From Conjunctive Queries to Instance Queries in Ontology-Mediated Querying

Cristina Feier¹, Carsten Lutz¹, Frank Wolter²,
¹ University of Bremen, Germany
² University of Liverpool, UK
feier@uni-bremen.de, clu@uni-bremen.de, wolter@liverpool.ac.uk

Abstract

We consider ontology-mediated queries (OMQs) based on expressive description logics of the $\mathcal{ALC}$ family and (unions) of conjunctive queries, studying the rewritability into OMQs based on instance queries (IQs). Our results include exact characterizations of when such a rewriting is possible and tight complexity bounds for deciding rewritability. We also give a tight complexity bound for the related problem of deciding whether a given MMSNP sentence is equivalent to a CSP.

1 Introduction

An ontology-mediated query (OMQ) is a database-style query enriched with an ontology that contains domain knowledge, aiming to deliver more complete answers. In OMQs, ontologies are often formulated in a description logic (DL) and query languages of interest include conjunctive queries (CQs), unions of conjunctive queries (UCQs), and instance queries (IQs). While CQs and UCQs are widely known query languages that play a fundamental role also in database systems and theory, IQs are more closely linked to DLs. In fact, an IQ takes the form $C(x)$ with $C$ a concept formulated in the DL that is also used for the ontology, and thus the expressive power of IQs depends on the ontology language. OMQs based on (U)CQs are more powerful than OMQs based on IQs as the latter only serve to return all objects from the data that are instances of a given class.

It is easy to see that IQs can express tree-shaped CQs with a single answer variable as well as unions thereof. In fact, this observation has been used in many technical constructions in the area, see for example([Calvanese et al., 1998; Glimm et al., 2008; Lutz, 2008; Eiter et al., 2012a]). Intriguingly, though, it was observed by Zolin [2007] that tree-shaped CQs are not the limit of IQ-rewritability when we have an expressive DL such as $\mathcal{ALC}$ or $\mathcal{ALCI}$ at our disposal. For example, the CQ $r(x, x)$, which asks to return all objects from the data that are involved in a reflexive $r$-loop, can be rewritten into the equivalent $\mathcal{ALC}$-IQ $P \rightarrow \exists r. P(x)$. Here, $P$ behaves like a monadic second-order variable due to the open-world assumption made for OMQs: we are free to interpret $P$ in any possible way and when making $P$ true at an object we are forced to make also $\exists r. P$ true if and only if the object is involved in a reflexive $r$-loop. It is an interesting question, raised in ([Zolin, 2007; Kikot and Zolin, 2013; Kikot et al., 2013]), to precisely characterize the class of CQs that are rewritable into IQs. An important step into this direction has been made by Kikot and Zolin [2013] who identify a large class of CQs that are rewritable into IQs: a CQ is rewritable into an $\mathcal{ALCI}$-IQ if it is connected and every cycle passes through the (only) answer variable; for rewritability into an $\mathcal{ALC}$-IQ, one additionally requires that all variables are reachable from the answer variable in a directed sense. It remained open whether these classes are depleting, that is, whether they capture all CQs that are IQ-rewritable.

There are two additional motivations to study the stated question. The first one comes from concerns about the practical implementation of OMQs. When the ontology is formulated in a more inexpressive ‘Horn DL’, OMQ evaluation is possible in PTIME data complexity and a host of techniques for practically efficient OMQ evaluation is available, see for example ([Perez-Urbina et al., 2010; Eiter et al., 2012a; Trivelu et al., 2015; Lutz et al., 2009]). In the case of expressive DLs such as $\mathcal{ALC}$ and $\mathcal{ALCI}$, OMQ evaluation is coNP-complete in data complexity and efficient implementation is much more challenging. In particular, there are hardly any systems that fully support such OMQs when the actual queries are (U)CQs. In contrast, the evaluation of OMQs based on (expressive DLs and) IQs is supported by several systems such as Pellet, Hermit, and PAGoDa ([Sirin et al., 2007; Glimm et al., 2014; Zhou et al., 2015]). For this reason, rewriting (U)CQs into IQs has been advocated in ([Zolin, 2007; Kikot and Zolin, 2013; Kikot et al., 2013]) as an approach towards efficient OMQ evaluation with expressive DLs and (U)CQs. The experiments and optimizations reported in ([Kikot et al., 2013]) show the potential (and challenges) of this approach.

The second motivation stems from the connection between OMQs and constraint satisfaction problems (CSPs) ([Bienvenu et al., 2014; Lutz and Wolter, 2017]). Let $\langle \mathcal{L}, \mathcal{Q} \rangle$ denote the class of OMQs based on ontologies formulated in the DL $\mathcal{L}$ and the query language $\mathcal{Q}$. It was observed in ([Bienvenu et al., 2014]) that $\mathcal{ALCI}$ is closely related to the complement of CSPs while $\mathcal{ALCI}$, UCQ is closely related to the complement of the logical generalization MM-
SNP of CSP; we further remark that MMSNP is a notational variant of the complement of (Boolean) monadic disjunctive Datalog. Thus, characterizing OMQs from (\(\mathcal{ALCI}, \text{UCQ}\)) that are rewritable into (\(\mathcal{ALCI}, \text{IQ}\)) is related to characterizing MMSNP sentences that are equivalent to a CSP, and we also study the latter problem. In fact, the main differences to the OMQ case are that unary queries are replaced with Boolean ones and that predicates can have unrestricted arity.

The main aim of this paper is to study the rewritability of OMQs from (\(\mathcal{L}, (\mathcal{U}\text{CQ})\)) into OMQs from (\(\mathcal{L}, \text{IQ}\)), considering as \(\mathcal{L}\) the basic expressive DL \(\mathcal{ALC}\) as well as extensions of \(\mathcal{ALC}\) with inverse roles, role hierarchies, the universal role, and functional roles. We provide precise characterizations, tight complexity bounds for deciding whether a given OMQ is rewritable, and show how to construct the rewritten query when it exists. In fact, we prove that the classes of CQs from \([\text{Kikot and Zolin, 2013}]\) are depleting, but we go significantly beyond that: while \([\text{Zolin, 2007}]; \text{Kikot and Zolin, 2013}; \text{Kikot et al., 2013}\) aim to find IQ-rewritings that work for any ontology, we consider the more fine-grained question of rewriting into an IQ an OMQ (\(T, \Sigma, q(x)\)) where \(T\) is a DL TBox formalizing the ontology, \(\Sigma\) is an ABox signature, and \(q(x)\) is the actual query. The ‘any ontology’ setup then corresponds to the special case where \(T\) is empty and \(\Sigma\) is full. However, giving a non-empty TBox or a non-full ABox signature results in additional (\(U\))CQs to become rewritable. While we admit modification of the TBox during rewriting, it turns out that this is mostly unnecessary: only in some rather special cases, a moderate extension of the TBox pays off. All this requires non-trivial generalizations of the query classes and IQ-constructions from \([\text{Kikot and Zolin, 2013}]\). Our completeness proofs involve techniques that stem from the connection between OMQs and CSP such as a lemma about ABoxes of high girth due to Feder and Vardi \([1998]\). The rewritings we construct are of polynomial size when we work with the empty TBox, but can otherwise become exponential in size.

Regarding IQ-rewritability as a decision problem, we show NP-completeness for the case of the empty TBox. This can be viewed as an underapproximation for the case with non-empty TBox and ABox signature. With non-empty TBoxes, complexities are higher. When the ABox signature is full, we obtain \(2\text{ExpTime}\)-completeness for DLs with inverse roles and an \(\text{ExpTime}\) lower bound and a \(\text{ConExpTime}\) upper bound for DLs without inverse roles. With unrestricted ABox signature, the problem is \(2\text{NExpTime}\)-complete for DLs with inverse roles and \(\text{NExpTime}\)-hard (and in \(2\text{NExpTime}\)) for DLs without inverse roles. All lower bounds hold for CQs and all upper bounds capture UCQs. We also prove that it is \(2\text{NExpTime}\)-complete to decide whether a given MMSNP sentence is equivalent to a CSP. This problem was known to be decidable \([\text{Madelaine and Stewart, 2007}]\), but the complexity was open.

We also consider \(\mathcal{ALCF}, \mathcal{ALCI}\), the extension of \(\mathcal{ALCI}\) with functional roles, for which IQ-rewritability turns out to be undecidable and much harder to characterize. We give a rather subtle characterization for the case of the empty TBox and full ABox signature and show that the decision problem is then decidable and NP-complete. Since it is not clear how to apply CSP techniques, we use an approach based on ultrafilters, starting from what was done for \(\mathcal{ALC}\) without functional roles in \([\text{Kikot and Zolin, 2013}]\).

Full proofs are in the appendix.

## 2 Preliminaries

We use standard description logic notation and refer to \([\text{Baader et al., 2017}]\) for full details. In contrast to the standard DL literature, we carefully distinguish between the concept language and the TBox language. We consider four concept languages. Recall that \(\mathcal{ALC}\)-concepts are formed according to the syntax rule

\[
C, D ::= A | \neg C | C \sqcap D | C \sqcup D | \exists r. C | \forall r. C
\]

where \(A\) ranges over concept names and \(r\) over role names. As usual, we use \(C \rightarrow D\) as an abbreviation for \(\neg C \sqcup D\). \(\mathcal{ALCI}\)-concepts additionally admit the use of inverse roles \(r^{-}\) in concept constructors \(\exists r^{-}. C\) and \(\forall r^{-}. C\). With a role, we mean a role name or an inverse role. \(\mathcal{ALCFI}\)-concepts additionally admit the use of the universal role \(u\) in concept constructors \(\exists u. C\) and \(\forall u. C\). In \(\mathcal{ALCI^u}\)-concepts, both inverse roles and the universal role are admitted.

We now introduce several TBox languages. For \(\mathcal{L}\) one of the four concept languages introduced above, an \(\mathcal{L}\)-TBox is a finite set of concept inclusions \(C \sqsubseteq D\) where \(C\) and \(D\) are \(\mathcal{L}\) concepts. So each concept language also serves as a TBox language, but there are additional TBox languages of interest. We include the letter \(H\) in the name of a TBox language to indicate that role inclusions \(r \sqsubseteq s\) are also admitted in the TBox and likewise for the letter \(F\) and \(G\). As usual, the semantics is defined in terms of interpretations, which take the form \(I = (\Delta^T, \mathcal{I})\) with \(\Delta^T\) a non-empty domain and \(\mathcal{I}\) an interpretation function. An interpretation is a model of a TBox \(T\) if it satisfies all inclusions and assertions in \(T\), defined in the usual way. We write \(T \models r \sqsubseteq s\) if every model of \(T\) also satisfies the role inclusion \(r \sqsubseteq s\).

An ABox is a set of concept assertions \(A(a)\) and role assertions \(r(a, b)\) where \(A\) is a concept name, \(r\) a role name, and \(a, b\) are individual names. We use \(\text{ind}(A)\) to denote the set of all individual names that occur in \(A\). An interpretation is a model of an ABox \(A\) if it satisfies all concept and role assertions in \(A\), that is, \(a \in A^u\) when \(A(a)\) is in \(A\) and \(a, b \in r^2\) when \(r(a, b)\) is in \(A\). An ABox is consistent with a TBox \(T\) if \(A\) and \(T\) have a common model. A signature \(\Sigma\) is a set of concept and role names. We use \(\text{sig}(T)\) to denote the set of concept and role names that occur in the TBox \(T\), and likewise for other syntactic objects such as ABoxes. A \(\Sigma\)-ABox is an ABox \(A\) such that \(\text{sig}(A) \subseteq \Sigma\).

A conjunctive query (CQ) is of the form \(q(x) = \exists y. \varphi(x, y)\), where \(x, y\) are variables and \(\varphi(x, y)\) is a conjunction of atoms of the form \(A(x)\) or \(r(x, y)\) with \(A\) a concept name, \(r\) a role name, and \(x, y\) variables. We call \(x, y\) the answer variables of \(q(x)\) in \(q(x)\).
For purposes of uniformity, we use \( r^-(x, y) \) as an alternative notation to denote an atom \( r(y, x) \) in a CQ. In fact, when speaking about an atom \( r(x, y) \) in a CQ \( q(x) \), we generally also include the case that \( r = s^- \) and \( s(y, x) \) is the actual atom in \( q(x) \), unless explicitly noted otherwise. Every CQ \( q(x) = \exists y \, \varphi(x, y) \) gives rise to a directed graph \( G_q \) whose nodes are the elements of \( x \cup y \) and that contains an edge from \( x \) to \( y \) if \( \varphi(x, y) \) contains an atom \( r(x, y) \). The corresponding undirected graph is denoted \( G_q^u \) (it might contain self loops).

We can thus use standard terminology from graph theory to CQs, saying for example that a CQ is connected. A homomorphism from \( q(x) \) to an interpretation \( \mathcal{I} \) is a function \( h : x \cup y \rightarrow A^\mathcal{I} \) such that \( h(x) \in A^\mathcal{I} \) for every atom \( A(x) \) of \( q(x) \) and \( (h(x), h(y)) \in r^\mathcal{I} \) for every atom \( r(x, y) \) of \( q(x) \).

We write \( \mathcal{I} \models q(a) \) and call \( a \) an answer to \( q(x) \) on \( \mathcal{I} \) if there is a homomorphism from \( q(x) \) to \( \mathcal{I} \) with \( h(x) = a \).

A union of conjunctive queries (UCQ) \( q(x) \) is a disjunction of one or more CQs that all have the same answer variables \( x \). We say that a UCQ is connected if every CQ in it is. The arity of a \((U)\)CQ is the number of answer variables in it. For brevity, we generally write \( \{r, s\} \) and \( q(x) = \exists y \, r(x, y) \land s(y, x) \). The following examples show that IQ-rewritability depends on the topology of the actual query. Let \( q_1(x) = r(x, x) \). The OMQ \( \{q_1(x)\} \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) since \( r \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \).

(2) If we are not allowed to extend the TBox, IQ-rewritability depends on whether or not inverse roles are available. Let \( \Sigma = \{r, s\} \) and \( q(x) = \exists y \, r(x, y) \land s(y, x) \). The OMQ \( \{q(x)\} \) is rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) since \( r \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) but it is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) with \( T = \emptyset \).

(3) IQ-rewritability depends on the TBox. Let \( q(x) = \exists y_1 \exists y_2 \exists z_1 A(x, y_1) \land r(x, y_1) \land r(y_1, y_2) \land r(y_1, z) \land r(y_2, z) \land B_1(y_1) \land B_2(y_2) \). The OMQ \( \{\text{ALC}, \text{IQ}\} \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) but it is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \).

(4) IQ-rewritability depends on the ABox signature. Let \( q(x) \) be the OMQ \( \{\text{ALC}, \text{IQ}\} \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \). Let \( \Sigma = \{A\} \). The OMQ \( \{\text{ALC}, \text{IQ}\} \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \).

Note that we are allowed to completely rewrite the TBox when constructing IQ-rewritings, which might seem questionable from a practical perspective. Fortunately, though it turns out the TBox can always be left untouched or, in some rare cases, only needs to be slightly extended. Also note that an alternative definition of IQ-rewritability obtained by dropping the restriction to ABoxes consistent with \( T \) in Definition 3.1 all results obtained in this paper hold under both definitions. We comment on this throughout the paper and refer to the alternative version as unrestricted IQ-rewritability.

3 Characterizations

We aim to provide characterizations of OMQs that are IQ-rewritable. On the one hand, these characterizations clarify which OMQs are IQ-rewritable and which are not. On the other hand, they form the basis for deciding IQ-rewritability.

3.1 The Case With Inverse Roles

To state the characterization, we need some preliminaries. Let \( q(x) \) be a CQ. A cycle in \( q(x) \) is a sequence of non-identical atoms \( r_0(x_0, x_1), \ldots, r_{n-1}(x_{n-1}, x_n) \) in \( q(x) \), \( r \geq 1 \), where

1. \( r_0, \ldots, r_{n-1} \) are (potentially inverse) roles,
2. We require the atoms be non-identical to prevent \( r(x_0, x_1), r^-(x_1, x_0) \) from being a cycle (both atoms are identical).

Example 2. (1) IQ-rewritability depends on the topology of the actual query. Let \( q_1(x) = \exists y \, r(x, y) \). The OMQ \( \{q_1(x)\} \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \) since \( r \) is not rewritable into any OMQ from \( \{\text{ALC}, \text{IQ}\} \).
2. \( x_i \neq x_j \) for \( 0 \leq i < j \leq n \), and \( x_0 = x_n \).

The length of this cycle is \( n \). We say that \( q(x) \) is \( x \)-acyclic if every cycle in it passes through \( x \) and use \( q^{\text{acyc}}(x) \) to denote the result of restricting \( q(x) \) to those atoms that only use variables reachable in \( G_q^u \) from \( x \). Both notions are lifted to UCQs by applying them to every CQ in the UQC. A contraction of \( q(x) \) is a CQ obtained from \( q(x) \) by zero or more variable identifications, where the identification of \( x \) with any other variable yields \( x \).

Let \( T \) be an \( \text{ALCIT}^u \)-TBox and \( q(x) \) a UCQ. We use \( q_{\text{acyc}}(x) \) to denote the UCQ that consists of all \( x \)-acyclic CQs obtained by starting with a contraction of a CQ from \( q(x) \) and then replacing zero or more atoms \( r(y, z) \) with \( s(y, z) \) when \( T \models s \subseteq r \). We write \( q_{\text{acyc}}^\text{con}(x) \) to denote \( q_{\text{acyc}}(x) \) con.

