig}(\psi_i) \subseteq \Sigma_C$ and $\vec{z}_1 \cup \cdots \cup \vec{z}_n \subseteq \vec{x} \cup \vec{y} \cup \vec{z}$.

Theorem 4. *If an OMQC $(\mathcal{T}, \Sigma_A, \Sigma_C, q)$ is FO-rewritable (in* PTIME*), then every $(\mathcal{T}, \Sigma_A, \Sigma_C, q')$ with q' an FO(Σ_C)-extension of q is also FO-rewritable (in* PTIME*).*

4 Quantifier-Free UCQs and FO-Rewritability

The aim of this section is to identify useful OBDA languages whose UCQs are guaranteed to be FO-rewritable. It turns out that FO-rewritability can be achieved by combining lightweight DLs with quantifier-free queries. We use qfUCQ to denote the class of *quantifier-free* UCQs, that is, none of the CQs is allowed to contain a quantified variable.

Our main result is that all OMQCs from the OBDA language (DL-Lite$_\mathcal{R}$, $N_C \cup N_R$, qfUCQ) are FO-rewritable under the mild restriction that there is no role inclusion which requires an open role to be contained in a closed one. We believe that this class of queries is potentially relevant for practical applications. In fact, DL-Lite$_\mathcal{R}$ is very popular as an ontology language in large-scale OBDA and, in practice, many queries turn out to be quantifier-free. Note that the query language SPARQL, which is used in many web applications, is closely related to qfUCQs and, in fact, does not admit existential quantification under its standard entailment regimes [Glimm and Krötzsch, 2010]. We remark that our result also covers the OBDA language (DL-Lite$_{\mathsf{core}}$, $N_C \cup N_R$, qfUCQ) without further restrictions.

Theorem 5. *Every OMQC $(\mathcal{T}, \Sigma_A, \Sigma_C, q)$ from (DL-Lite$_\mathcal{R}$, $N_C \cup N_R$, qfUCQ) such that $\mathcal{T}$ contains no role inclusion of the form $s \sqsubseteq r$ with $s \notin \Sigma_C$ and $r \in \Sigma_C$ is FO-rewritable.*

We first show that ABox consistency is FO-rewritable for each of the OMQCs $(\mathcal{T}, \Sigma_A, \Sigma_C, q)$ covered by Theorem 5. On inconcistent ABoxes, every query returns all possible answers, but in most practical cases it is more useful to detect and point out the inconsistency instead. We assume w.l.o.g. that $\Sigma_A = \Sigma_C$ and denote Σ_A by Σ. Let $\mathsf{sub}(\mathcal{T})$ be the set of all concept names in $\mathcal{T}$, their negations, and all concepts $\exists r, \exists r^-$ such that r is a role name that occurs in $\mathcal{T}$. A *$\mathcal{T}$-type* is a set $t \subseteq \mathsf{sub}(\mathcal{T})$ such that for all $B_1, B_2 \in \mathsf{sub}(\mathcal{T})$:

- if $B_1 \in t$ and $\mathcal{T} \models B_1 \sqsubseteq B_2$, then $B_2 \in t$;
- if $B_1 \in t$ and $\mathcal{T} \models B_1 \sqsubseteq \neg B_2$, then $B_2 \notin t$.

A *$\mathcal{T}$-typing* is a set T of $\mathcal{T}$-types. A *path in T* is a sequence $t, r_1, \ldots, r_n$ where $t \in T$, $\exists r_1, \ldots, \exists r_n \in \mathsf{sub}(\mathcal{T})$ use no symbols from Σ, $\exists r_1 \in t$ and for $i \in \{1, \ldots, n-1\}$, $\mathcal{T} \models \exists r_i^- \sqsubseteq \exists r_{i+1}$ and $r_i^- \neq r_{i+1}$. The path is *Σ-participating* if for all $i \in \{1, \ldots, n-1\}$, there is no $B \in \mathsf{sub}(\mathcal{T})$ with $\mathsf{sig}(B) \subseteq \Sigma$ and $\mathcal{T} \models \exists r_i^- \sqsubseteq B$ while there is such a B for $i = n$. A $\mathcal{T}$-typing T is *Σ-realizable* if for every Σ-participating path $t, r_1, \ldots, r_n$ in T, there is some $u \in T$ such that $\{B \in \mathsf{sub}(\mathcal{T}) \mid \mathcal{T} \models \exists r_n^- \sqsubseteq B\} \subseteq u$.