**Theorem 3.** Let \( \mathcal{L} \in \{ \text{ALCIT}, \text{ALCHIT} \} \) and let \( Q = (T, \Sigma, q(x)) \) be a unary OMQ from \( (\mathcal{L}, \text{UCQ}) \) that is non-empty. Then the following are equivalent:

1. \( Q \) is IQ-rewriteable, that is, it is rewritable into an OMQ \( Q' = (T', \Sigma, C(x)) \) from \( (\mathcal{L}, \text{IQ}) \);
2. \( Q \) is rewritable into an OMQ \( Q' = (T, \Sigma, C(x)) \) from \( (\mathcal{L}, \text{IQ}) \);
3. \( Q \equiv (T, \Sigma, q_{\text{acyc}}^\text{con}(x)) \).

When \( \mathcal{L} \) is replaced with \( \mathcal{L}^u \), then the same equivalences hold except that \( q_{\text{acyc}}^\text{con}(x) \) is replaced with \( q_{\text{acyc}}(x) \).

Note that Theorem 3 excludes empty OMQs, but these are trivially IQ-rewriteable. It implies that, in the considered cases, it is never necessary to modify the TBox when constructing an IQ-rewriting. Further, it emerges from the proof that it is never necessary to introduce fresh role names in the rewriting (while fresh concept names are crucial). Theorem 3 also applies to unrestricted IQ-rewriteability (where also ABoxes are admitted that are inconsistent with the TBox from the OMQ): unrestricted IQ-rewriteability trivially implies IQ-rewriteability and the converse is an easy consequence of the fact that every OMQ that is IQ-rewriteable has an IQ-rewriting based on the same TBox.

We now give some ideas about the proof of Theorem 3. The most interesting implication is “1 \( \Rightarrow 3 \)”. A central step is to show that if \( Q = (T, \Sigma, q(x)) \) is IQ-rewriteable into an OMQ \( Q' \), then \( Q \subseteq q_{\text{acyc}}(x) \) for some \( \Sigma \)-ABox \( A \), then \( A \models q_{\text{acyc}}(x) \). To this end, we first construct from \( A \) a \( \Sigma \)-ABox \( A^g \) of high girth (that is, without small cycles) in a way such that (a) \( A^g \) homomorphically maps to \( A \) and (b) from \( A \models Q(a) \) it follows that \( A^g \models Q'(a) \), thus \( A^g \models Q(a) \). Due to the high girth of \( A^g \) and exploiting (a variation of the) tree model property for \( \text{ALCIT}^u \), we can then show that \( A^g \models Q(a) \) implies \( A^{\text{acyc}} \models Q_{\text{acyc}}(a) \). Because of (a), it follows that \( A \models Q_{\text{acyc}}(a) \). In the direction “3 \( \Rightarrow 2 \)”, we construct actual rewritings, based on the following lemma, an extension of a result of Kikot and Zolin [Kikot and Zolin, 2013] with TBoxes and ABox signatures (and UCQs instead of CQs).

**Lemma 4.** Let \( Q = (T, \Sigma, q(x)) \) be an OMQ from \( (\text{ALCIT}^u, \text{UCQ}) \). Then

1. if \( q(x) \) is \( x \)-acyclic and connected, then \( Q \) is rewritable into an OMQ \( (T, \Sigma, C(x)) \) with \( C(x) \) an \( \text{ALCIT}^u \)-IQ and
2. if \( q(x) \) is \( x \)-acyclic, then \( Q \) is rewritable into an OMQ \( (T, \Sigma, C(x)) \) with \( C(x) \) an \( \text{ALCIT}^u \)-IQ.

The size of the IQs \( C(x) \) is polynomial in the size of \( q(x) \).

We give the construction of the \( \text{ALCIT}^u \)-IQ \( q'(x) \) in Point 1 of Lemma 4. Let \( Q = (T, \Sigma, q(x)) \) be an OMQ from \( (\text{ALCIT}^u, \text{UCQ}) \) with \( q(x) \) \( x \)-acyclic and connected. To construct \( q'(x) \), we first construct for each CQ \( p(x) \) in \( q(x) \) an \( \text{ELI} \)-concept \( C_p \), that is, an \( \text{ALCIT} \)-concept that uses only the constructors \( \exists r.C \) and \( \exists r^- .C \). In fact, since \( p(x) \) is \( x \)-acyclic and connected, we can repeatedly choose and remove atoms of the form \( r(y, z) \) that occur in a cycle in \( p(x) \) and will eventually end up with a tree-shaped CQ \( p'(x) \). Here, tree-shaped means that the undirected graph \( G_p^u \) is a tree and that there are no multi-edges, that is, if \( r(y, z) \) is an atom, then there is no atom \( s(y, z) \) with \( s \neq r \). Next, extend \( p'(x) \) to obtain another tree-shaped CQ \( q'(x) \) by taking a fresh concept name \( P \notin \Sigma \), and adding \( r(x', y) \) and \( P(x') \) for each removed atom \( r(x', y) \). For \( x' \) a fresh variable. We can now view \( p'(x) \) as an \( \text{ELI} \)-concept \( C_p \) in the obvious way. The desired \( \text{ALCIT}^u \)-IQ \( q'(x) \) is \( (P \rightarrow p(x)) \text{ a CQ in } q(x) \).

### 3.2 The Case Without Inverse Roles

We consider OMQs whose TBoxes are formulated in a DL \( \mathcal{L} \) that does not admit inverse roles. Note that inverse roles are then also not admitted in the IQ used in the rewriting. We first observe that this has less impact than one might expect: inverse roles in the IQ-rewriting can be eliminated and in fact Points 1 and 3 from Theorem 3 are still equivalent. However, there is also a crucial difference: unless the universal role is present, the elimination of inverse roles requires an extension of the TBox and thus the equivalence of Points 1 and 2 of Theorem 3 fails. In fact, this is illustrated by Point (2) of Example 2. We thus additionally characterize IQ-rewritability without modifying the TBox. We also show that, with the universal role, it is not necessary to extend the TBox.

We start with some preliminaries. An extended conjunctive query (eCQ) is a CQ that also admits atoms of the form \( C(x) \), \( C \) a (potentially compound) concept, and \( \text{UCQs} \) and extended ABoxes (eABoxes) are defined analogously. The semantics is defined in the expected way. Every eCQ \( q(x) \) gives rise to an eabox \( A_q \) by viewing the variables in \( q(x) \) as individual names and the atoms as assertions.

Let \( q(x) \) be an eCQ. We use \( dreach(q) \) to denote the set of all variables reachable from \( x \) in the directed graph \( G_q^u \) and say that \( q(x) \) is \( x \)-accessible if \( dreach(q) \) contains all variables. For \( V \) a set of variables from \( q(x) \) that includes \( x \), \( q(x)|_V \) denotes the restriction of \( q(x) \) to the atoms that use only variables from \( V \).

Let \( T \) be an \( \text{ALC} \)-TBox. An eCQ \( p(x) \) is a \( T \)-decoration of a CQ \( q(x) \) if

1. \( p(x) \) is obtained from \( q(x) \) by adding, for each \( y \in dreach(q) \) and each subconcept \( C \) of \( T \), the atom \( C(y) \) or the atom \( \neg C(y) \);
2. the eABox \( A_q \) is consistent with \( T \).

---

Note that \( x \) is the answer variable and recall that we might have \( r = s^- \) and thus also choose atoms \( s(y, x) \).
For a UCQ \( q(x) \), we use \( q^{\text{deco}}(x) \) to denote the UCQ that consists of all eCQs \( p(x)\) where \( p(x) \) is a \( T \)-decoration of a CQ from \( q(x) \). We write \( q^{\text{acyc}}(x) \) to denote \((q_{\text{acyc}})^{\text{deco}}(x)\). We now give the results announced above.

**Theorem 5.** Let \( \mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALC}^n\} \) and let \( Q = (\mathcal{T}, \Sigma, q(x)) \) be a unary OMQ from \((\mathcal{L}, \mathcal{UCQ})\) that is non-empty. Then the following are equivalent:

1. \( Q \) is rewritable into an OMQ from \((\mathcal{L}, \mathcal{IQ})\);
2. \( Q \) is rewritable into an OMQ \((\mathcal{L} \cup \mathcal{T}' \cup \Sigma, C(x))\) from \((\mathcal{L}, \mathcal{IQ})\);
3. \( Q \) is rewritable into an OMQ from \((\mathcal{L}^I, \mathcal{IQ})\);
4. \( Q' \) is rewritable into an OMQ \( Q'' = (\mathcal{T}, \Sigma_{\text{full}}, C(x)) \) from \((\mathcal{L}, \mathcal{IQ})\);
5. \( Q \equiv (\mathcal{T}, \Sigma_{\text{full}}, q^{\text{acyc}}(x)) \).

If, furthermore, \( \mathcal{L} \) is replaced with \( \mathcal{L}^u \) and \( \mathcal{L}^I \) with \( \mathcal{L}^u \), then Conditions 1 to 3 are further equivalent to:

6. \( Q \) is rewritable into an OMQ \( Q' = (\mathcal{T}, \Sigma, C(x)) \) from \((\mathcal{L}^u, \mathcal{IQ})\).

Characterizing IQ-rewritability in the case where \( \mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALC}^n\} \), the TBox (is non-empty and) cannot be extended, and \( \Sigma \neq \Sigma_{\text{full}} \) remains an open problem.

In the directions “\( 3 \Rightarrow 2 \)”, “\( 5 \Rightarrow 4 \)” and “\( 3 \Rightarrow 6 \)”, we have to construct IQ-rewritings. This is done by starting with the rewriting from the proof of Lemma 3 and then modifying it appropriately. As in the case of Theorem 3, it is straightforward to see that all results stated in Theorem 5 also apply to unrestricted IQ-rewritability.

**3.3 The Case of Empty TBoxes**

We consider OMQs in which the TBox is empty as an important special case. Since it is then not interesting to have an ABox signature, this corresponds to the rewritability of (U)CQs into \( \mathcal{L} \)-instance queries, for some concept language \( \mathcal{L} \) (and thus no OMQs are involved). The importance of this case is due to the fact that it provides an ‘underapproximation’ of the IQ-rewritability of OMQs, while also being easier to characterize and computationally simpler.

We say that an UCQ \( q(x) \) is \( \mathcal{L} \)-IQ-rewritable if there is an \( \mathcal{L} \)-IQ \( q'(x) \) that is equivalent to \( q(x) \) in the sense that the OMQs \((\emptyset, \Sigma_{\text{full}}, q(x))\) and \((\emptyset, \Sigma_{\text{full}}, q'(x))\) are equivalent (and in passing, we define the equivalence between two UCQs in exactly the same way). The following proposition makes precise what we mean by underapproximation.

**Proposition 6.** Let \( \mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALC}^I, \mathcal{ALC}^n, \mathcal{ALC}^I^n\} \). If a UCQ \( q(x) \) is \( \mathcal{L} \)-IQ-rewritable, then so is any OMQ \((\mathcal{T}, \Sigma, q(x))\) from \((\mathcal{L}^H, \mathcal{UCQ})\).

Proposition 6 is essentially a corollary of Theorem 7 below. As illustrated by Case (3) of Example 2, its converse fails.

We now characterize IQ-rewritability in the case of the empty TBox. A subquery of a CQ \( q(x) \) is a CQ \( q'(x) \) obtained from \( q(x) \) by dropping atoms. A subquery of a UCQ \( q(x) \) is a UCQ obtained by including as a UCQ at most one subquery of each CQ in \( q(x) \).

**Theorem 7.** Let \( q(x) \) be a UCQ. Then

1. \( q(x) \) is rewritable into an \( \mathcal{ALCI} \)-IQ iff there is a subquery \( q'(x) \) of \( q(x) \) that is x-acyclic, connected, and equivalent to \( q(x) \);
2. \( q(x) \) is rewritable into an \( \mathcal{ALC}^I \)-IQ iff there is a subquery \( q'(x) \) of \( q(x) \) that is x-acyclic, x-accessible, and equivalent to \( q(x) \).

When \( \mathcal{L} \)-IQs are replaced with \( \mathcal{L}^u \)-IQs, then the same equivalences hold except that connectedness/\( x \)-accessibility is dropped.

Note that Theorem 7 also characterizes rewritability of CQs; the query \( q'(x) \) is then also a CQ rather than a UCQ. This is in contrast to Theorems 3 and 5 where the queries \( q^{\text{con}}(x) \) and \( q^{\text{acyc}}(x) \) are UCQs even when the query \( q(x) \) from the OMQ that we start with is a CQ. Another crucial difference is that \( q^{\text{con}}(x) \) and \( q^{\text{acyc}}(x) \) can be of size exponential in the size of the original OMQ while the query \( q'(x) \) in Theorem 7 is of size polynomial in the size of \( q(x) \).

**4 Complexity**

We determine the complexity of deciding IQ-rewritability in various OMQ languages, based on the established characterizations and starting with the case of empty TBoxes.

**Theorem 8.** For every \( Q \in \{\mathcal{CQ}, \mathcal{UCQ}\} \) and \( \mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALC}^I, \mathcal{ALC}^n, \mathcal{ALC}^I^n\} \), it is NP-complete to decide whether a given query from \( Q \) is \( \mathcal{L} \)-IQ-rewritable.

The upper bound in Theorem 8 is by guessing the query \( q'(x) \) from Theorem 7 and verifying that it satisfies the properties stated there. The lower bound is by a reduction from 3-colorability.

We next consider the case where TBoxes can be non-empty, starting with the assumption that the ABox signature is full since this results in (slightly) lower complexity.

**Theorem 9.** Let \( Q \in \{\mathcal{CQ}, \mathcal{UCQ}\} \). For OMQs based on the full ABox signature, IQ-rewritability is

1. \( \text{ExpTime-hard in } \{\mathcal{ALC}, \mathcal{Q}\} \) and in \( \text{ConExpTime in } \{\mathcal{ALC}^I, \mathcal{Q}\} \) and
2. \( 2\text{ExpTime-complete in } \{\mathcal{ALC}^I, \mathcal{Q}\} \) and \( \{\mathcal{ALC}^I, \mathcal{Q}\} \).

The lower bounds are by reduction from OMQ evaluation on ABoxes of the form \( \{A(a)\} \) where \( A \) is a concept name, which is \( \text{ExpTime-complete in } \{\mathcal{ALC}, \mathcal{Q}\} \) and \( 2\text{ExpTime-complete in } \{\mathcal{ALC}^I, \mathcal{Q}\} \) \cite{Lutz08}. The upper bounds are derived from the OMQ containment checks suggested by Condition 3 of Theorem 3 and Condition 4 of Theorem 5. Since we work with the full ABox signature, the non-emptiness condition from these theorems is void (there are no empty OMQs) and OMQ containment is closely related to OMQ evaluation, which allows us to derive upper bounds for the former from the latter; in fact, these bounds are exactly the ones stated in Theorem 9. We have to exercise some care, for two reasons: first, we admit UCQs as the actual query and thus the trivial reduction of OMQ containment to OMQ evaluation that is possible for CQs (which can be viewed as an ABox) does not apply. And second, we aim for upper bounds that exactly match the complexity of OMQ containment while
Theorem 10. IQ-rewritability is 

1. NEXPTIME-hard in \((\text{ALC}, \text{CQ})\) and
2. 2NEXPTIME-complete in all of \((\text{ALCIF}, \text{CQ})\), \((\text{ALCIT}, \text{UCQ})\), \((\text{ALCHI}, \text{CQ})\), \((\text{ALCHI}, \text{UCQ})\).

The lower bound in Point 1 is by reduction from OMQ emptiness in \((\text{ALC}, \text{CQ})\), which is NEXPTIME-complete \cite{Baader2016}. For the one in Point 2, we use a reduction from OMQ containment, which is 2NEXPTIME-complete in \((\text{ALCIT}, \text{CQ})\) \cite{BourhisLutz2016}. The upper bounds are obtained by appropriate containment checks as suggested by our characterizations, and we again have to deal with UCQs with exponentially many CQs. Note that Theorem 10 leaves open the complexity of IQ-rewritability in \((\text{ALC}, \text{CQ})\), between NEXPTIME and 2NEXPTIME. The same gap exists for OMQ containment \cite{BourhisLutz2016} as well as in the related problems of FO-rewritability and Datalog-rewritability \cite{FeierLutz2017}.

5 Functional Roles

We consider DLs with functional roles. A fundamental observation is that for the basic such DL \text{ALCF}, IQ-rewritability is undecidable. This can be proved by a reduction from OMQ emptiness in \((\text{ALCF}, \text{IQ})\) \cite{Baader2016}.

Theorem 11. In \((\text{ALCF}, \text{CQ})\), IQ-rewritability is undecidable.

In the following, we show that decidability is regained in the case where the TBox is empty (apart from functionality assertions). This is challenging because functionality assertions have a strong and subtle impact on rewritability. As before, the only interesting ABox signature to be combined with ‘empty’ TBoxes is the full ABox signature. We use \(\mathcal{F}\) to denote the TBox language in which TBoxes are sets of functionality assertions and concentrate on rewriting into IQs that may use inverse roles.

Example 12. Consider the CQ \(p(x) = \exists y (s(x, y) \land r(y, y))\) from Point 1 of Example 2. Then \(Q_s = (\mathcal{T}_s, \Sigma_{\text{full}}, p(x))\) and \(Q_r = (\mathcal{T}_r, \Sigma_{\text{full}}, p(x))\) with \(\mathcal{T}_w = \{\text{func}(w)\}\) for \(w \in \{r, s\}\) are both rewrivable into an OMQ \((\mathcal{T}_w, \Sigma_{\text{full}}, q_w(x))\) with \(q_w(x)\) an \text{ALCIT}-IQ. The rewrites are neither trivial to find nor entirely easy to understand. In fact, for \(q_w(x)\) we can use \(\forall s.P \rightarrow \exists s.(P \rightarrow \exists r.P)\). For \(q_r(x)\), we introduce three fresh concept names rather than a single one and use them in a way inspired by graph colorings:

\[q_r(x) = (\forall s. \bigcup_{1 \leq i \leq 3} P_i) \rightarrow (\exists s.(\bigcap_{1 \leq i \leq 3} (P_i \rightarrow \exists r.P_i))).\]

Before giving a characterization of rewritable queries, we introduce some preliminaries. Let \(q(x)\) be a CQ and \(\mathcal{T}\) an \text{ALCIT}-TBox. A sequence \(x_0, \ldots, x_n\) of variables in \(q(x)\) is a functional path in \(q(x)\) from \(x_0\) to \(x_n\) w.r.t. \(\mathcal{T}\) if for all \(i < n\) there is a role \(r\) such that \(\text{func}(r) \in \mathcal{T}\) and \(r(x_i, x_{i+1})\) is in \(q(x)\). We say that \(q(x)\) is \(f\)-acyclic w.r.t. \(\mathcal{T}\) if for every cycle \(r_0(x_0, x_1), \ldots, r_{n-1}(x_{n-1}, x_n)\) in \(q(x)\), one of the following holds:

- there is a functional path in \(q(x)\) from \(x\) to some \(x_i\);
- \(\text{func}(r_i) \in \mathcal{T}\) or \(\text{func}(r_i^{-1}) \in \mathcal{T}\) for all \(i < n\) and there is a functional path \(y_0, \ldots, y_m\) in \(q(x)\) with \(x_0 = y_0 = y_m\) such that \([x_0, \ldots, x_{n-1}] \subseteq [y_0, \ldots, y_m]\).

We are now ready to state the characterization.

Theorem 13. An OMQ \(Q = (\mathcal{T}, \Sigma_{\text{full}}, q(x))\) from \((\mathcal{F}, \text{UCQ})\) is rewritable into an OMQ from \((\mathcal{F}, \text{ALCIT}-\text{IQ})\) iff there is a subquery \(q'(x)\) of \(q(x)\) that is \(f\)-acyclic, connected, and equivalent to \(q(x)\).

When \text{ALCIT}-IQ is replaced with \text{ALCF}-IQ, the same equivalence holds except that connectedness is dropped.

The proof of Theorem 13 extends the ultrafilter construction from \cite{KikotZolin2013}. We remark that the “if” direction in Theorem 13 even holds for OMQs \(Q = (\mathcal{T}, \Sigma, q(x))\) from \((\mathcal{F}, \text{ALCIT}, \text{UCQ})\) using an MMSNP-sentence is equivalent to a CSP. In a nutshell, we prove that an MMSNP-sentence is equivalent to a CSP iff it is preserved under disjoint union and equivalent to a generalized CSP (a CSP with multiple templates), and that both properties can be reduced to containment between MMSNP sentences which is 2NEXPTIME-complete \cite{BourhisLutz2016}. The latter reduction involves constructing an MMSNP sentence \(\varphi_{\text{acyc}}\) that is reminiscent of the query \(q_{\text{acyc}}\) in Theorem 3. Full details are given in the appendix.

Theorem 15. It is 2NEXPTIME-complete to decide whether a given MMSNP-sentence is equivalent to a CSP.

7 Conclusion

We made a step forward in understanding the relation between (U)CQs and IQs in ontology-mediated querying. Interesting open problems include a characterization of IQ-rewritability for DLs with functional roles when the TBox is non-empty and characterizations for DLs with transitive roles. The remarks after Theorem 4 and 10 mention further problems left open. It would also be worthwhile to follow \cite{Kikot2013} and understand the value of IQ-rewritings for the purposes of efficient practical implementation.

Acknowledgements. Cristina Feier and Carsten Lutz were supported by ERC Consolidator Grant 647289 CODA. Frank Wolter was supported by EPSRC UK grant EP/M012646/1.
References

[Baader et al., 2016] Franz Baader, Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query and predicate emptiness in ontology-based data access. *J. Artif. Intell. Res.*, 56:1–59, 2016.

[Baader et al., 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[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. Springer, 2015.

[Bienvenu et al., 2012] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query containment in description logics reconsidered. In *Proc. of KR2012*, 2012.

[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. *ACM Trans. Database Syst.*, 39(4):33:1–33:44, 2014.

[Blackburn et al., 2002] Patrick Blackburn, Marten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2002.

[Bourhis and Lutz, 2016] Pierre Bourhis and Carsten Lutz. Containment in monadic disjunctive datalog, MMSNP, and expressive description logics. In *Proc. of KR*, pages 207–216. AAAI Press, 2016.

[Calvanese et al., 1998] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS1998*, pages 149–158. ACM Press, 1998.

[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 2009*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.

[Eiter et al., 2012a] Thomas Eiter, Magdalena Ortiz, and Mantas Simkus. Conjunctive query answering in the description logic SH using knots. *J. Comput. Syst. Sci.*, 78(1):47–85, 2012.

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

[Feder and Vardi, 1998] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[Feier et al., 2017] Cristina Feier, Antti Kuusisto, and Carsten Lutz. Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics (invited talk). In *Proc. ICDT*, volume 68 of *LIPIcs*, pages 1:1–1:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Glimm et al., 2008] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic SHIQ. *J. Artif. Intell. Res.*, 31:157–204, 2008.

[Glimm et al., 2014] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: An OWL 2 reasoner. *J. of Autom. Reasoning.*, 53(3):245–269, 2014.

[Kikot and Zolin, 2013] Stanislav Kikot and Evgeny Zolin. Modal definability of first-order formulas with free variables and query answering. *J. Applied Logic*, 11(2):190–216, 2013.

[Kikot et al., 2013] Stanislav Kikot, Dmitry Tsarkov, Michael Zakharyaschev, and Evgeny Zolin. Query answering via modal definability with FaCT++: First blood. In *Proc. DL*, volume 1014 of *CEUR Workshop Proceedings*, pages 328–340. CEUR-WS.org, 2013.

[Lutz and Wolter, 2017] Carsten Lutz and Frank Wolter. The data complexity of description logic ontologies. *Logical Methods in Computer Science*, 13(4), 2017.

[Lutz et al., 2014] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. IJCAI*, pages 2070–2075, 2019.

[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. Springer, 2008.

[Madelaine and Stewart, 2007] Florent R. Madelaine and Iain A. Stewart. Constraint satisfaction, logic and forbidden patterns. *SIAM J. Comput.*, 37(1):132–163, 2007.

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

[Schild, 1991] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. IJCAI*, pages 466–471, 1991.

[Sirin et al., 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2):51 – 53, 2007.

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

[Zhao et al., 2015] Yujiao Zhou, Bernardo Cuenca Grau, Yavor Nenov, Mark Kaminski, and Ian Horrocks. Pagoda: Pay-as-you-go ontology query answering using a datalog reasoner. *J. Artif. Intell. Res.*, 54:309–367, 2015.

[Zolin, 2007] Evgeny Zolin. Modal logic applied to query answering and the case for variable modalities. In *Proc. of DL*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
Appendices

A Some Technical Preliminaries

Every interpretation $\mathcal{I}$ can be viewed as an undirected graph $G^*_\mathcal{I}$, analogously to the definition of the undirected graph $G^*_A$ of a CQ $q$. The universal role does not give rise to edges in $G^*_\mathcal{I}$.

An interpretation is tree-shaped or a tree interpretation if $G^*_\mathcal{I}$ is a tree and there are no multi-edges, that is, $(d, e) \in r^T$ implies $(d, e) \notin s^T$ for all (potentially inverse) roles $s \neq r$. Let $\mathcal{T}$ be an $\mathcal{ALCHI}$-TBox and $\mathcal{A}$ an ABox. An interpretation $\mathcal{I}$ is a forest model of $\mathcal{A}$ if there are tree interpretations $(\mathcal{I}_A)_{a \in \text{ind}(\mathcal{A}) \cup D}$, where $D$ is a (potentially empty) set of individuals, with mutually disjoint domains, and

$$\Delta_{\mathcal{I}_A} \cap \text{ind}(\mathcal{A}) = \begin{cases} \{a\}, & \text{if } a \in \text{ind}(\mathcal{A}) \\ \emptyset, & \text{if } a \in D, \end{cases}$$

such that $\mathcal{I}$ is the (non-disjoint) union of $\mathcal{I}_A$ and $(\mathcal{I}_A)_{a \in \text{ind}(\mathcal{A}) \cup D}$ where $\mathcal{I}_A$ is $\mathcal{A}$ viewed as an interpretation. An extended forest model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ is a model of $\mathcal{A}$ and $\mathcal{T}$ that can be obtained from a forest model $\mathcal{J}$ of $\mathcal{A}$ by closing under role inclusions from $\mathcal{T}$, that is, adding $(d, e) \in r^T$ when $(d, e) \in s^T$ and $\mathcal{T} \models s \subseteq r \in \mathcal{T}$. We also say that $\mathcal{J}$ underlies $\mathcal{I}$.

Lemmas of the following kind have been widely used in the literature on ontology-mediated querying. The proof of the "if" direction uses a standard unraveling argument and is omitted, see for example [Lutz, 2008].

Lemma 16. Let $\mathcal{Q} = (T, \Sigma, \text{q}(x))$ be an OMQ from $\mathcal{ALCHI}$, $\mathcal{A}$ a $\Sigma$-ABox, and $a \subseteq \text{ind}(\mathcal{A})$. Then $\mathcal{A} \models \mathcal{Q}(a)$ iff for all extended forest models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$, $\mathcal{I} \models \mathcal{Q}(a)$.

We introduce some more helping lemmas. An ABox $\mathcal{A}$ can be seen as a directed graph $G_\mathcal{A}$ and as an undirected graph $G^*_\mathcal{A}$ in the expected way, analogously to the definition of $G_q$ and $G^*_q$ for a CQ $q$. For an ABox $\mathcal{A}$ and $a \in \text{ind}(\mathcal{A})$, we use $\mathcal{A}^\text{con}_a$ to denote the restriction of $\mathcal{A}$ to the individuals reachable in $G^*_\mathcal{A}$ from $a$. We also denote with $\text{CON}_\mathcal{A}$ the set of ABoxes induced by the maximal connected components of $G^*_\mathcal{A}$.

Lemma 17. Let $\mathcal{Q} = (T, \Sigma, \text{q}(x))$ be an OMQ from $\mathcal{ALCHI}$, $\mathcal{Q}$. Then $\mathcal{A} \models \mathcal{Q}(a)$ implies $\mathcal{A}^\text{con}_a \models \mathcal{Q}(a)$.

A homomorphism from an ABox $\mathcal{A}$ to an ABox $\mathcal{B}$ is a function $h : \text{ind}(\mathcal{A}) \to \text{ind}(\mathcal{B})$ such that $A(a) \in \mathcal{A}$ implies $h(A(a)) \in \mathcal{B}$ and $r(a, b) \in \mathcal{A}$ implies $r(h(a), h(b)) \in \mathcal{B}$. We write $\mathcal{A} \to \mathcal{B}$ to indicate that there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. For $a \in \text{ind}(\mathcal{A})$ and $b \in \text{ind}(\mathcal{B})$, we further write $(\mathcal{A}, a) \to (\mathcal{B}, b)$ to indicate that there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ with $h(a) = b$. The following lemma is well-known, see for example [Bienvenu et al., 2014].

Lemma 18. Let $\mathcal{Q} = (T, \Sigma, \text{q}(x))$ be a unary OMQ from $\mathcal{ALCHI}$, $\mathcal{Q}$, with $\mathcal{Q} \in \mathcal{UCQ}$. Then $\mathcal{A} \models \mathcal{Q}(a)$ implies $\mathcal{A}^\text{con}_a \models \mathcal{Q}(a)$.

B Proofs for Section 3

We start with introducing several lemmas concerned with certain constructions on ABoxes. These lemmas are closely related to the connection between ontology-mediated querying and constraint satisfaction problems (CSPs), see for example [Bienvenu et al., 2014; Lutz and Wolter, 2017].

Note that in assertions $r(x, y)$ in an ABox, $r$ must be a role name but cannot be an inverse role. For purposes of uniformity, we use $r^-(x, y)$ as an alternative notation to denote an assertion $r(y, x)$ in an ABox. A cycle in an ABox is defined exactly like a cycle in a CQ, repeated here for convenience. A cycle in an ABox $\mathcal{A}$ is a sequence of non-identical assertions $r_0(a_0, a_1), \ldots, r_{n-1}(a_{n-1}, a_n)$ in $\mathcal{A}$, $n \geq 1$, where

1. $r_0, \ldots, r_{n-1}$ are (potentially inverse) roles,
2. $a_i \neq a_j$ for $0 \leq i < j < n$, and $a_0 = a_n$.

The length of this cycle is $n$. The girth of $\mathcal{A}$ is the length of the shortest cycle in it and $\infty$ if $\mathcal{A}$ has no cycle.

The following is a DL formulation of what is often known as the sparse incomparability lemma in CSP [Feder and Vardi, 1998].

Lemma 19. For every ABox $\mathcal{A}$ and all $g, s \geq 0$, there is an ABox $\mathcal{A}^g$ of girth exceeding $g$ such that

1. $\mathcal{A}^g \to \mathcal{A}$ and
2. for every ABox $\mathcal{B}$ with $|\text{ind}(\mathcal{B})| \leq s$, $\mathcal{A} \to \mathcal{B}$ iff $\mathcal{A}^g \to \mathcal{B}$.

We next establish a ‘pointed’ version of Lemma 19 that is crucial for the subsequent proofs. The a-girth of $\mathcal{A}$ is defined exactly like the girth except that we only consider cycles that do not pass through $a$.

Lemma 20. For all ABoxes $\mathcal{A}$, $a \in \text{ind}(\mathcal{A})$, and $g, s \geq 0$, there is an ABox $\mathcal{A}^g$ of a-girth exceeding $g$ such that

1. $(\mathcal{A}^g, a) \to (\mathcal{A}, a)$
2. for every ABox $\mathcal{B}$ with $|\text{ind}(\mathcal{B})| \leq s$ and every $b \in \text{ind}(\mathcal{B})$, $(\mathcal{A}, a) \to (\mathcal{B}, b)$ iff $(\mathcal{A}^g, a) \to (\mathcal{B}, b)$.

Proof. Let $\mathcal{A}$ be an ABox, $a \in \text{ind}(\mathcal{A})$, and $g, s \geq 0$. Further, let $\mathcal{A}_a$ be the ABox obtained from $\mathcal{A}$ by adding the assertion $P(a)$, $P$ a fresh concept name, let $\mathcal{A}_a^g$ be the ABox obtained from $\mathcal{A}_a$ by applying Lemma 19 for $g$ and $s$, and let $h$ be a homomorphism from $\mathcal{A}_a^g$ to $\mathcal{A}_a$. Assume w.l.o.g. that the individual name $a$ does not occur in $\mathcal{A}_a^g$. We use $\mathcal{A}^g$ to denote the ABox obtained from $\mathcal{A}_a^g$ by dropping all facts of the form $P(b)$ and identifying all individual names $b$ with $h(b) = a$, replacing them with $a$. We show that $\mathcal{A}^g$ is as required:

(a) $\mathcal{A}^g$ has a-girth higher than $g$.

Every cycle in $\mathcal{A}^g$ that does not pass through $a$ is also in $\mathcal{A}_a^g$, thus is of length exceeding $g$.

(b) Point 1 of Lemma 20 is satisfied.

Let $h' : \text{ind}(\mathcal{A}^g) \to \text{ind}(\mathcal{A})$ be such that $h'(a) = a$ and $h'(b) = h(b)$ if $a \neq b$. It can be verified that $h'$ is a homomorphism from $\mathcal{A}^g$ to $\mathcal{A}$. It clearly witnesses $(\mathcal{A}^g, a) \to (\mathcal{A}, a)$, as required.

(c) Point 2 of Lemma 20 is satisfied.

Let $\mathcal{B}$ be an ABox with $|\text{ind}(\mathcal{B})| \leq s$ and $b \in \text{ind}(\mathcal{B})$. We have to show that $(\mathcal{A}, a) \to (\mathcal{B}, b)$ iff $(\mathcal{A}^g, a) \to (\mathcal{B}, b)$. We write $A(a) \in \mathcal{A}$, $A\in\mathcal{A}$, and $A \models \mathcal{Q}(a)$.
The “only if” direction is immediate by (b). For the “if” direction, assume that $(A^q, a) \rightarrow (B, b)$. Then $A^q \cup \{ P(a) \} \rightarrow B \cup \{ P(b) \}$. This implies $A^q \rightarrow B \cup \{ P(b) \}$ and by Point 2 of Lemma 1, also $A^q \rightarrow B \cup \{ P(b) \}$. As $P(a)$ is the only assertion of this form in $A^q$, it follows that $(A^q, a) \rightarrow (B \cup \{ P(b) \}, b)$. Thus $(A, a) \rightarrow (B, b)$.