A $\mathcal{T}$-typing T provides an abstraction of a model of $\mathcal{T}$ and a Σ-ABox $\mathcal{A}$, where T contains the types that are realized by ABox elements. Σ-realizability ensures that we can build from T a model that respects the closed predicates in Σ. To make this more precise, define a *$\mathcal{T}$-decoration of* a Σ-ABox $\mathcal{A}$ to be a mapping f that assigns to each $a \in \mathsf{Ind}(\mathcal{A})$ a $\mathcal{T}$-type $f(a)$ such that $f(a)|_\Sigma = t^a_\mathcal{A}|_\Sigma$ where $t^a_\mathcal{A} = \{B \in \mathsf{sub}(\mathcal{T}) \mid a \in B^{\mathcal{I}_\mathcal{A}}\}$ and $S|_\Sigma$ denotes the restriction of the set S of concept to those member that only use symbols from Σ. For brevity, let $R_\Sigma = \{s \sqsubseteq r \mid \mathcal{T} \models s \sqsubseteq r \text{ and } \mathsf{sig}(s \sqsubseteq r) \subseteq \Sigma\}$.

Lemma 6. *A Σ-ABox $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ and closed Σ iff*

1. *$\mathcal{A}$ has a $\mathcal{T}$-decoration f whose image is a Σ-realizable $\mathcal{T}$-typing and*
2. *$s(a, b) \in \mathcal{A}$ and $s \sqsubseteq r \in R_\Sigma$ implies $r(a, b) \in \mathcal{A}$.*

We now construct the required FO-query. For all role names r and variables x, y, define $\psi_r(x, y) = r(x, y)$ and $\psi_{r^-}(x, y) = r(y, x)$. For all concept names A and roles r, define $\psi_A(x) = A(x)$ and $\psi_{\exists r}(x) = \exists y\, \psi_r(x, y)$. For each $\mathcal{T}$-type t, set

$$\psi_t(x) = \bigwedge_{B \in \mathsf{sub}(\mathcal{T}) \setminus t \text{ with } \mathsf{sig}(B) \subseteq \Sigma} \neg \psi_B(x) \wedge \bigwedge_{B \in t \text{ with } \mathsf{sig}(B) \subseteq \Sigma} \psi_B(x)$$

and for each $\mathcal{T}$-typing $T = \{t_1, \ldots, t_n\}$, set

$$\psi_T = \forall x \bigvee_{t \in T} \psi_t(x) \wedge \exists x_1 \cdots \exists x_n (\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_i \psi_{t_i}(x_i)).$$

Let $\mathcal{R}$ be the set of all Σ-realizable typings and set

$$\Psi_{\mathcal{T},\Sigma} = \bigvee_{T \in \mathcal{R}} \psi_T \wedge \bigwedge_{s \sqsubseteq r \in R_\Sigma} \forall x \forall y (\psi_s(x, y) \to \psi_r(x, y)).$$

Note that the two conjuncts of $\Psi_{\mathcal{T},\Sigma}$ express exactly Points 1 and 2 of Lemma 6. We have thus shown FO-rewritability of ABox consistency for Q.

Proposition 7. *A Σ-ABox $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ and closed Σ iff $\mathcal{I}_\mathcal{A} \models \Psi_{\mathcal{T},\Sigma}$.*

The next step is to construct an FO-rewriting of Q over ABoxes that are consistent w.r.t. $\mathcal{T}$ and closed Σ. If desired, this rewriting can be combined with the one for ABox consistency given above. Due to space limitations and the slightly intricate details of the construction, we only give a rough intuition and defer the details to the appendix.