The following lemma is a straightforward variation of similar lemmas from [Bienvenu et al., 2014]. The constructed ABoxes are called CSP templates in [Bienvenu et al., 2014].

**Lemma 21.** Let $Q = (T, \Sigma, q(x))$ be an OMQ from $(ALCHIT^n, UCQ)$. Then one can find a set $\Gamma$ of pairs $(B, b)$ with $B \subseteq A \Sigma$-ABox and $b \in \text{ind}(B)$ such that for every $A \Sigma$-Box $A$ and all $a \in \text{ind}(A),

1. $A \models Q(a)$ \iff $(A, a) \not\rightarrow (B, b)$ for all $(B, b) \in \Gamma$;
2. $A$ is consistent with $\Gamma$ \iff $A \rightarrow B$ for some $(B, b) \in \Gamma$.

When $Q$ is from $(ALCHT, IQ)$, then $\Gamma$ can be chosen so that all ABoxes in it are identical.

An $\mathcal{ECT}^\mathcal{u}$-concept is an $\mathcal{ECT}$-concept that uses only the constructors $\exists r.C, \exists r^- C, \exists u.C$ where $u$ is the universal role.

**Lemma 4.** Let $Q = (T, \Sigma, q(x))$ be an OMQ from $(ALCHIT^n, UCQ)$. Then

1. if $q(x)$ is $x$-acyclic and connected, then $Q$ is rewritable into an OMQ $(T', \Sigma, C(x))$ with $C(x)$ an $\mathcal{ECT}$-IQ and $T'$ contains $\Delta$.
2. if $q(x)$ is $x$-acyclic, then $Q$ is rewritable into an OMQ $(T', \Sigma, C(x))$ with $C(x)$ an $\mathcal{ECT}^\mathcal{u}$-IQ.

The size of the IBox $C(x)$ is polynomial in the size of $q(x)$.

**Proof.** For Point 1, let $Q = (T, \Sigma, q(x))$ be an OMQ from $(ALCHIT^n, UCQ)$ where $q(x)$ is $x$-acyclic and connected. Further, let $Q' = (T, \Sigma, (P \rightarrow p(x), C(x)))$ be as constructed in the main part of the paper. We have to show the following:

- “$Q \subseteq Q'$”: Let $A$ be a $\Sigma$-ABox with $A \models Q(a)$. Then, for every model $I$ of $T$ and $A$, $I \models p(a)$, for some concept $P(x)$ in $q(x)$, and thus $I \models P(a)$ or $I \models C(x) \land p(a)$. As $A^q$ is $\Sigma$-Box with $\{ P(x) \}$, the latter is the same as $I \models p'(a)$ or $I \models C'(a)$. Thus $I \models (P \rightarrow C'(a))$, or $I \models Q'(a)$.

- “$Q' \subseteq Q$”: Let $A$ be a $\Sigma$-ABox with $A \models Q'(a)$. Then, for every model $I'$ of $T$ and $A$, $I' \models (P \rightarrow p(x)) C(x) \land p(a)$. Then, there must be a model $I'' = (\Delta, I')$ of $T$ and $A$ (possibly the same as $I'$) such that $P I'' = \{ a \}$ and $I'' \models I''$ coincide on all other symbols. We have $I'' \models (P \rightarrow p(x)) C(x) \land p(a)$, thus $I'' \models C'(a)$, for some concept $C(x)$ in $q(x)$. From the construction of $C'(a)$, $I'' \models p'(a)$, and from the fact that $P$ is interpreted as a singleton in $I''$, $I'' \models p(a)$, or $I'' \models Q(a) \iff (P \rightarrow p(x)) C(x) \land p(a)$. As $P$ is fresh, and in particular does not occur in $q$, and $I' \models I''$ might differ only w.r.t. the interpretation of $P$. $I' \models Q'(a)$ iff $(A, a) \not\rightarrow (B, b)$ for $1 \leq i \leq k$;
2. $A$ is consistent with $\Gamma$ \iff $A \rightarrow B$.

We show Points (a) and (b) from above.

(a) We have $Q_{\text{acyc}} \subseteq Q$ by definition of $Q_{\text{acyc}}$, no matter whether $Q$ is IQ-rewritable or not, and thus it remains to show that $Q \subseteq Q_{\text{acyc}}$. If all COQs in $q(x)$ are $x$-acyclic, the result clearly holds. In the following we assume that at least one COQ in $q(x)$ is not $x$-acyclic.
Let \( \mathcal{A} \) be a \( \Sigma \)-ABox with \( \mathcal{A} \models Q(\alpha) \). Thus, \( (A, a) \not\models (B, b) \) for \( 1 \leq i \leq k \). We apply Lemma 20 with \( g \) the maximum between 2 and the girths of CQs from \( q(x) \) which are not \( x \)-acyclic, obtaining a \( \Sigma \)-ABox \( \mathcal{A}' \) of \( g \)-girth exceeding \( g \) such that \( (A', a) \rightarrow (A, a) \) and \( (A', a) \not\models (B', b) \) for \( 1 \leq i \leq k \). The latter yields \( S' \models Q(a) \). We aim to show that \( S' \models Q_{\text{acyc}}(a) \). Since \( (A', a) \rightarrow (A, a) \), it follows by Lemma 10 that \( S \models Q_{\text{acyc}}(a) \), as desired.

By Lemma 16, it suffices to show that for every extended forest model \( \mathcal{I} \) of \( \mathcal{A}' \) and \( \mathcal{T} \), we have \( \mathcal{I} \models q_{\text{acyc}}(a) \). Thus let \( \mathcal{I} \) be such a model. Since \( S' \models Q(a) \), we have \( \mathcal{I} \models q(a) \) and thus there is a CQ \( p(x) \) in \( q(x) \) such that \( \mathcal{I} \models p(a) \). Consequently, there is a homomorphism \( h \) from \( p(x) \) to \( \mathcal{I} \) with \( h(a) = a \). Let \( p'(x) \) be the contraction of \( p(x) \) obtained by identifying all variables \( y_1 \) and \( y_2 \) such that \( h(y_1) = h(y_2) \). As witnessed by \( h \), \( \mathcal{I} \models p'(a) \). Note that the \( x \)-girth of \( p'(x) \) is either infinite or it is bounded from above by \( g \) since the \( x \)-girth of \( p(x) \) is. Also note that \( h \) is an injective homomorphism from \( p'(x) \) to \( \mathcal{I} \). By definition of extended forest models, all cycles in \( \mathcal{I} \) are either cy cycles from \( \mathcal{A}' \), or they are cycles of the form \( r(y, z), s(z, y) \). This together with the fact that the girth of \( \mathcal{A}' \) exceeds \( g \) implies that every cycle in \( p'(x) \) passes through \( x \) or is of the latter kind. In fact, \( p'(x) \) is \( x \)-acyclic when \( \mathcal{T} \) contains no role inclusions since then \( \mathcal{I} \) is a forest model of \( \mathcal{A}' \). Since \( p'(x) \) is a CQ in \( q_{\text{acyc}}(x) \), we are done in that case.

Now for the case where \( \mathcal{T} \) contains role inclusions. Let \( \mathcal{J} \) be the forest model of \( \mathcal{A}' \) underlying \( \mathcal{I} \). Construct a CQ \( p''(x) \) from \( p'(x) \) as follows: for all distinct variables \( y, z \), with \( y \neq x \) and \( z \neq x \), whenever \( r_1(y, z), \ldots, r_k(y, z), s_1(z, y), \ldots, s_l(z, y) \) are all atoms of this form in \( p'(x) \), then replace them with \( r(x, y) \) if \( h(x), h(y) \) is an \( r \)-edge with \( r(x, y) \) if \( h(x), h(y) \) is in \( r' \). Note that by definition of extended forest models and due to the fact that \( g \) the girth of \( \mathcal{A}' \) is greater than 2, such an \( r \) always exists. As witnessed by \( h \), \( \mathcal{I} \models p''(a) \). Moreover, \( p''(x) \) is \( x \)-acyclic and a CQ in \( q_{\text{acyc}}(x) \), thus we are again done.

(b) It is immediate by definition of \( Q_{\text{acyc}} \) that \( Q \subseteq Q_{\text{acyc}} \). We thus have to show that \( Q_{\text{acyc}} \subseteq Q \). Assume the contrary. Then, there is a \( \Sigma \)-ABox \( \mathcal{A} \) and an \( a \in \text{ind}(\mathcal{A}) \) such that \( \mathcal{A} \models Q_{\text{acyc}}(a) \) and \( \mathcal{A} \models Q(a) \). Note that \( \mathcal{A} \) must be consistent with \( \mathcal{T} \). Since \( Q \) is non-empty, there is a \( \Sigma \)-ABox \( Q \mathcal{A} \) such that \( \mathcal{A} \models Q(b) \) for some \( b \in \text{ind}(\mathcal{A}) \). Let \( \mathcal{A}' \) be the disjoint union of \( \mathcal{A} \) and \( \mathcal{A}_Q \). We get \( \mathcal{A}' \models \mathcal{A} \models Q(a) \) from Lemma 18 and thus \( \mathcal{A}' \models Q''(a) \). Since \( Q''(x) \) is from \( \mathcal{ALCI}, \mathcal{IQ} \) and \( \mathcal{A}'_{\text{acyc}} = \mathcal{A} \), the latter and Lemma 17 implies \( \mathcal{A} \models Q'(a) \), thus \( \mathcal{A} \models Q(a) \), a contradiction.

When the OMQ language \( \mathcal{L} \) is replaced by \( \mathcal{L}^* \), we can show that \( Q \equiv Q_{\text{acyc}}^* \) exactly as above. The second part of the proof showing that \( Q \equiv Q_{\text{acyc}}^* \) (does not go through and) is no longer needed.

**Lemma 23.** Let \( Q = (T, \Sigma, q(x)) \) be an OMQ from \( \mathcal{ALCH}, \mathcal{UCQ} \) such that \( q(x) \) is \( x \)-acyclic and connected. Then \( Q \) is rewritable into an OMQ \( (T \cup T', \Sigma, q(x)) \) from \( \mathcal{ALCI}, \mathcal{IQ} \) whose size is polynomial in the size of \( Q \).

**Proof.** Let \( Q = (T, \Sigma, q(x)) \) be an OMQ from \( \mathcal{ALCH}, \mathcal{UCQ} \) such that \( q(x) \) is \( x \)-acyclic and connected.
where $A$ ranges over all concept names, $r$ over all (potentially inverse) roles, and $E$ over all $\text{ALC}^\text{all}$-concepts. Note that every $\mathcal{ELT}^\text{all}$-concept decorated with $\text{ALC}^\text{all}$-concepts is an $\text{ALC}^\text{all}$-concept, but that the converse is false.

We construct from $Q'$ an $(\text{ALCH}^\text{all}, \text{IQ})$-rewriting $(\mathcal{T}, \Sigma, C'(x))$ of $Q$ where $C'$ has the form $C_{\text{pre}} \rightarrow C_{\text{con}}$. To start, let $D_1 = \exists r_1 . P \ldots , D_k = \exists r_k . P$ be all subconcepts of $C$ that are of this form and let

- $C_{\text{con}}$ be obtained from $C$ by replacing each concept $D_i$ with a fresh concept name $F_{D_i} \notin \Sigma$ and
- $C_{\text{pre}} = \forall r_1 . P_{D_1} \land \ldots \land \forall r_k . P_{D_k}$.

Next, exhaustively apply the following transformation step:

if $D = \exists r^- . E$ is a subconcept of $C_{\text{con}}$ where $E$ is an $\text{ALC}^\text{all}$-concept (that is, does not contain any inverse roles), then

- replace $D$ in $C_{\text{con}}$ with a fresh concept name $P_D \notin \Sigma$ and
- set $C_{\text{pre}} = C_{\text{pre}} \land \forall u.(E \rightarrow \forall r.P_{D})$.

We end up with $C_{\text{con}}$ being an $\text{ALC}^\text{all}$-concept because if there is a subconcept $\exists r^- . E$ of $C_{\text{con}}$ left, then in the innermost such subconcept $E$ must be an $\text{ALC}^\text{all}$-concept and thus the transformation rule applies. It can be proved that the initial IQ $C_{\text{pre}} \rightarrow C_{\text{con}}(x)$ is equivalent to $C(x)$ and that the transformation step is equivalence preserving. We omit details, please see the proof of Lemma 3.4 for very similar arguments.

We now turn to Point 1. Let $Q = (\mathcal{T}, \Sigma, q(x))$ be an OMQ from $(\text{ALCH}^\text{all}, \text{UeCQ})$ with $q(x)$ $x$-acyclic and $x$-accessible. Then $q(x)$ is also connected. By the (extended) Lemma 3.4 there is an equivalent IQ $Q' = (\mathcal{T}, \Sigma, C(x))$ with $C(x)$ an $\text{ELT}^\text{all}$-IQ. In fact, the IQ $C(x)$ constructed in the proof of Lemma 3.4 is of the form $P \rightarrow p(x) \rightarrow C_p(x)$, where each $C_p$ is an $\mathcal{ELT}^\text{all}$-concept decorated with $\text{ALC}^\text{all}$-concepts, that is, an $\mathcal{ELT}^\text{all}$-concept decorated with $\text{ALC}^\text{all}$-concepts that does not mention the universal role. However, the syntactic structure of $C$ is even more restricted.

**Claim.** In each subconcept $\exists r^- . D$ of $C$, $D = P$ or $D$ has the form $D_0 \land \exists r^- . (D_1 \land \exists r^-. (D_2 \land \ldots \land \exists r^-. (P) \ldots ))$, $n \geq 1$.

**Proof of claim.** Let $p(x)$ be a CQ in $q(x)$. Recall that, when constructing $C(x)$ in the proof of Lemma 3.4, we first remove atoms of the form $r(x, y)$ from $p(x)$ to obtain a tree-shaped CQ $\rho'(x)$, then add back $r^-(y, u)$ and $P(u)$ for each removed $r(x, y)$ where $u$ is a fresh variable producing a CQ $\rho''(x)$, and finally view $\rho'''(x)$ as an $\mathcal{ELT}^\text{all}$-concept decorated with $\text{ALC}^\text{all}$-concepts.

Let $\exists r^- . D$ be a subconcept of $C_p$. Then there is a variable $y$ in $\rho'''(x)$ and an atom $r^-(y, z)$ such that $D$ describes the subtree of $\rho'''(x)$ rooted at $z$ and $z$ is a successor of $z$ in the tree-shaped $\rho'''(x)$, that is, $y$ is on the path from the root of $\rho'''(x)$ to $z$. First assume that $r^-(y, z)$ was one of the atoms added back in the construction of $\rho''(x)$. Then $D = P$ and we are done. Now assume that $r^-(y, z)$ was already in $\rho'(x)$ and thus in $p(x)$. Since $p(x)$ is $x$-accessible, $z$ is reachable from $x$ in the directed graph $G_p$. Since $z$ is not reachable from $x$ in the directed graph $G_p$, it follows from the construction of $\rho''(x)$ and $\rho'''(x)$ that $z$ is reachable from $G_p$, from a leaf node labeled with $P$. Consequently, $D$ must have the stated form. This finishes the proof of the claim.

We construct from $Q'$ an $(\text{ALCH}^\text{all}, \text{IQ})$-rewriting $(\mathcal{T}, \Sigma, C'(x))$ of $Q$ where $C'$ has the form $C_{\text{pre}} \rightarrow C_{\text{con}}$. To start, let $D_1 = \exists r_1 . P \ldots , D_k = \exists r_k . P$ be all subconcepts of $C$ that are of this form and let

- $C_{\text{con}}$ be obtained from $C$ by replacing each concept $D_i$ with a fresh concept name $F_{D_i}$ and
- $C_{\text{pre}} = \forall r_1 . P_{D_1} \land \ldots \land \forall r_k . P_{D_k}$.

It is easy to see that the following condition is satisfied:

(*) in every subconcept $D = \exists r^- . E$ of $C_{\text{con}}$ with $E$ an $\text{ALC}$-concept, $E$ is of the form $F \land P_D$.

Next, exhaustively apply the following transformation step, which preserves (*): if $D = \exists r^- . (F \land P_D)$ is a subconcept of $C_{\text{con}}$ where $F$ is an $\text{ALC}$-concept, then

- replace $D$ in $C_{\text{con}}$ with a fresh concept name $P_D$ and
- replace $P_D$ in $C_{\text{pre}}$ with $F \rightarrow \forall R.P_D$.

It can be verified that, because of the claim, the transformation step indeed preserves (*). It can also be seen that all subconcepts of the form $\exists r^- . E$ will eventually be eliminated. Finally, it can be shown that the initial IQ $C_{\text{pre}} \rightarrow C_{\text{con}}(x)$ is equivalent to $C(x)$ and that the transformation step is equivalence preserving. We omit details.

**Example 25.** Let $Q$ be the OMQ from Example 22. Towards obtaining an $\text{ALC-IQ}$ rewriting, we start with the $\text{ELT-IQ}$ rewriting $Q'$ described in the same example. The only subconcept of the form $\exists r^- . P$ in $C_q$ is $D = \exists s^- . P$. We thus introduce a fresh concept name $P_D$ and initialize $C_{\text{pre}}$ and $C_{\text{con}}$ with $\forall s.P_D$ and $\exists r. \exists t. (\exists s \land \exists t \land P_D)$.

We now consider concepts of the form $\exists r^- . (F \land P_D)$, with $F$ an $\text{ALC}$ and $P_D$ previously introduced. The only such concept is $E = \exists t. (\exists s \land \exists t \land P_D)$. We replace $E$ in $C_{\text{pre}}$ with $P_E$ and $P_D$ in $C_{\text{pre}}$ with $\exists t. \exists s(P_E \land P_D)$. At this point both $C_{\text{pre}} = \forall t. (\exists s \land \exists t \land P_D)$ and $C_{\text{con}} = \exists r.P_E$ are $\text{ALC}$ concepts, thus no further transformation is possible (and neither needed): $Q$ can be rewritten into an OMQ $Q'' = (\mathcal{T}, \Sigma, C'(x))$ with $C'$ the $\text{ALC}$ concept $\forall s. (\forall t. (\exists s \land \exists t \land P_E) \rightarrow \exists r.P_E)$.

**Theorem 5.** Let $\mathcal{L} \in \{\text{ALC}, \text{ALCH}\}$ and let $Q = (\mathcal{T}, \Sigma, q(x))$ be a unary OMQ from $(\mathcal{L}, \text{UeCQ})$ that is non-empty. Then the following are equivalent:

1. $Q$ is rewritable into an OMQ from $(\mathcal{L}, \text{IQ})$.
2. $Q$ is rewritable into an OMQ $(\mathcal{T} \cup \mathcal{T}', \Sigma, C(x))$ from $(\mathcal{L}, \text{IQ})$.
3. $Q$ is rewritable into an OMQ from $(\mathcal{L}, \text{IQ})$.
4. $Q$ is rewritable into an OMQ $Q' = (\mathcal{T}, \Sigma_{\text{full}}, C(x))$ from $(\mathcal{L}, \text{IQ})$.
5. $Q \equiv (\mathcal{T}, \Sigma_{\text{full}}, q_{\text{deco}}(x))$. 

---

4The first two steps can together be viewed as an unfolding construction.
If, furthermore, $\mathcal{L}$ is replaced with $\mathcal{L}^u$ and $\mathcal{L}^u I$ with $\mathcal{L}^u I$, then Conditions 1 to 3 are further equivalent to:

6. $Q$ is rewritable into an OMQ $Q' = (T, \Sigma, C(x))$ from $(\mathcal{L}^u, IQ)$.

**Proof.** "$2 \Rightarrow 1$" and "$1 \Rightarrow 3$" are trivial.

"3 $\Rightarrow$ 2". We know from Theorem 3 that $(\mathcal{L}^u, IQ)$-rewritability of $Q$ implies that $q(x)$ is $\times$-acyclic and connected. By Lemma 23, $Q$ is rewritable into an OMQ from $(\mathcal{L}, IQ)$ that is of the desired shape.

"4 $\Rightarrow$ 5". Let $Q = (T, \Sigma_{full}, q(x))$ be an OMQ from $(\mathcal{ALCI}^I, UCQ)$ and assume that $Q$ is rewritable into an OMQ $Q' = (T, \Sigma_{full}, C(x))$ with $C(x)$ an $\mathcal{ALCI}$-IQ. Let $Q_{acyc} = (T, \Sigma_{full}, \mathcal{q}_{acyc}(x))$. It is established in the proof of the "1 $\Rightarrow$ 3" direction of Theorem 3 that, since $Q$ is rewritable into $(\mathcal{ALCI}^I, IQ)$, $Q \equiv Q_{acyc}$. It thus suffices to show that $Q_{acyc} \equiv Q_{deco}$.

Using the definition of $Q_{deco}$, it can be shown that $Q_{acyc} \subseteq Q_{deco}$. To establish the converse direction, assume towards a contraction that there is a $\Sigma_{full}$-ABox $A$ such that $A \models Q_{deco}(a)$ but $A \not\models Q_{acyc}(a)$. Then $A \not\models Q'(a)$. Take a model $I$ of $A$ and $T$ such that $I \not\models C(a)$. We have $I \models Q_{acyc}(a)$, thus $I \models p(x)\dreach(p)$ for some $T$-decoration $p(x)$ of a CQ in $q_{acyc}(x)$. Let $h$ be a homomorphism from $p(x)\dreach(p)$ to $I$ with $h(x) = a$.

To finish the proof, it suffices to show that we can construct from $I$ a model $I'$ of $T$ such that $I' \not\models C(a)$ and $I' \models q_{acyc}(a)$. In fact, we can then take a homomorphism $h'$ in $q_{acyc}(x)$ to $I'$ with $h'(x) = a$ and let $A'$ be $I'$ restricted to the range of $h'$, viewed as an ABox. Clearly, $A' \models Q_{acyc}(a)$ since already $A' \models (\emptyset, \Sigma_{full}, q_{acyc}(x))(a)$. Moreover, $I'$ is a model of $A'$ and thus $A' \not\models Q'(a)$, in contradiction to $Q'$ being equivalent to $Q_{acyc}$.

It thus remains to construct $I'$. Informally, we do this by adding to $I$ the part of $p(x)$ that is not reachable from the answer variable along a directed path. By the second condition of $T$-decorations, there is a model $J$ of $T$ and a homomorphism $h'$ from $p(x)$ to $J$. We can assume that $I$ and $J$ have disjoint domains.

Let $I'$ be the disjoint union of $I$ and $J$, extended as follows: for every atom $r(y_1, y_2)$ in $p(x)$ with $y_1 \not\in \dreach(p)$ and $y_2 \in \dreach(p)$, add $(h'(y_1), h(y_2))$ to $I'$. It can be verified that the map $h''$ defined by setting $h''(y) = h(y)$ for all $y \in \dreach(p)$ and $h''(y) = h'(y)$ for all variables $y$ in $p$ that are not in $\dreach(p)$ is a homomorphism from $p(x)$ to $I'$ with $h''(x) = a$. Thus, $I' \models q_{acyc}(a)$ as desired. It thus remains to show that $I'$ is a model of $T$ and that $I' \not\models C(a)$. This is a consequence of the following:

(a) for all $\mathcal{ALCI}$-concepts $C$ and all $d \in \Delta_T, d \in C \iff d \in C' \cap \Delta_T$;

(b) for all subconcepts $C$ of a concept in $T$ and all $d \in \Delta_J, d \in C \iff d \in C' \cap \Delta_J$.

Both points are proved by induction on the structure of $C$. This is straightforward for (a) since for every element $d \in \Delta_T$, the subinterpretation of $I$ induced by the set of elements reachable from $d$ in $I$ by traveling roles in the forwards direction is identical to the corresponding subinterpretation of $I'$ (and since $\mathcal{ALCI}$-concepts do not admit inverse roles). For (b), it is important to observe that if we have added $(h'(y_1), h(y_2))$ to $I'$ in the construction of $I'$, then $h'(y_1)$ has an $r$-successor $d$ in $I'$ such that for all subconcepts $C$ of a concept in $T$, $d \in C' \iff h(y_2) \in C'$. In fact, this is a consequence of the decoration of every variable in $p(x)$ with such concepts: when choosing $d = h'(y_2)$, the stated condition must be satisfied.

"5 $\Rightarrow$ 4". We have that $Q \equiv (T, \Sigma, q(x))$, where each $q_i$ is an $\times$-acyclic, accessible eCQ. We can apply Lemma 24 to obtain an $(\mathcal{L}, IQ)$-rewriting.

For the case with the universal role, it is enough to show that 3 $\Rightarrow$ 6. Again, we can apply Lemma 24.

**Theorem 7.** Let $q(x)$ be a UCQ. Then

1. $q(x)$ is rewritable into an $\mathcal{ALCI}$-IQ iff there is a subquery $q'(x)$ of $q(x)$ that is $\times$-acyclic, connected, and equivalent to $q(x)$;

2. $q(x)$ is rewritable into an $\mathcal{ALCI}$-IQ if there is a subquery $q'(x)$ of $q(x)$ that is $\times$-acyclic, $\times$-accessible, and equivalent to $q(x)$.

When $\mathcal{L}$-IQs are replaced with $\mathcal{L}^u$-IQs, then the same equivalences hold except that connectedness/$\times$-accessibility is dropped.

**Proof.** For the statement at Point 1, the “if” direction is a consequence of Lemma 3 while for the statement at Point 2, the same direction is a consequence of Lemma 24 (and similarly for the cases where the universal role is present). We will thus show the “only if” direction in each case. We first show that $\mathcal{L}$-IQ rewritability of a UCQ $q(x)$, for every $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCI}^I, \mathcal{ALCI}^u, \mathcal{ALCI}^u\}$, implies the existence of a subquery $q'(x)$ of $q(x)$ that is $\times$-acyclic and equivalent to $q(x)$.

A homomorphism minimal CQ (also hom-minimal) is a CQ which does not admit any equivalent strict subquery.

**Claim 1.** Let $q$ and $q'$ be two CQs such that $q \equiv q'$ and $q'$ is hom-minimal. Then:

1. $q'$ is a subquery of $q$;

2. $q'$ is a contraction of $q$.

**Proof of claim.** Consider any homomorphisms $h_1$ and $h_2$ from $q'$ to $q$ and from $q$ to $q'$, respectively. Then $h_1$ must be injective and $h_2$ must be surjective (otherwise $h_1 \circ h_2$ is a non-injective homomorphism from $q'$ to itself, and thus $q'$ is not hom-minimal). The existence of $h_1$ implies that $q'$ is a subquery of $q$, while the existence of $h_2$ implies that $q'$ is a contraction of $q$.

**Claim 2.** Let $p(x)$ be a CQ in $q_{acyc}(x)$. Then, there exists a hom-minimal CQ $p'(x)$ in $q_{acyc}(x)$ such that $p(x) \equiv p'(x)$.

**Proof of claim.** We show that, in fact, every hom-minimal subquery $p'(x)$ of $p(x)$ which is equivalent to $p(x)$ is a CQ in $q_{acyc}(x)$. From Claim 1, $p'(x) \equiv p(x)$ and $p'(x)$ being hom-minimal, implies that $p'(x)$ is a contraction of $p(x)$. As $p(x)$ is a contraction of a CQ in $q(x)$, it follows that $p'(x)$ is itself a contraction of some CQ in $q(x)$. As $p(x)$ is $\times$-acyclic and
p'(x) is a subquery of p(x), it follows that p'(x) is x-acyclic. Thus, p'(x) is an x-acyclic contraction of some CQ in q(x), or, in other words, p'(x) is a CQ in q_{acyc}(x).

Assume now L-IQ rewritability of q(x). By inspecting Point (a) in the proof of direction “1 \Rightarrow 3” of Theorem 3 we observe that q(x) \equiv q_{acyc}(x).

For didactic purposes, we first consider the case where q(x) is a CQ. Then, there must be a CQ p(x) in the UCQ q_{acyc}(x) such that q(x) \equiv p(x). From Claim 2, there must be some CQ p'(x) in q_{acyc}(x) which is hom-minimal and equivalent to p(x), and thus also to q(x). From Claim 1, it follows that p'(x) is a subquery of q(x), and from the fact that p'(x) is a CQ in q_{acyc}(x), it follows that p'(x) is x-acyclic.

We now consider the case where q(x) is a UCQ. Let q_1(x), \ldots, q_k(x) be the CQs in q(x) that are minimal in q(x) in the following sense: for all CQs p(x) in q(x), q_i(x) \subseteq p(x) implies q_i(x) \equiv p(x). Take such a minimal CQ q_i(x).

Since q(x) \equiv q_{acyc}(x), there must be a CQ p_i(x) in q_{acyc}(x) such that q_i(x) \subseteq p_i(x). By construction of q_{acyc}(x), p_i(x) must be the contraction of some CQ q_i(x) in q(x) and thus p_i(x) \subseteq q_i(x). We obtain q_i(x) \subseteq q_i(x) and thus q_i(x) \equiv q_i(x) and consequently q_i(x) \equiv p_i(x). From Claim 2, there must be some hom-minimal query p_i(x) \in q_{acyc}(x) such that p_i(x) \equiv p_i(x). Then, q_i(x) \equiv p_i(x) and from Claim 1, p_i(x) is a sub-query of q_i(x). Let q_{acyc}(x) be the restriction of q_{acyc}(x) to the chosen CQs p_1(x), \ldots, p_k(x). Clearly, q_{acyc}(x) is equivalent to q(x).

Now we concentrate on the “only if” direction for Point 1, i.e. the case where L is \(\mathcal{ACCT}\), and thus q(x) is \(\mathcal{ACCT}\)-IQ rewritable. We already know that q_{acyc}(x) is equivalent to q(x), thus q_{acyc}(x) is also \(\mathcal{ACCT}\)-IQ rewritable. From the proof of direction “1 \Rightarrow 3” Point (b) in Theorem 3, \(\mathcal{ACCT}\)-IQ rewritability implies q(x) \equiv q^{con}(x), and thus also q^{con}(x) \equiv q_{acyc}(x) and q(x) \equiv q_{acyc}(x). It is easy to see that (q_{acyc})^{con}(x) is an x-acyclic connected subquery of q(x). Point 2 and the cases with universal roles are treated similarly.

Theorem 8. For every Q \in \{CQ, UCQ\} and L \in \{\mathcal{ALC}, \mathcal{ACCT}, \mathcal{ACCT}^u, \mathcal{ACCT}^u\}, it is NP-complete to decide whether a given query from Q is L-IQ-rewritable.

Proof. Let L \in \{\mathcal{ACCT}, \mathcal{ACCT}^u, \mathcal{ACCT}^u\}. We start with the upper bound, that is, given a UCQ q(x), it is in NP to decide whether q(x) is L-IQ-rewritable.

We guess a subquery q'(x) of the original query q(x) and check whether q'(x) \equiv q(x). This is the case when for every CQ p(x) in q(x) there exists a CQ q'(x) in q'(x) such that p(x) \subseteq p'(x) and vice versa. We guess for every CQ p(x) in q(x) a target CQ q'(x) in q(x) and a potential homomorphism h_p : ind(\mathcal{A}_p) \rightarrow ind(\mathcal{A}_p). We also guess for every CQ p'(x) in q'(x) a target CQ p(x) in q(x) and a potential homomorphism h_{p'} : ind(\mathcal{A}_{p'}) \rightarrow ind(\mathcal{A}_p). We then check that every h_p and every h_{p'} is an actual homomorphism.

To show NP-hardness of whether a given CQ is L-IQ-rewritable, we employ a reduction from the 3-colorability problem (3COL). A CQ G = (V,E) be an undirected graph, let q_G be G viewed as a conjunctive query where every \{v_1,v_2\} \in E is represented by two atoms \(r(v_1,v_2), r(v_2,v_1)\), and choose a v \in V. Let

\[
q(x_0) = \exists y q_G \land r(x_0,v) \land r(v,x_0) \land \
\land \{r(x_i,x_j) \mid i, j \leq 2 \text{ with } i \neq j\}
\]

where y contains all elements of V (as variables) as well as the fresh variables x_1 and x_2.

Claim. q(x_0) is L-IQ-rewritable iff G is 3-colorable.

Proof of claim. For the “if” direction, assume that G is 3-colorable. Then G admits a homomorphism into the 3-clique (without reflexive loops). Consequently, q_G(x) is homomorphically equivalent to the restriction q_{3C}(x_0) of q(x) to the variables x_0, x_1, x_2. In particular, q_G(x_0) and q_{3C}(x_0) are then equivalent in the sense of query containment. Since q_{3C}(x_0) is x_0-acyclic and x_0-accessible, by Lemma 24 it is rewritable into an ALC-IQ.

Conversely, assume that G is not 3-colorable. By Lemma 24 it suffices to show that any subquery p(x_0) of q(x_0) that is equivalent to q(x_0) is not x_0-acyclic. Thus let p(x_0) be such a subquery. There is no homomorphism h from p(x_0) to q_{3C}(x_0) since the equivalence of p(x_0) and q(x_0) implies the existence of a homomorphism h' from q(x_0) to p(x_0) and composing h' with h would establish 3-colorability of G. It is easy to verify, though, that when p(x_0) contains no cycle that does not pass through any of x_0, x_1, x_2, there is such a homomorphism h. Consequently, p(x_0) is not x_0-acyclic.

Theorem 9. Let Q \in \{CQ, UCQ\}. For OMQs based on the full ABox signature, IQ-rewritability is
1. ExpTime-hard in \((ALC, Q)\) and in CoNExpTime in \((ALCHI, Q)\) and
2. 2ExpTime-complete in \((ALCIT, Q)\) and \((ALCHIT, Q)\).

**Proof.** We start with the lower bounds. Points 1 and 2 are treated uniformly. In fact, for \(L \in \{ALC, ALCIT\}\), we reduce a special case of OMQ evaluation in \((L, Q)\) to IQ-rewritability in \((L, CQ)\) where \(OMQ\) evaluation in \((L, Q)\) means to decide, given an OMQ \(Q = (T, \Sigma_{\text{full}}, q(x))\) from \((L, Q)\), an ABox \(A\), and a tuple \(a\) whether \(A \models Q(a)\). The mentioned special case is that OMQs are Boolean and \(T\) takes the form \(\{a(a)\}\) and we refer to this as singleton BOMQ evaluation.