Whereas the FO-query $\Psi_{\mathcal{T},\Sigma}$ above is Boolean and identifies ABoxes that have a common model with $\mathcal{T}$, we now aim to construct an FO-formula $\varphi(\vec{x})$ (where $\vec{x}$ are the answer variables of the actual query q from Q) such that for all Σ-ABoxes $\mathcal{A}$ and $\vec{a} \in \mathsf{Ind}(\mathcal{A})$, we have $\mathcal{I}_\mathcal{A} \models \varphi[\vec{a}]$ iff there is a common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that there is no homomorphism from some CQ in q to $\mathcal{I}$ that takes $\vec{x}$ to $\vec{a}$. The desired FO-rewriting $\Phi_Q(\vec{x})$ of Q over consistent ABoxes is then simply the negation of $\varphi(\vec{x})$. The construction of $\varphi(\vec{x})$ is based on an extended notion of $\mathcal{T}$-typing called $\mathcal{T}, q$*-typing* that provides an abstraction of a model of $\mathcal{T}$ and a Σ-ABox $\mathcal{A}$ that avoids a homomorphism from $\vec{x}$ to certain individuals $\vec{a}$. This finishes the proof of Theorem 5.

We now show that without the restriction on role inclusions adopted in Theorem 5, OMQCs from $(\text{DL-Lite}_\mathcal{R}, \mathsf{N_C} \cup \mathsf{N_R}, \text{qfUCQ})$ are no longer FO-rewritable. In fact, we prove the following, slightly stronger result by reduction from propositional satisfiability.

Theorem 8. *There is a DL-Lite*$_\mathcal{R}$ *TBox* $\mathcal{T}$ *and set of predicates* Σ_C *such that ABox consistency w.r.t.* $\mathcal{T}$ *and closed* Σ_C *is* NP-*complete.*

We close this section with noting that, for the case of $\mathcal{EL}$ and its extensions, quantifier-free queries are computationally no more well-behaved than unrestricted queries; in fact, all hardness results established in the remainder of this paper for $\mathcal{EL}$ and its extensions can be adapted to the case of quantifier-free queries.

5 Closing Concept Names: Connection to CSP

We consider OBDA languages that only allow to close concept names, but not role names. Unlike in the previous section, queries admit unrestricted existential quantification. Our main contribution here is to establish a close connection between such OBDA languages based on a wide range of DLs and surjective constraint satisfaction problems. This result implies that a full complexity classification of these two problem classes is intimately related. In fact, a full complexity classification of surjective CSPs is a very difficult, ongoing research effort. As pointed out in the introduction, there are even concrete surjective CSPs whose complexity is unknown and, via the established connection, these problems can be used to derive concrete OMQCs whose computational properties are currently not understood.

We start with introducing CSPs. An interpretation $\mathcal{I}$ is a Σ-*interpretation* if it only interprets symbols in Σ, that is, all other symbols are interpreted as empty. Every finite Σ-interpretation $\mathcal{I}$ defines a constraint satisfaction problem $\mathsf{CSP}(\mathcal{I})$ in signature Σ: given a finite Σ-interpretation $\mathcal{I}'$, decide whether there is a homomorphism from $\mathcal{I}'$ to $\mathcal{I}$, i.e., a mapping $h : \Delta^{\mathcal{I}'} \to \Delta^\mathcal{I}$ such that $d \in A^{\mathcal{I}'}$ implies $h(d) \in A^\mathcal{I}$ and $(d, e) \in r^{\mathcal{I}'}$ implies $(h(d), h(e)) \in r^\mathcal{I}$. The problem $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$ is the variant of $\mathsf{CSP}(\mathcal{I})$ where we require h to be surjective. Note that we do not consider CSPs with relations of arity larger than two.

We first show that for every problem $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$, there is an OMQC Q from $(\text{DL-Lite}_\text{core}, \mathsf{N_C}, \text{BtUCQ})$ that has the same complexity as the complement of $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$, up to polynomial time reductions. Here, the complexity of an OMQC $Q = (\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ is the complexity to decide, given a Σ_A-ABox $\mathcal{A}$, whether $\mathcal{T}, \mathcal{A} \models_{c(\Sigma_\mathsf{C})} q$.