Singleton BOMQ evaluation is ExpTime-hard in \(ALC\) \cite{Lutz:2008}. We observe that it is ExpTime-hard in \(ALC\) since concept (un)satisfiability w.r.t. \(ALC\)-TBoxes is ExpTime-hard \cite{Schild:1991} and an \(ALC\)-concept \(C\) is unsatisfiable w.r.t. an \(ALC\)-TBox \(T\) iff \(\{a(a)\} \models (T \cup \{A \sqsubseteq C\}, \Sigma_{\text{full}}, \exists y D(y))\) where \(A\) and \(D\) are fresh concept names.

Now for the reduction to IQ-rewritability. Let \(Q = (T, \Sigma_{\text{full}}, q(\bar{x}))\) be an OMQ from \((L, CQ)\). Let \(L \in \{ALC, ALCIT\}\), and let \(A = \{a(x)\}\) be an ABox. Further, \(Q(x)\) is the extension of \(q(\bar{x})\) with the atom \(A(x)\), \(x\) a fresh answer variable. It is important to note that \(Q(x)\) is a disconnected CQ.

**Claim.** \(A \models Q\) iff \(Q' = (T, \Sigma_{\text{full}}, q(x))\) is IQ-rewritable.

**Proof of claim.** If \(A \models Q\), then \(Q'\) is equivalent to \((T, \Sigma_{\text{full}}, A(x))\) which is from \((L, IQ)\). Conversely, assume that \(A \not\models Q\). The query \(Q_{\text{con}}\) from Point (b) in the proof of the “\(1 \Rightarrow 3\) direction of Theorem 3 applied to \(Q\), is exactly \((T, \Sigma_{\text{full}}, A(x))\). As shown there, IQ-rewritability of \(Q\) implies \(Q \models Q_{\text{con}}\), in contradiction to \(A \not\models Q\). This finishes the proof of the claim.

For the upper bounds, we use the characterizations from Theorem 3 and Theorem 5 deciding IQ-rewritability in \((ALCHI, UCQ)\) amounts to checking containment between \(Q\) and \((T, \Sigma_{\text{full}}, q_{\text{deco}}(x))\) while deciding IQ-rewritability in \((ALCHI, UCQ)\) amounts to checking containment between \(Q\) and \((T, \Sigma_{\text{full}}, q_{\text{deco}}(x))\). Note that the two involved OMQs share the same TBox and are based on the full ABox signature. There is also an initial emptiness check, which however is just another containment check. We thus have to argue that these containment checks can be carried out in ExpTime and NExpTime, respectively.

We start with the case of \((ALCHI, UCQ)\) and first observe that containment in \((ALCHI, UCQ)\) is in ExpTime. In fact, it is shown in \cite{Lutz:2008} that OMQ evaluation in \((ALCHI, CQ)\) is in 2ExpTime and the algorithm given there is straightforwardly extended to \((ALCHI, UCQ)\). It follows that containment between an OMQ \(Q_1 = (T, \Sigma_{\text{full}}, q_1(x_1))\) from \((ALCHI, CQ)\) in an OMQ \(Q_2 = (T, \Sigma_{\text{full}}, q_2(x_2))\) from \((ALCHI, UCQ)\) is in 2ExpTime since \(Q_1 \subseteq Q_2\) iff \(A_{Q_1} \models Q_2(x_2)\).

We next observe how this can be lifted to containment in \((ALCHI, UCQ)\). In fact, it suffices to show that for \(Q_1 = (T, \Sigma_{\text{full}}, q_1)\) from \((ALCHI, UCQ)\), \(i \in \{1, 2\}\), and \(q_1 = p_1 \lor \cdots \lor p_k\), we have \(Q_1 \subseteq Q_2\) iff \(T, \Sigma_{\text{full}}, p_i \subseteq Q_2\) for all \(i \in \{1, \ldots, k\}\). The “if” direction is trivial. For the “only if” direction, we argue as follows. Assume that \((T, \Sigma_{\text{full}}, p_i) \subseteq Q_2\) for all \(i \in \{1, \ldots, k\}\). Let \(A\) be an ABox and \(a \in \text{ind}(A)\) such that \(A \models Q_1(a)\). It suffices to show that \(I \models Q_2(a)\) for every finite model \(I\) of \(A\) and \(T\). Let \(I\) be such a model and let \(A_I\) be \(I\) viewed as an ABox. Since \(A \models Q_1(a)\), we must have \(I \models p_i(a)\) for some \(j \in \{1, \ldots, k\}\). Then clearly also \(A_I \models p_j\). Since \((T, \Sigma_{\text{full}}, p_j) \subseteq Q_2\), this yields \(A_I \models Q_2(a)\). Since \(I\) is a model of \(A_I\) and \(T\), from this we obtain \(I \models Q_2(a)\), as required.

The argument is not yet complete since the UCQs \(q_{\text{deco}}(x)\) can be exponentially large. In fact, it may contain exponentially many CQs, but each CQ is only of polynomial size. For checking \(Q \subseteq (T, \Sigma_{\text{full}}, q_{\text{deco}}(x))\), using the above argument we can use exponentially many containment checks between an OMQ from \((ALCHI, CQ)\) and an OMQ from \((ALCHI, UCQ)\), both of polynomial size. The overall complexity is thus 2ExpTime, as required. For checking \(Q \subseteq (T, \Sigma_{\text{full}}, q_{\text{con}}(x))\), we observe that, by Lemma 4 \((T, \Sigma_{\text{full}}, q_{\text{con}}(x))\) is rewritable into an equivalent OMQ \((T, \Sigma_{\text{full}}, C(x))\) with \(C(x)\) an \(ALCHI\)-IQ (independently of the properties of \(Q\)) and such that the size of \(C(x)\) is polynomial in the size of \(q_{\text{con}}(x)\), which in turn is single exponential in the size of \(Q\). We can thus replace the check \(Q \subseteq (T, \Sigma_{\text{full}}, q_{\text{deco}}(x))\) with \(Q \subseteq (T, \Sigma_{\text{full}}, C(x))\). This boils down to deciding OMQ entailment in \((ALCHI, IQ)\), which is in ExpTime. So despite \(C(x)\) being of (single) exponential size, we achieve 2ExpTime overall complexity.

For the case of \((ALCHI, UCQ)\), the argument is essentially the same. However, as also shown in \cite{Lutz:2008} OMQ evaluation in \((ALCHI, UCQ)\) is in ExpTime and thus so is our basic containment check between an OMQ from \((ALCHI, CQ)\) and an OMQ from \((ALCHI, UCQ)\). Therefore, the check \((T, \Sigma_{\text{full}}, q_{\text{deco}}(x)) \subseteq Q\) can be implemented in ExpTime despite the exponential number of CQs in \(q_{\text{deco}}(x)\). It is not clear, however, how to implement the containment check \(Q \subseteq (T, \Sigma_{\text{full}}, q_{\text{deco}}(x))\) in ExpTime. We give a sketch of how it can be implemented in CoNExpTime. In fact, what we have to implement in CoNExpTime is the evaluation of an OMQ \(Q = (T, \Sigma_{\text{full}}, q(x))\) where \(q(x)\) is a UCQ with exponentially many connected CQs, each of polynomial size. Let \(A\) be an ABox and \(a \in \text{ind}(A)\). By Lemma 6 \(A \models Q(a)\) iff there is an extended forest model \(I\) of \(A\) and \(T\) such that \(I \not\models q(a)\). It is easy to see that we can further demand that (the tree-shaped parts of) \(I\) be of outdegree polynomial in the size of \(T\). Our NExpTime algorithm for the complement of OMQ evaluation is as follows. Let \(m\) be the maximum number of variables of a CQ in \(q(x)\). We guess an initial piece of the extended forest model \(I\) that consists of the ‘ABox part’ of \(A\) together with the tree-shaped parts restricted to depth \(m + 1\), along with a type adornment, that is, a function \(\mu\) that assigns a \(T\)-type to every element of the guessed initial part in a way that is consistent with the initial part. Note that we guess an object of single exponential size here. Since the CQ in \(q(x)\) is connected and thus are independent of the part of \(I\) that lies beyond the guessed initial part, we can verify that \(I\) can be extended to a full model by considering the type \(\mu(d)\) for every leaf \(d\) in the initial part on level \(m + 1\) and verifying that \(\mu(d)\) is satisfiable with \(T\).
This can be implemented in \textsc{ExpTime}. \hfill \Box

We introduce a preparatory lemma and notation for the proof of Theorem 5. An atomic query (AQ) is an IQ of the form $A(x)$, with $A$ a concept name. A Boolean atomic query (BAQ) is a query of the form $\exists x A(x)$, with $A$ a concept name, and a Boolean conjunctive query (BCQ) is a CQ of arity zero.

\textbf{Lemma 26.} Let $Q = (T, \Sigma, \exists x C(x))$ be an OMQ from $(\text{ALCII}, \text{BAQ})$. Then, there exists an OMQ $Q' = (T', \Sigma, M(x))$ from $(\text{ALCII}, \text{AQ})$, such that for all $\Sigma$-ABoxes $A$, $A \models Q$ iff there is an $a \in \text{ind}(A)$ such that $A \models Q'(a)$.

\textbf{Proof.} We first observe that $A \models Q$ iff $A' \models Q$, for some $A' \subseteq \text{CON}_A$. Thus, it is enough to show the statement of the lemma for connected $\Sigma$-ABoxes.

Let $A'$ be a connected $\Sigma$-ABox and $M$ a fresh concept name. Let $T'$ be the TBox obtained from $T$ by adding $C \sqsubseteq M$ and $\exists r.M \sqsubseteq M$ for every role $r$ such that $r$ or the inverse of $r$ occurs in $T$. It can be verified that $Q' = (T', \Sigma, M(x))$ is as required. \hfill \Box

We also introduce a more fine-grained version of a complexity result from [Bourhis and Lutz, 2016] which highlights that the complexity of containment is double exponential only in the maximum size of CQs in the input OMQs, but not in their number. This only requires a careful analysis of the constructions in [Bourhis and Lutz, 2016].

\textbf{Theorem 27.} Containment between OMQs from $(\text{ALCII}, \text{UCQ})$ is in 2\textsc{NExpTime}. More precisely, for OMQs $Q_1 = (T_1, \Sigma, q_1)$ and $Q_2 = (T_2, \Sigma, q_2)$ with arity $a$ and where $n_1$ is the number of CQs in $q_1$ and $n_2$ the maximum size of a CQ in $q_2$, $i \in \{1, 2\}$, it can be decided in time $2^{2^n \log n_1 + 1 + \log n_2 + 2 + \log |T_{i+1}| + \log \log a}$, $p$ a polynomial.

\textbf{Theorem 10.} IQ-rewritability is

1. \textsc{NExpTime-hard in $(\text{ALC}, \text{CQ})$ and $(\text{ALCII}, \text{CQ})$, $(\text{ALCII}, \text{UCQ})$, $(\text{ALCHI}, \text{CQ})$, $(\text{ALCHI}, \text{UCQ})$.}

\textbf{Proof.} We start with the lower bound for Point 1, using a reduction from OMQ emptiness in $(\text{ALCII}, \text{AQ})$ which is known to be \textsc{NExpTime}-hard. Let $Q_0 = (T, \Sigma, A(x))$ be an OMQ from this language. Also let

$$q(x) = \exists y A(x) \land r(x, y) \land r(y, y),$$

where $r$ is a role name that does not occur in $T$ and let $Q = (T, \Sigma, \{r\}, q(x))$. It suffices to show that $Q_0$ is empty iff $Q$ is IQ-rewritable.

In fact, emptiness of $Q_0$ implies emptiness of $Q$ and thus $(\text{ALC}, \text{AQ})$-rewritability. Conversely, assume that $Q_0$ is non-empty. To show that $Q$ is not IQ-rewritable, by Theorem[8] it suffices to show that $Q \neq Q_{\text{deco-cy}} := (T, \Sigma \cup \{r\}, q_{\text{deco-cy}}(x))$ where $q_{\text{deco-cy}}(x)$ is a UCQ in which every CQ contains the sub-query $r(x, x)$. Since $Q_0$ is non-empty, there is an $\Sigma$-ABox $A$ and an $a \in \text{ind}(A)$ such that $A \models Q_0(a)$. Since $r$ does not occur in $T$, we can assume w.l.o.g. that it does not occur in $A$ as well. Let $A' = A \cup \{r(a, b), r(b, b)\}$. By definition of $Q(a)$, clearly $A' \models Q(a)$. Moreover, $A' \neq Q_{\text{deco-cy}}(a)$ because $A'$ does not contain the assertion $r(a, a)$.

To establish the lower bound for Point 2, we use a reduction from containment between an OMQ from $(\text{ALCII}, \text{BAQ})$ and an OMQ from $(\text{ALCII}, \text{BCQ})$. This problem has been shown to be 2\textsc{NExpTime}-hard in [Bourhis and Lutz, 2016]. The reduction presented there uses different TBoxes in the two involved OMQs. However, by Theorem 3 in [Bienvenu et al., 2012], we can assume w.l.o.g. that they both share the same TBox.

Now for the reduction to IQ-rewritability. Let $Q_1 = (T, \Sigma, \exists x A(x))$ be an OMQ from $(\text{ALCII}, \text{BAQ})$ and $Q_2 = (T, \Sigma, q())$ an OMQ from $(\text{ALCII}, \text{BCQ})$. We first show that $Q_1$ and $Q_2$, which are Boolean, can be replaced with unary OMQs. By Lemma 26 we find an OMQ $Q'_1 = (T', \Sigma, M(x))$ from $(\text{ALCII}, \text{AQ})$ such that for all $\Sigma$-ABoxes $A, A \models Q_1$ iff $A \models Q'_1(a)$ for some $a \in \text{ind}(A)$. Clearly, the construction from the proof of Lemma 26 is such that $Q_1$ is equivalent to $(T', \Sigma, \exists x A(x))$ and by choosing the fresh concept $M$ to also not occur in $q()$ we can further ensure that $Q_2$ is equivalent to $(T', \Sigma, q())$. Thus, we can assume that $Q_1$, $Q_2$, and $Q'_1$ all use the same TBox $T$ and we can further assume $T$ contains a CI $T \subseteq N$ where $N$ is a concept name not occurring anywhere else, including $\Sigma$. Set $Q_2 = (T, \Sigma, q')$ where $q'(x)$ is $q()$ extended with the atom $N(x)$, a fresh (answer) variable. It can be verified that $Q_1 \subseteq Q_2$ iff $Q'_1 \subseteq Q_2$.

Now let $q_0(x)$ be $q()$ extended with the atom $M(x)$, a fresh (answer) variable. It is important to note that $q_0(x)$ is a disconnected CQ.

\textbf{Claim.} $Q_1 \subseteq Q_2$ iff $Q = (T, \Sigma, q_0(x))$ is IQ-rewritable.

\textbf{Proof of claim.} If $Q_1 \subseteq Q_2$, then $Q$ is equivalent to $Q'_1$, which is from $(\text{ALCII}, \text{IQ})$. Conversely, assume that $Q_1 \not\subseteq Q_2$. The query $Q^{\text{con}}$ from Point (b) in the proof of the “$1 \Rightarrow 3$” direction of Theorem 3 applied to $Q$, is exactly $Q'_1$. As shown there, IQ-rewritability of $Q$ implies $Q \equiv Q^{\text{con}}$, in contradiction to $Q_1 \not\subseteq Q_2$.

The upper bounds are a consequence of the characterization of IQ-rewritability in $(\text{ALCHI}, \text{UCQ})$ from Theorem 5 in terms of query containment. Containment in $(\text{ALCHI}, \text{UCQ})$ is in 2\textsc{NExpTime} [Bourhis and Lutz, 2016]. Note that our characterizations use UCQs with exponentially many CQs, each of which is of polynomial size so we cannot apply the containment complexity result as a black box. However, by perusing the refined OMQ containment complexity result from Theorem 27 we observe that containment checking for OMQs is double exponential only in the size of the CQs in the UCQs while it is only exponential in the number of CQs.