Consider $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$ in signature Σ. We may assume w.l.o.g. that there is at least one Σ-interpretation $\mathcal{J}$ that does *not* homomorphically map to $\mathcal{I}$.[1] Define the OMQC $(\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ as follows:

$$
\begin{aligned}
\mathcal{T} &= \{A \sqsubseteq \exists \mathsf{val},\ \exists \mathsf{val}^- \sqsubseteq V\} \cup \\
&\quad\ \{A \sqsubseteq \exists \mathsf{aux}_d,\ \exists \mathsf{aux}_d^- \sqsubseteq V \sqcap V_d \mid d \in \Delta^\mathcal{I}\} \cup \\
&\quad\ \{A \sqsubseteq \exists \mathsf{force}_d,\ \exists \mathsf{force}_d^- \sqsubseteq A \mid d \in \Delta^\mathcal{I}\} \\
\Sigma_\mathsf{C} &= \{A, V\} \cup \{V_d \mid d \in \Delta^\mathcal{I}\} \\
\Sigma_\mathsf{A} &= \Sigma \cup \Sigma_\mathsf{C} \\
q &= q_1 \vee q_2 \vee q_3
\end{aligned}
$$

where

$$
\begin{aligned}
q_1 = \bigvee_{d,d' \in \Delta^\mathcal{I} \mid d \neq d'} &\exists x \exists y_1 \exists y_2\, A(x) \wedge \mathsf{val}(x, y_1) \wedge \\
&\quad \mathsf{val}(x, y_2) \wedge V_d(y_1) \wedge V_{d'}(y_2) \\
q_2 = \bigvee_{d,d' \in \Delta^\mathcal{I}, r \in \Sigma \mid (d,d') \notin r^\mathcal{I}} &\exists x \exists y \exists x_1 \exists y_1\, A(x) \wedge A(y) \wedge r(x, y) \wedge \\
&\quad \mathsf{val}(x, x_1) \wedge \mathsf{val}(y, y_1) \wedge \\
&\quad V_d(x_1) \wedge V_{d'}(y_1) \\
q_3 = \bigvee_{d,d' \in \Delta^\mathcal{I} \mid d \neq d'} &\exists x \exists y \exists z\, A(x) \wedge \mathsf{force}_d(z, x) \wedge \\
&\quad \mathsf{val}(x, y) \wedge V_{d'}(y).
\end{aligned}
$$

To understand the construction, it is useful to consider the reduction of (the complement of) $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$ to $(\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$. Given a Σ-interpretation $\mathcal{J}$ that is an input to $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$, we construct a Σ_A-ABox $\mathcal{A}$ as an input to $(\mathcal{T}, \Sigma_\mathsf{A}, \Sigma_\mathsf{C}, q)$ as

$$\mathcal{A}_\mathcal{J} \cup \{A(d) \mid d \in \Delta^\mathcal{J}\} \cup \{V(a_d), V_d(a_d) \mid d \in \Delta^\mathcal{I}\}$$

where $\mathcal{A}_\mathcal{J}$ is $\mathcal{J}$ viewed as an ABox (with the elements of $\mathcal{J}$ serving as ABox individuals) and where a_d is a fresh individual name for each $d \in \Delta^\mathcal{I}$. We show in the appendix that $\mathcal{J} \in \mathsf{CSP}(\mathcal{I})^\mathsf{sur}$ iff $\mathcal{T}, \mathcal{A} \not\models_{c(\Sigma_\mathsf{C})} q$. For the "if" direction, we extract a homomorphism h from a model $\mathcal{I}'$ of $\mathcal{T}$ and $\mathcal{A}$ that respects Σ_C and satisfies $\mathcal{I} \not\models q$ by setting $h(d) = e$ when $(d, a_e) \in \mathsf{val}^{\mathcal{I}'}$. The first line of $\mathcal{T}$ and the closing of V thus ensure that $h(d)$ is defined for every $d \in \Delta^\mathcal{J}$ and q_1 ensures that the value of $h(d)$ is unique; q_2 ensures that h is a homomorphism and Line 3 of $\mathcal{T}$ together with the closing of A and q_3 guarantees that it is surjective. Line 2 of $\mathcal{T}$ is only needed for the converse reduction to go through, ensuring that we always consider homomorphisms onto $\mathcal{I}$. We say that two decision problems P_1 and P_2 are *polynomially equivalent* if P_1 polynomially reduces to P_2 and vice versa.

[1]Otherwise we can simply use as Q any OMQC $(\mathcal{T}, \Sigma, \Sigma_\mathsf{C}, q)$ such that $\mathcal{T}, \mathcal{A} \not\models_{c(\Sigma_\mathsf{C})} q$ for all Σ-ABoxes $\mathcal{A}$. Then $\mathsf{CSP}(\mathcal{I})^\mathsf{sur}$ is exactly the complement problem of Q.

Lemma 9. *The complement of* $\mathsf{CSP}(\mathcal{I})^{\mathsf{sur}}$ *is polynomially equivalent to* $(\mathcal{T}, \Sigma_{\mathsf{A}}, \Sigma_{\mathsf{C}}, q)$.

Note that the same reduction works when DL-Lite$_{\text{core}}$ is replaced with $\mathcal{EL}$. For example, the first line of $\mathcal{T}$ then reads $A \sqsubseteq \exists \mathsf{val}.V$. Since all disjuncts of q are tree-shaped, we can view them as $\mathcal{EL}$-concepts, extend $\mathcal{T}$ with $q' \sqsubseteq A_0$ for every disjunct q' of q, and replace q with the BAQ $\exists x\, A_0(x)$. We have thus established the following result.

Theorem 10. *For every* $\mathsf{CSP}(\mathcal{I})^{\mathsf{sur}}$*, there is an OMQC* Q *from* (*DL-Lite*$_{core}$, $\mathsf{N_C}$, *BtUCQ*) *such that the complement of* $\mathsf{CSP}(\mathcal{I})^{\mathsf{sur}}$ *has the same complexity as* Q*, up to polytime reductions. The same holds for* ($\mathcal{EL}$, $\mathsf{N_C}$, *BAQ*).

We remark that, as can easily be verified by checking the constructions in the proof of Lemma 9, the complement of $\mathsf{CSP}(\mathcal{I})^{\mathsf{sur}}$ and Q actually have the same complexity up to *FO reductions* [Immerman, 1999]. This links the complexity of the two problems even closer. For example, if one is complete for LOGSPACE or in AC$_0$, then so is the other.

We now establish (almost) a converse of Theorem 10 by showing that for every OMQC Q from ($\mathcal{ALCHI}$, $\mathsf{N_C}$, BtUCQ), there is a *generalized* surjective CSP that has the same complexity as the complement of Q, up to polytime reductions. A generalized surjective CSP in signature Σ is characterized by a *finite set* Γ of finite Σ-interpretations instead of a single such interpretation, denoted $\mathsf{CSP}(\Gamma)^{\mathsf{sur}}$. The problem is to decide, given a Σ-interpretation $\mathcal{I}'$, whether there is a homomorphism from $\mathcal{I}'$ to some interpretation in Γ. Note that, in the non-surjective case, every generalized CSP can be translated into an equivalent non-generalized CSP [Foniok *et al.*, 2008]. In the surjective case, such a translation is not known, so there remains a gap in our analysis. Closing this gap is primarily a (difficult!) CSP question rather than an OBDA question.