Note that we also need an emptiness check beforehand: if the check succeeds and the OMQ is empty, it is also rewritable and so we answer ‘yes’, if not we proceed to perform the containment check. Emptiness is simply a special case of containment, so we end up in the right complexity class. \hfill \Box

\section{D Proofs for Section 5}

We postpone the proof of Theorem 11 as we need the ultrafilter technique introduced in the proof of Theorem 13. We
Lemma 29. Let \( \mathcal{T} = \{ \text{func}(s_1), \text{func}(s_2) \} \) and
\[
p(x) = \exists y, z(r(x, y) \land s_1(y, z) \land s_2(y, z))
\]
and obtain \( p'(x) \) from \( p(x) \) by adding the atom \( s_1(z, y) \).
Then \((T, \Sigma_{\text{full}}, p(x))\) is not rewritable into an OMQ in
\((T, \Sigma_{\text{full}}, q(x))\) with \( q(x) \) a CI but \((T, \emptyset, \Sigma_{\text{full}}, p'(x))\)
is rewritable into the OMQ \((T, \Sigma_{\text{full}}, q(x))\), where
\[
q(x) = \forall r.\left( \bigcup_{1 \leq i \leq 3} P_i \right) \cap \forall r.\exists s_1, \quad \left( 1 \leq i \leq 3 \right), Q_i
\to \left( \exists r.\exists_{1 \leq i \leq 3} P_i \to \exists s_1, (C \cap \exists s_1, P_i) \right)
\]
where
\[
C = \bigcap_{1 \leq i \leq 3} (Q_1 \to \exists s_1, Q_2, Q_3).
\]

We split the proof of Theorem 13 into two parts and state the claims in such a way that they cover the extensions discussed in the main text. Call a role \( r \) functional w.r.t. \( T \) if \( \text{func}(r) \in \mathcal{T} \).
The functional closure \( FC_p(x_0) \) of a variable \( x_0 \) in a CQ \( p(x) \) is the set of all variables \( x_n \) in \( p(x) \) such that there is a functional path from \( x_0 \) to \( x_n \) in \( p(x) \). Let \( \text{var}(p(x)) \) denote the set of all variables in \( p(x) \) and set
\[
\text{nFC}_p(x) = \text{var}(p(x)) \setminus \text{FC}_p(x).
\]

Lemma 29. Let \( Q = (T, \Sigma_{\text{full}}, q(x)) \) be an OMQ from (\( \text{ACCLF}, \text{UCQ} \)).
Then \( Q \) is rewritable into an OMQ \((T, \Sigma_{\text{full}}, q'(x))\) with \( q'(x) \) an ACCLF-IQ if there is a subquery \( q'(x) \) of \( q(x) \) that is \( f \)-acyclic, connected, and equivalent to \( q(x) \).

When ACCLF is replaced with ACCLF\(^n\), then the implication holds without the connectedness assumption.

Proof. Fix an \( f \)-acyclic and connected UCQ \( q(x) \) and let \( \mathcal{T} \) be a ACCLF TBox. Let \( Q = (T, \Sigma_{\text{full}}, q(x)) \).
We construct an ACCLF-IQ \( C(x) \) such that \( Q \equiv Q' \) for \( Q' = (T, \Sigma_{\text{full}}, C(x)) \).
Let \( p(x) \) be a CQ in \( q(x) \). A cluster in \( p(x) \) is any maximal subset \( X \) of \( \text{nFC}_p(x) \) such that there is a functional path from any \( y \in X \) to any \( y' \in X \).

When ACCLF is replaced with ACCLF\(^n\), then the implication holds without the connectedness assumption.

Proof. Fix an \( f \)-acyclic and connected UCQ \( q(x) \) and let \( \mathcal{T} \) be a ACCLF TBox. Let \( Q = (T, \Sigma_{\text{full}}, q(x)) \).
We construct an ACCLF-IQ \( C(x) \) such that \( Q \equiv Q' \) for \( Q' = (T, \Sigma_{\text{full}}, C(x)) \).
Let \( p(x) \) be a CQ in \( q(x) \). A cluster in \( p(x) \) is any maximal subset \( X \) of \( \text{nFC}_p(x) \) such that there is a functional path from any \( y \in X \) to any \( y' \in X \).

When ACCLF is replaced with ACCLF\(^n\), then the implication holds without the connectedness assumption.

Proof. Fix an \( f \)-acyclic and connected UCQ \( q(x) \) and let \( \mathcal{T} \) be a ACCLF TBox. Let \( Q = (T, \Sigma_{\text{full}}, q(x)) \).
We construct an ACCLF-IQ \( C(x) \) such that \( Q \equiv Q' \) for \( Q' = (T, \Sigma_{\text{full}}, C(x)) \).
Let \( p(x) \) be a CQ in \( q(x) \). A cluster in \( p(x) \) is any maximal subset \( X \) of \( \text{nFC}_p(x) \) such that there is a functional path from any \( y \in X \) to any \( y' \in X \).

When ACCLF is replaced with ACCLF\(^n\), then the implication holds without the connectedness assumption.
the following holds: for all \(0 \leq i < n\), there is a functional path \(y^0_i, \ldots, y^n_i \) in \(p^i|X\) such that \(y^0_i = y'_i, y^n_i = x^i_{i+1}\) and \(y^i_k = x^i_{2^k+1}\). Now take for every \(0 \leq i < n\) fresh concept names \(P^1_{X,i}, P^2_{X,i}, P^3_{X,i}\) and obtain \(p^i_{\text{plain}}(x)\) from \(p^i_{\text{plain}}(x)\) by adding the compound atoms

\[
P^1_{X,i} \cup P^2_{X,i} \cup P^3_{X,i}(y^i_0)
\]

Take functional roles \(s^i_{-1}, \ldots, s^i_{k-2}\) and consider the query

\[
s^i_0(y^i_0, y^i_1), \ldots, s^i_{k-2}(y^i_{k-2}, y^i_{k-1}) \in p^i|X
\]

and consider the CQ \(p^i_X(y^i_0)\) defined by the conjunction of the atoms \(s^i_0(y^i_0, y^i_1), \ldots, s^i_{k-2}(y^i_{k-2}, y^i_{k-1})\) and taking \(y^i_0\) as the answer variable. Define for \(1 \leq j \leq 3\) an eCQ \(p^i_X(y^i_0)\) by adding to \(p^i_X(y^i_0)\) the compound atom

\[
\exists r_{i+1}.P^i_{X,i}(y^i_{k-1})
\]

Take the ECI concept \(D^i_{X,i}\) corresponding to \(p^i_{\text{deco}}(x)\), for \(j = 1, 2, 3\). Obtain the CQ \(p^i_{\text{deco}}(x)\) from \(p^i_{\text{deco}}(x)\) by adding the compound atoms

\[
\prod_{1 \leq j \leq 3}(p^i_{\text{deco}}(x) \rightarrow D^i_{X,i}(y^i_0))
\]

We do this for all non-degenerate clusters \(X\) and obtain eCQs \(p^i_{\text{plain}}(x)\) and \(p^i_{\text{deco}}(x)\) by taking the conjunction of all \(p^i_{\text{plain}}(x)\) and \(p^i_{\text{deco}}(x)\), respectively. Now define the concepts \(C^i_{\text{plain}}\) and \(D^i_{\text{deco}}\) in the obvious way. It is not difficult to prove that \((\mathcal{T}, \Sigma_{\text{full}}; C^i_{\text{plain}} \rightarrow D^i_{\text{deco}}(x))\) is a rewriting of \((\mathcal{T}, \Sigma_{\text{full}}, p(x))\). By constructing \(C^i_{\text{plain}}\) and \(D^i_{\text{deco}}\) for all CQs \(p(x)\) in \(q(x)\) and taking the disjunction of all \(C^i_{\text{plain}} \rightarrow D^i_{\text{deco}}(x)\) we obtain a rewriting of \(q(x)\). The extension to \(ALCI^{-}\)without the connectedness assumption is straightforward.

For the proof of the other direction of Theorem 13 we require some preparation. We use standard notation and results for ultrafilter extensions of interpretations [Blackburn et al., 2002]. For \(U \subseteq \Delta^X\) for an interpretation \(\mathcal{I}\) we set \(\mathcal{I} U = \Delta^X \setminus U\).

**Definition 30.** Let \(\mathcal{I}\) be an interpretation. A set \(\mathfrak{U} \subseteq 2^{\Delta^X}\) is a ultrafilter over \(\Delta^X\) if the following conditions hold for all \(U, V \subseteq \Delta^X:\)

- if \(U, V \in \mathfrak{U}\), then \(U \cap V \in \mathfrak{U}\);
- if \(U \in \mathfrak{U}\) and \(U \subseteq V\), then \(V \in \mathfrak{U}\);
- if \(U \in \mathfrak{U}\) iff \(\mathcal{I} U \not\in \mathfrak{U}\).

For every \(d \in \Delta^X\), the set

\[
\mathfrak{U}_d = \{X \subseteq \Delta^X \mid d \in X\}
\]

is an ultrafilter, called the principal ultrafilter generated by \(d\) in \(\Delta^X\). An ultrafilter \(\mathfrak{U}\) for which there exists no \(d \in \Delta^X\) with \(\mathfrak{U} = \mathfrak{U}_d\) is called a non-principal ultrafilter. It is known that for every set \(\mathfrak{V} \subseteq 2^{\Delta^X}\) with the finite intersection property (if \(U_1, \ldots, U_n \in \mathfrak{V}\), then \(U_1 \cap \cdots \cap U_n \neq \emptyset\)) there exists an ultrafilter \(\mathfrak{U} \supseteq \mathfrak{V}\). For a set \(U \subseteq \Delta^X\) and role \(r\) we set

\[
(\exists r.U)^\mathfrak{U} = \{d \in \Delta^X \mid \text{there exists } d' \in U \text{ with } (d, d') \in r^\mathfrak{U}\}
\]

We are in the position now to define ultrafilter extensions of interpretations.

**Definition 31.** Let \(\mathcal{I}\) be an interpretation. The ultrafilter extension \(\mathcal{I}U\) of \(\mathcal{I}\) is defined as follows:

- \(\Delta^{\mathcal{I}U}\) is the set of ultrafilters over \(\Delta^X\);
- \(\mathfrak{U} \in \Delta^{\mathcal{I}U}\) iff \(\mathcal{I} U \in \mathfrak{U}\), for all concept names \(A\);
- \((\mathfrak{U}_1, \mathfrak{U}_2) \in \Delta^{\mathcal{I}U}\) iff \((\exists r.U)^\mathfrak{U}_1 \in \mathfrak{U}_2\) for all \(U \in \mathfrak{U}_1\), for all role names \(r\).

Observe the following equivalence for all principal ultrafilters \(\mathfrak{U}_d, \mathfrak{U}_e\) and all roles \(r\):

\[
(d, e) \in r^{\mathfrak{U}_d} \Leftrightarrow (d, e) \in r^{\mathfrak{U}_e}
\]

The fundamental property of ultrafilter extensions is the following anti-preservation result:

**Lemma 32.** For all interpretations \(\mathcal{I}\), \(ALCI^{-}\)concept \(C\), and roles \(r\):

- if \(\mathcal{I} \models C(a)\), then \(\mathcal{I}U \models C(U_a)\);
- if \(r^\mathfrak{U}\) is functional, then \(r^{\mathcal{I}U}\) is functional.

We are now in the position to prove the second part of Theorem 13.

**Lemma 33.** If an OMQ \(Q = (\mathcal{T}, \Sigma_{\text{full}}, q(x))\) from \((\mathcal{T}, \Sigma_{\text{full}}, C(x))\) is rewrittable into an OMQ from \((\mathcal{T}, \Sigma_{\text{full}}, C(x))\), then there is a subquery \(q'(x)\) of \(q(x)\) that is \(r\)-acyclic, connected, and equivalent to \(q(x)\).

When \(ALCI^{-}\)is replaced with \(ALCI^{\mathcal{I}U}\)-IQ, then the same equivalence holds except that connectedness is dropped.

**Proof.** Assume that \(\mathcal{T}\) contains functionality assertions only and \(q(x)\) is a UQ such that there does not exist an equivalent subquery \(q'(x)\) of \(q(x)\) which is \(r\)-acyclic and connected. We may assume that

- there is no homomorphism from any disjunct \(p(x)\) of \(q(x)\) to another disjunct \(p'(x)\) of \(q(x)\);
- every homomorphism from any disjunct \(p(x)\) into itself is surjective.

Take a disjunct \(p(x)\) of \(q(x)\) which is not \(r\)-acyclic or not connected. We consider the case that \(p(x)\) is not \(r\)-acyclic but connected. The case that \(p(x)\) is not connected is straightforward. We have a cycle \(r_0(x_0, x_1), \ldots, r_{n-1}(x_{n-1}, x_n)\) in \(p(x)\) such that \(FC_q(x) \cap \{x_0, \ldots, x_{n-1}\} = \emptyset\) and

1. \(r_i\) or \(r^{-}\) is not functional w.r.t. \(\mathcal{T}\) for some \(i < n\) or
2. there exists no functional path \(y_0, \ldots, y_m\) in \(p(x)\) with \(x_0 = y_0 = y_m\) such that \(\{x_0, \ldots, x_{n-1}\} \subseteq \{y_0, \ldots, y_m\}\).

The basic idea of the proof for both Point 1 and Point 2 is as follows. Assume there exists a rewriting \((\mathcal{T}, \Sigma_{\text{full}}, C(x))\) of \((\mathcal{T}, \Sigma_{\text{full}}, q(x))\).

(a) Using the CQ \(p(x)\) we construct an infinite ABox \(\mathcal{A}\) such that \(\mathcal{T}, \mathcal{A} \models q(x)\);

(b) By compactness of FO there exists a forest model \(\mathcal{I}\) of \(\mathcal{T}\) and \(\mathcal{A}\) with \(\mathcal{I} \models \neg C(a)\);

(c) Then \(\mathcal{I}U \models \neg C(U_a)\) by Lemma 32.
Moreover, $\mathcal{T}^w$ is a model of a finite ABox $A'$ with individual name $\mathcal{U}_0$ such that $\mathcal{T}, A' \models q(\mathcal{U}_0)$. But this contradicts $\mathcal{T}^w \models \neg C(\mathcal{U}_0)$.

For Point 1, the construction of $A$ is a variant of a construction given in [Kikot and Zolin, 2013]. Consider the cycle $r_0(x_0, x_1), \ldots, r_{n-1}(x_{n-1}, x_n)$ in $p(x)$ such that for some $i$ neither $r'$ nor $r''$ are functional w.r.t. $\mathcal{T}$. We may assume that $i = 0$. Let $V$ be the connected component of $\{x_0, \ldots, x_{n-1}\}$ in $\text{nFC}_p(x)$. Regard the variables of $p(x)$ as individual names. Define an ABox $A$ with individuals $\text{FC}_p(x) \cup (\text{nFC}_p(x) \setminus V) \cup (V \times \mathbb{N})$ by setting:

- for all concept names $A$ and variables $y \in \text{FC}_p(x) \cup (\text{nFC}_p(x) \setminus V)$: $A(y) \in A$ iff $A(y)$ is in $p(x)$;
- for all concept names $A$, variables $y \in V$, and $i \in \mathbb{N}$: $A(y, i) \in A$ iff $y \in q(x)$;
- for all roles $r$ and variables $y, z \in \text{FC}_p(x) \cup (\text{nFC}_p(x) \setminus V)$: $r(y, z) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y \in \text{FC}_p(x) \cup (\text{nFC}_p(x) \setminus V)$ and $z \in V$, and $i \in \mathbb{N}$: $r(y, i, z) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y, y' \in V$, and $i \in \mathbb{N}$: $r(y, i, y') \in A$ iff $r(y, y')$ is in $p(x)$ and $r_0(x_0, x_1) \neq r(y, y')$;
- for all roles $r$, variables $y, y' \in V$, and $i \neq j \in \mathbb{N}$: $r((y, i), (y', j)) \in A$ iff $i < j$ and $r_0(x_0, x_1) = r(y, y')$.

We now check that $A$ satisfies Points (a) to (d).

Point (a). Clearly all $r$ functional w.r.t. $\mathcal{T}$ are functional in $A$. Thus, $\mathcal{T}, A \models q(x)$ iff $\emptyset, A \models p'(x)$ for some disjunct $p'(x)$ of $q(x)$. It therefore suffices to show that there is no homomorphism from any disjunct $p'(x)$ of $q(x)$ to $A$ mapping $x$ to $x$. It has been observed in [Kikot and Zolin, 2013] already that the mapping $\pi$ from $A$ to $p(x)$ mapping every variable to itself and every $(y, i) \in V \times \mathbb{N}$ to $y$ is a homomorphism. Using this observation it has been shown that there is no homomorphism from $p(x)$ to $A$ as the composition of $\pi$ with such a homomorphism would be a non-surjective homomorphism mapping $x$ to $x$ which contradicts our assumptions. It also follows that there is no homomorphism from any disjunct $p'(x)$ of $q(x)$ distinct from $p(x)$ to $A$ as the composition of $\pi$ with such a homomorphism would be a homomorphism from $p'(x)$ to $p(x)$ mapping $x$ to $x$.

Point (b). Assume there exists no model of $\mathcal{T}$ and $\mathcal{A}$ with $\mathcal{I} \models \neg C(x)$. Then, by compactness, there exists a finite subset $A'$ of $\mathcal{A}$ such that there exists no model of $\mathcal{T}$ and $\mathcal{A}'$ with $\mathcal{I} \models \neg C(x)$. But then $\mathcal{T}, \mathcal{A}' \models q(x)$ which contradicts Point (a) and the assumption that $(\mathcal{T}, \Sigma_{\text{full}}, C(x))$ is a rewriting of $(\mathcal{T}, \Sigma_{\text{full}}', q(x))$.

Point (c). This is by Lemma [32].

Point (d). It follows directly from [Kikot and Zolin, 2013] that $\mathcal{T}^w$ contains a homomorphic image of $p(x)$ under a homomorphism mapping $x$ to $\mathcal{U}_0$. Regard this image as an ABox $A'$. Then $\mathcal{T}, A' \models q(\mathcal{U}_0)$. This finishes the proof if Point 1 holds.