Let $Q = (\mathcal{T}, \Sigma_{\mathsf{A}}, \Sigma_{\mathsf{C}}, q)$ be an OMQC from ($\mathcal{ALCI}$, $\mathsf{N_C}$, BtUCQ). We assume w.l.o.g. that $\mathcal{T}$ contains only a single concept inclusion $\top \sqsubseteq C_{\mathcal{T}}$ (plus role inclusions) where $C_{\mathcal{T}}$ uses only the constructors $\neg$, $\sqcap$, and $\exists$, and that q has the form $\exists x\, A_0(x)$ with A_0 occurring in $\mathcal{T}$. The latter is possible since every tCQ q can be expressed as an $\mathcal{ALCI}$ concept C_q in an obvious way, and thus when the BtUCQ is $q = \bigvee_i \exists x_i\, q_i$, then we can add $C_{q_i} \sqsubseteq A_0$ to the TBox with A_0 a fresh concept name and replace q with $\exists x\, A_0(x)$. We use $\mathsf{cl}(\mathcal{T})$ to denote the set of subconcepts of $C_{\mathcal{T}}$, extended with all concepts $\exists s.C$ such that $\exists r.C$ is a subconcept of $C_{\mathcal{T}}$ and $r \sqsubseteq s \in \mathcal{T}$, as well as the negations of these concepts. A *Q-type* is a subset $t \subseteq \mathsf{cl}(\mathcal{T})$ such that for some model $\mathcal{I}$ of $\mathcal{T}$ and some $d \in \Delta^{\mathcal{I}}$, we have $t = \mathsf{tp}_{\mathcal{I}}(d) := \{C \in \mathsf{cl}(\mathcal{T}) \mid d \in C^{\mathcal{I}}\}$. Let $\mathsf{TP}(\mathcal{T})$ denote the set of all types for Q. For $t, t' \in \mathsf{TP}(\mathcal{T})$ and a role r, we write $t \rightsquigarrow_r t'$ if for all roles r with $\mathcal{T} \models r \sqsubseteq s$, we have

- $C \in t'$ implies $\exists s.C \in t$, for all $\exists s.C \in \mathsf{cl}(\mathcal{T})$ and
- $C \in t$ implies $\exists s^-.C \in t'$, for all $\exists s^-.C \in \mathsf{cl}(\mathcal{T})$.

A subset $T \subseteq \mathsf{TP}(\mathcal{T})$ is *realizable in a countermodel* if there is a Σ_{A}-ABox $\mathcal{A}$ and model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ that respects closed predicates Σ_{C} such that $\mathcal{I} \not\models q$ and

$$T = \{\mathsf{tp}_{\mathcal{I}}(a) \mid a \in \mathsf{Ind}(\mathcal{A})\}.$$

We define the desired surjective generalized CSP by taking one template for each $T \subseteq \mathsf{TP}(\mathcal{T})$ that is realizable in a countermodel. The signature Σ of the CSP comprises the symbols in Σ_{A}, one concept name $\overline{A}$ for each concept name in Σ_{C}, and the concept name A_0 from q. We assume w.l.o.g. that there is at least one concept name in Σ_{C} and at least one concept name $A_{\mathsf{open}} \in \Sigma_{\mathsf{A}} \setminus \Sigma_{\mathsf{C}}$.

Each $T \subseteq \mathsf{TP}(\mathcal{T})$ realizable in a countermodel gives raise to a template $\mathcal{I}_T$, defined as follows:

$$\begin{array}{rcl}
\Delta^{\mathcal{I}_T} & = & T \uplus \{d_A \mid A \in \Sigma_{\mathsf{C}}\} \\
A^{\mathcal{I}_T} & = & \{t \in T \mid A \in t\} \cup \{d_B \mid B \neq A\} \\
\overline{A}^{\mathcal{I}_T} & = & \{t \in T \mid A \notin t\} \cup \{d_B \mid B \neq A\} \\
r^{\mathcal{I}_T} & = & \{(t, t') \in T \times T \mid t \rightsquigarrow_r t'\} \cup \\
 & & \{(d, d') \in \Delta^{\mathcal{I}_T} \times \Delta^{\mathcal{I}_T} \mid \{d, d'\} \setminus T \neq \emptyset\}.
\end{array}$$

Note that, in $\mathcal{I}_T$ restricted to domain T, $\overline{A}$ is interpreted as the complement of A. At each element d_A, all concept names except A and $\overline{A}$ are true, and these elements are connected to all elements with all roles. Intuitively, we need the concept names $\overline{A}$ to ensure that when an assertion $A(a)$ is missing in an ABox $\mathcal{A}$ with A closed, then a can only be mapped to a template element that does not make A true; this is done by extending $\mathcal{A}$ with $\overline{A}(a)$ and exploiting that $\overline{A}$ is essentially the complement of A in each $\mathcal{I}_T$. The elements d_A are then needed to deal with inputs to the CSP where some point satisfies neither A nor $\overline{A}$. Let Γ be the set of all interpretations $\mathcal{I}_T$ obtained in the described way.

Lemma 11. $(\mathcal{T}, \Sigma_{\mathsf{A}}, \Sigma_{\mathsf{C}}, q)$ *is polynomially equivalent to the complement of* $\mathsf{CSP}(\Gamma)^{\mathsf{sur}}$.

We have thus established the following result

Theorem 12. *For every OMQC* Q *from* ($\mathcal{ALCHI}$, $\mathsf{N_C}$, *BtUCQ*)*, there is a (generalized)* $\mathsf{CSP}(\Gamma)^{\mathsf{sur}}$ *in binary signature such that* Q *has the same complexity as the complement of* $\mathsf{CSP}(\Gamma)^{\mathsf{sur}}$*, up to polytime reductions.*

Again, the theorem can actually be strengthened to state the same complexity up to FO reductions. Note that the DL $\mathcal{ALCHI}$ used in Theorem 12 is a significant extension of the DLs referred to in Theorem 10 and thus our results apply to a remarkable range of DLs: all DLs between DL-Lite$_{\text{core}}$ and $\mathcal{ALCHI}$ as well as all DLs between $\mathcal{EL}$ and $\mathcal{ALCHI}$.

6 Closing Role Names: TM Equivalence

We generalize the setup from the previous section by allowing also role names to be closed. Our main result is that for every NP Turing machine M, there is an OMQC in (DL-Lite$_{\mathcal{R}}$, $\mathsf{N_C} \cup \mathsf{N_R}$, BtUCQ) that is polynomially equivalent to the complement of M's word problem, and the same is true for ($\mathcal{EL}$, $\mathsf{N_C} \cup \mathsf{N_R}$, BAQ). By Ladner's theorem, it follows that there are CONP-intermediate OMQCs in both of the mentioned OBDA languages (unless P = NP) and that a full complexity classification of the queries in these languages is currently far beyond reach.

To establish the above, we utilize a result from [Lutz and Wolter, 2012; Bienvenu *et al.*, 2014] which states that for every NP Turing machine M, there is a monadic disjunctive datalog program of a certain restricted shape that is polynomially equivalent to the complement of M's word problem. It thus suffices to show that for every such datalog program, there is a polynomially equivalent OMQC of the required form. The actual reduction then uses similar ideas as the proof of Theorem 10.

Theorem 13. *For every simple disjunctive datalog program* Π, *there exists an OMQC in* $(\mathcal{EL}, \mathsf{N_C} \cup \mathsf{N_R}, \mathit{BAQ})$ *that is polynomially equivalent to* Π. *The same is true for* $(\mathit{DL\text{-}Lite}_{\mathcal{R}}, \mathsf{N_C} \cup \mathsf{N_R}, \mathit{BtUCQ})$.

The computational status of $(\text{DL-Lite}_{\text{core}}, \mathsf{N_C} \cup \mathsf{N_R}, \text{BtUCQ})$ remains open. In particular, it is open whether Theorem 13 can be strengthened to this case. We are, however, able to clarify the status of $(\text{DL-Lite}_{\text{core}}, \mathsf{N_C} \cup \mathsf{N_R}, \text{BtCQ})$, where BtUCQs are replaced with BtCQs. In fact, we show that every OMQC in this language is polynomially equivalent to an OMQC that is formulated in the same language but does *not* use closed roles. Via the results in Section 5, OMQCs in $(\text{DL-Lite}_{\text{core}}, \mathsf{N_C} \cup \mathsf{N_R}, \text{BtCQ})$ are thus linked to surjective CSPs.

Theorem 14. *For every OMQC in* $(\mathit{DL\text{-}Lite}_{\mathsf{core}}, \mathsf{N_C} \cup \mathsf{N_R}, \mathit{BtCQ})$ *there exists a polynomially equivalent OMCQ in* $(\mathit{DL\text{-}Lite}_{\mathsf{core}}, \mathsf{N_C}, \mathit{BtCQ})$.

Interestingly, Theorem 14 can be extended from BtCQs to BtUCQs for the case of ABoxes in which there is at least one assertion for each closed role name. The unrestricted case of $(\text{DL-Lite}_{\text{core}}, \mathsf{N_C} \cup \mathsf{N_R}, \text{BtUCQ})$, though, remains open.

7 Conclusion

Admitting closed predicates in OBDA has significant advantages as it can result in more complete answers and allows to use FO queries for the closed part of the vocabulary. The main results of this paper characterize the computational challenges: for a wide range of DLs, closing concept names corresponds to moving from CSPs to surjective CSPs, while closing role names yields the full computational power of NP. As future work, it would be interesting to exploit closed predicates in OBDA practice. Our results on FO-rewritability of quantifier-free UCQs provide a starting point.

References

[Artale *et al.*, 2009] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *JAIR*, 36:1–69, 2009.

[Bienvenu *et al.*, 2014] M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.*, 39(4):33, 2014.

[Bodirsky *et al.*, 2012] M. Bodirsky, J. Kára, and B. Martin. The complexity of surjective homomorphism problems–a survey. *Disc. Appl. Math.*, 160(12):1680–1690, 2012.

[Bulatov, 2011] A.A. Bulatov. On the CSP dichotomy conjecture. In *CSR*, 2011.

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

[Calvanese *et al.*, 2007b] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. EQL-Lite: Effective first-order query processing in description logics. In *IJCAI*, 2007.

[Chen, 2014] H. Chen. An algebraic hardness criterion for surjective constraint satisfaction. *Algebra universalis*, 72(4):393–401, 2014.

[Donini *et al.*, 2002] F.M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. on Comp. Logic*, 3(2):177–225, 2002.

[Feder and Vardi, 1993] T. Feder and M.Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *STOC*, 1993.

[Foniok *et al.*, 2008] J. Foniok, J. Nesetril, and C. Tardif. Generalised dualities and maximal finite antichains in the homomorphism order of relational structures. *Eur. J. Comb.*, 29(4):881–899, 2008.

[Franconi *et al.*, 2011] E. Franconi, Y. Angélica Ibáñez-García, and İ. Seylan. Query answering with DBoxes is hard. *El. Notes in Theo. Comp. Sci.*, 278:71–84, 2011.

[Glimm and Krötzsch, 2010] B. Glimm and M. Krötzsch. SPARQL beyond subgraph matching. In *ISWC*, 2010.

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

[Hustadt *et al.*, 2005] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *IJCAI*, 2005.

[Immerman, 1999] N. Immerman. *Descriptive Complexity*. Springer, 1999.

[Kun and Szegedy, 2009] G. Kun and M. Szegedy. A new line of attack on the dichotomy conjecture. In *STOC*, 2009.

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

[Lutz *et al.*, 2013] C. Lutz, İ. Seylan, and F. Wolter. Ontology-based data access with closed predicates is inherently intractable (sometimes). In *IJCAI*, 2013.

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

[Ngo *et al.*, 2015] N. Ngo, M. Ortiz, and M. Simkus. The combined complexity of reasoning with closed predicates in description logics. In *DL*, 2015.

[Ortiz *et al.*, 2008] M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. *J. of Auto. Reas.*, 41(1):61–98, 2008.

[Sengupta *et al.*, 2011] K. Sengupta, A.A. Krisnadhi, and P. Hitzler. Local closed world semantics: Grounded circumscription for OWL. In *ISWC*, 2011.

[Seylan *et al.*, 2009] İ. Seylan, E. Franconi, and J. de Bruijn. Effective query rewriting with ontologies over DBoxes. In *IJCAI*, 2009.