Now suppose that Point 2 does not hold. Thus, there exists no functional path $y_1, \ldots, y_m$ in $p(x)$ with $x_0 = y_0 = y_m$ such that $\{x_0, \ldots, x_{n-1}\} \subseteq \{y_0, \ldots, y_m\}$. Let $p'' = p|_{\text{nFC}_p(x)}$. We observe the following

**Claim 1.** There exist $V_1 \subseteq \text{nFC}_p(x)$ of the form $V_2 = \text{FC}_p(x')$ for some $x'$ in $p(x)$ such that for $V_2 := \text{nFC}_p(x')$, there are $y_1, y_2 \in V_2$ such that there is a path from $y_1$ to $y_2$ in $V_2$ and $z_1, z_2 \in V_1$ such that there are distinct $s_1(y_1, z_1)$, $s_2(y_2, z_2)$ in $q(x)$ with $s_1, s_2$ functional w.r.t. $\mathcal{T}$.

For the proof of Claim 1 take the cycle $r_0(x_0, x_1), \ldots, r_{n-1}(x_{n-1}, x_n)$ in $p(x)$. There must exist $x_1$ such that some $x_j$ with $x_1 \neq x_j$ is not in $\text{FC}_p(x_1)$. Let $V_1 = \text{FC}_p(x_1)$. Then we find $x_j \in \text{FC}_p(x_1)$ such that $x_{j+1} \notin \text{FC}_p(x_1)$ and we find a path (possibility of length 0) from $x_{j+1}$ to some $x_j$ within $V_2$ such that there exist $x'$ and $x''$ within $V_1$ with $r''(x', x'') \in p(x)$. Then $s_1(y_1, z_1) := r''(x_{j+1}, x_j)$ and $s_2(y_2, z_2) := r''(x', x'')$ are as required.

We define an ABox $A$ with individual names $\text{FC}_p(x) \cup (V_1 \times \mathbb{N}) \cup (V_2 \times I)$ where $I = \{(x, z) \mid z \in V_1, y \in V_2\}$ as follows:

- for all concept names $A$ and variables $y \in \text{FC}_p(x)$: $A(y) \in A$ iff $A(y)$ is in $p(x)$;
- for all roles $r$ and variables $y, z \in \text{FC}_p(x)$: $r(y, z) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all concept names $A$, variables $y \in V_1$, and $i \in \mathbb{N}$: $A(y, i) \in A$ iff $A(y)$ is in $p(x)$;
- for all concept names $A$, variables $y \in V_2$, and $i \in I$: $A(y, i) \in A$ iff $A(y)$ is in $p(x)$;
- for all roles $r$, variables $y, z \in V_1$ and $i \in \mathbb{N}$: $r((y, i), (z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y, z \in V_2$ and $i \in I$: $r((y, i), (z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y, z \in V_2$ and $i \in I$: $r((y, i), (z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y, z \in \text{FC}_p(x)$ and $z \in V_1$, and $i \in I$: $r((y, z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, variables $y, z \in \text{FC}_p(x)$ and $z \in V_2$, and $i \in I$: $r((y, z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, $(y, (x, z)) \in (V_2 \times I)$ and $(z, i) \in V_1 \times \mathbb{N}$: $r((y, (x, z)), (z, i)) \in A$ iff $r(y, z)$ is in $p(x)$;
- for all roles $r$, $(y, (x, z)) \in (V_2 \times I)$ and $(z, i) \in V_1 \times \mathbb{N}$: $r((y, (x, z)), (z, i)) \in A$ iff $r(y, z)$ is in $p(x)$.

This construction of the ABox $A$ achieves the following:

- for all copies $V_1'$ of $V_2$ and any $r(y, z) \in S$, there is a copy $V_1'$ of $V_1$ such that $r(y', z') \in A$ for the copies $y', z'$ of $y$ and $z$ in $V_1'$ and $V_1'$, respectively;
• for all copies $V'_2$ of $V_2$ and copies $V'_1$ of $V_1$ there is at most one atom $r(y, z) \in A$ with $y \in V_2$ and $z \in V_1$;

• Let $V'_1, \ldots, V''_1$ be distinct copies of $V_2$ and $r_1(y_1, z_1), \ldots, r_n(y_n, z_n)$ be distinct atoms in $S$. Then there is a single copy $V'_1$ of $V_1$ such that $r_1(y'_1, z'_1), \ldots, r_n(y'_n, z'_n) \in A$ for the copies $y'_1, \ldots, y'_n$ of $y_1, \ldots, y_n$ in $V'_1, \ldots, V''_1$, respectively, and the copies $z'_1, \ldots, z'_n$ of $z_1, \ldots, z_n$ in $V'_1$.

We first show Point (a) above.

Point (a). $T, A \models q(a)$.

Proof of Point (a). By construction, all $r$ functional w.r.t. $T$ are functional in $A$. Thus, $T, A \models q(x)$ iff $\emptyset, A \models p'(x)$ for some disjunct $p'(x)$ of $q(x)$. It therefore suffices to show that there is no homomorphism from any disjunct $p'(x)$ of $q(x)$ to $A$ mapping $x$ to $x$. Consider the mapping

$$
\pi : A \to p(x)
$$

mapping every variable in $FC_p(x)$ to itself and every $(y, i) \in (V_1 \times N) \cup (V_2 \times I)$ to $y$. It is easy to see that $\pi$ is a homomorphism. It also follows that there is no homomorphism from any disjunct $p'(x)$ of $q(x)$ distinct from $p(x)$ to $A$ as the composition of $\pi$ with such a homomorphism would be a homomorphism from $p'(x)$ to $p(x)$ mapping $x$ to $x$. It remains to prove that there is no homomorphism from $p(x)$ to $A$ mapping $x$ to $x$. Assume there is such a homomorphism $h$. Then $\pi \circ h$ is a homomorphism from $p(x)$ to $p(x)$ mapping $x$ to $x$. We obtain a contradiction if we can show that $\pi \circ h$ is not surjective. To this end assume that $\pi \circ h$ is surjective. As $p(x)$ is finite, it is an isomorphism. Let $h[p(x)] = \{h(y) | y \in p(x)\}$ be the image of $h$ in $A$. Then $h$ is an isomorphism from $p(x)$ onto the restriction $A|_{h[p(x)]}$ of $p(x)$ to $h[p(x)]$ and the restriction $\pi|_{h[p(x)]}$ of $\pi$ to $h[p(x)]$ is an isomorphism onto $p(x)$. It follows that $h[p(x)]$ contains for every $y$ in $p(x)$ exactly one individual $a$ with $\pi(a) = y$ and

• $h[p(x)]$ contains $FC_p(x)$;

• as $V_1$ is connected, there exists $i \in N$ such that $\pi(h[p(x)]) \supseteq V \times \{i\}$ and $h(p(x)) \cap V \times \{j\} = \emptyset$ for all $j \neq i$;

• for every connected component $V$ of $V_2$ of there exists $i \in I$ such that $h(p(x)) \supseteq V \times \{i\}$ and $h(p(x)) \cap V \times \{j\} = \emptyset$ for all $j \neq i$.

Now recall that there are distinct atoms $s_1(y_1, z_1) \in p(x)$ and $s_2(y_2, z_2) \in p(x)$ such that $y_1, y_2$ are in the same connected component in $V_2$ and $z_1, z_2 \in V_1$. Thus, there exists $i \in N$ such that $(y_1, i), (y_2, i) \in h[p(x)]$ and for some $j \in I$, $s_1((y_1, i), (z_1, j)) \in A$ and $s_2((y_2, i), (z_2, j)) \in A$. As observed above, no such two atoms exist in $A$ and we have derived a contradiction. This finishes the proof of Point (a).

**Example 34.** Let $T = \{\text{func}(s_1), \text{func}(s_2)\}$ and consider the CQ

$$q(x) = \exists y, z(r(x, y) \land s_1(y, z) \land s_2(y, z))$$

Then

$$FC_q(x) = \{x\}, \quad V_1 = \{z\}, \quad V_2 = \{y\}$$

Thus,

$$S = \{s_1(y, z), s_2(y, z)\}$$

and so the individuals of $A$ are

$$\{x\} \cup \{(z) \times N\} \cup \{(y) \times I\}$$

and the essential properties of $A$ are:

• $r(x, (y, i)) \in A$ for all $i \in I$;

• for every $(y, i) \in I$ there are distinct $(z, i_1), (z, i_2)$ with $s_1((y, i), (z, i_1)), s_2((y, i), (z, i_2)) \in A$;

• for any two $(z, i_1), (z, i_2)$ there exists $(y, i)$ such that $s_1((y, i), (z, i_1)), s_2((y, i), (z, i_2)) \in A$.

Observe that $A \not\models q(x)$.

Points (b) and (c) are as before. It remains to show Point (d). Let $I$ be a model of $T$ and $A$ with $I \models \neg C(a)$ and consider the ultrafilter $I^w$. We define a homomorphism $h$ from $p(x)$ to $I^w$ mapping $x$ to $U$. For every $y \in FC_p(x)$, we set $h(y) = U$. To define $h$ for the remaining variables, fix a non-principal ultrafilter $\mathcal{U}$ over $\mathbb{N}$. For every variable $y \in V_1$ we obtain an ultrafilter $\mathcal{N}(y)$ over $\Delta^2$ by setting $U \in \mathcal{N}(y)$ iff $\{i | (z, i) \in U \cap (V_1 \times N)\} \in \mathcal{N}$. Observe that for $z_1, z_2 \in V_1, r(z_1, z_2) \in p(x)$ implies $(\mathcal{N}(z_1), \mathcal{N}(z_2)) \in \mathcal{N}$. Observe as well that for $y \in FC_p(x)$ and $z \in V_1, r(y, z) \in p(x)$ implies $(\mathcal{U}_y, \mathcal{N}(y)) \in \mathcal{N}$. We set $h(z) = \mathcal{N}(z)$ for $z \in V_1$. It remains to define $h$ for variables in $V_2$.

Let for $y \in V_2$, and $X \subseteq \Delta^2, \rho_y(X) = \{i \in I | (y, i) \in X\}$. By construction, the set

$$X = \rho_y(\exists r. Z) \mid r(y, z) \in S \land z \in \mathcal{N}(z)$$

has the finite intersection property. Thus, there exists an ultrafilter $\mathcal{I}$ over $I$ containing $X$. For every variable $y \in V_2$ we obtain an ultrafilter $\mathcal{I}(y)$ over $\Delta^2$ by setting $U \in \mathcal{I}(y)$ iff $\{i | (y, i) \in U \cap (V_2 \times I)\} \in \mathcal{I}$. Observe that for $y_1, y_2 \in V_1, r(y_1, y_2) \in p(x)$ implies $\mathcal{I}(y_1, Y(2)) \in \mathcal{I}$. Observe as well that for $y \in FC_p(x)$ and $y' \in V_2, r(y, y') \in p(x)$ implies $(\mathcal{U}_y, \mathcal{I}(y')) \in \mathcal{I}$. Finally, observe that by construction $r(y, z) \in S$ implies that $\mathcal{I}(y), \mathcal{N}(z) \in \mathcal{I}$. We set $h(y) = \mathcal{I}(y)$ for $y \in V_2$. It follows that $h$ is a homomorphism from $p(x)$ to $I^w$, as required.

Now let $A'$ be the image of $p(x)$ under $h$, regarded as an ABox. Then $T, A' \models q(\mathcal{U}_x)$, as required.

The proof when $\mathcal{ALCF}$ is replaced with $\mathcal{ALCF}^*$-IQ and connectedness is dropped is a straightforward variation of the proof above.

**Theorem 11.** In $(\mathcal{ALCF}, CQ)$, IQ-rewritability is undecidable.

**Proof.** We use a reduction from emptiness checking in $(\mathcal{ALCF}, AQ)$ which is known to be undecidable [Baader et al., 2016]. Let $Q = (T, \Sigma, A(x))$ be an OMQ from this language and let $q(x) = \exists y A(x) \land r(x, y) \land r(y, y)$, where $r$ is a role name that does not occur in $T$. We show that $Q$ is empty iff $Q' = (T, \Sigma \cup \{r\}, q(x))$ is IQ-rewritable. Clearly, if $Q$ is empty, then $Q'$ is empty and, therefore, IQ-re writable. Conversely, if $Q$ is not empty, then we show that
Q’ is not IQ-rewritable. To this end we take a Σ-ABox A₀ and a ∈ ind(A) such that A₀ |= Q(a) and A₀ is consistent with T. Assume for a proof by contradiction that there is a rewriting Q’” = (T’, Σ ∪ {r}, C(x)) of Q’. We use a minor modification of the construction in the proof of Theorem[3]

(a) Using the CQ q(x) and A₀ we construct an infinite (∆ ∪ {r})-ABox A ⊇ A₀ such that A ̸|= Q’(a);
(b) By compactness of FO there exists a forest model I of T’ and A with I |= ¬C(a);
(c) Then T[4] |= ¬C(∪a) and I |= T’ by Lemma[2];
(d) Moreover, T[4] is a model of a finite (∆ ∪ {r})-ABox A’ with individual name ∪a such that A’ |= Q’(∪a). But this contradicts T[4] |= ¬C(∪a).

The construction of A is the same as in Point 1 of the proof of Theorem[3] above using the query q(x) and its cycle r(y, y) except that we also attach A₀ to the ABox constructed by identifying x and a. Now the proof of Points (a) to (d) is exactly as before.

E Proofs for Section 6

We start with definitions of MMSNP and CSP together with some preliminaries. We consider signatures S that consist of predicate symbols with unrestricted arity, known as schemas. An S-fact is an expression of the form S(a₁, ..., aₙ) where S ∈ S is an n-ary predicate symbol, and a₁, ..., aₙ are elements of some fixed, countably infinite set const of constants. An S-instance I is a set of S-facts. The domain of I, denoted dom(I), is the set of constants that occur in some fact in I. The notions of cycles, girth, acyclicity, and connectedness can be lifted from ABoxes to S-instances, for details see [Feier et al., 2017]. We use CONI to denote the set of S-instances that are the maximal connected components of the S-instance I.

An MMSNP sentence ϕ over schema S has the form

∃X₁ ... ∃Xₙ ∀x₁ ... ∀xₘ ψ,

with X₁, ..., Xₙ monadic second-order (SO) variables, x₁, ..., xₘ first-order (FO) variables, and ψ a conjunction of quantifier-free formulas of the form

α₁ ∧ ... ∧ αₙ → β₁ ∨ ... ∨ βₘ with n, m ≥ 0, (¢)

where each αᵢ is of the form Xᵢ(xᵢ) or R(x) with R from S, and each βᵢ is of the form xᵢ. We refer to a formula of the form (¢) as a rule in ϕ, to the conjunction α₁ ∧ ... ∧ αₙ as its body, and to β₁ ∨ ... ∨ βₘ as its head. A rule body can be seen as an S ∪ {X₁, ..., Xₙ}-instance in the obvious way, which we shall sometimes do implicitly. An MMSNP sentence ϕ is connected if the body of every rule in ϕ is connected. The rule size of ϕ is the maximum size of a rule in ϕ.

Every MMSNP sentence ϕ can be seen as a Boolean query in the obvious way, that is, for an S-instance I, I |= ϕ whenever ϕ evaluates to true on I. We also consider disjunctions of MMSNP sentences over schema S, that is, sentences of the form ∨i ϕᵢ, where each ϕᵢ is an MMSNP sentence over S. For an S-instance I, I |= ∨i ϕᵢ, whenever there exists some i such that ϕᵢ evaluates to true on I. An MMSNP sentence ϕ over S is contained in an MMSNP sentence ϕ’ over S, written ϕ ⊆ ϕ’, if for every S-instance I, I |= ϕ implies I |= ϕ’. We say that ϕ₁ and ϕ₂ are equivalent if ϕ₁ ⊆ ϕ₂ and ϕ₂ ⊆ ϕ₁. Containment and equivalence are defined in the same way for disjunctions of MMSNP sentences.

A constraint satisfaction problem (CSP) is defined by an S-instance T that is called the template. The problem associated with T is to decide whether an input S-instance I admits a homomorphism to T, denoted I → T. An MMSNP sentence ϕ over schema S is said to be CSP-definable if there exists an S-template T such that for every S-instance I, I |= ϕ iff I → T. A generalized CSP over schema S is defined by a finite set T = {T₁, ..., Tₙ} of S-templates. For an S-instance I, we write I → T if I → T₁ for some i. An MMSNP sentence ϕ over schema S is definable by a generalized CSP T if for every S-instance I, I |= ϕ iff I → T.

For two S-instances I₁ and I₂ with disjoint domains, we use I₁ ∪ I₂ to denote the disjoint union of I₁ and I₂. An MMSNP sentence ϕ is preserved under disjoint union if for all S-instances I₁ and I₂ with disjoint domains, I₁ |= ϕ and I₂ |= ϕ implies I₁ ⊕ I₂ |= ϕ.

Lemma 35. An MMSNP sentence ϕ is CSP-definable if it is definable by a generalized CSP and preserved under disjoint union.

Proof. “⇒”. Clear.

“⇐”. Assume that ϕ is definable by a generalized CSP T = {T₁, ..., Tₙ} and preserved under disjoint union. For every i, Tᵢ → T and thus Tᵢ |= ϕ. Assume w.l.o.g. that the domains of all templates in T are mutually disjoint. As ϕ is preserved under disjoint union, ∪i ≤ n Tᵢ |= ϕ. Thus, ∪i ≤ n Tᵢ → T and consequently ∪j ≤ n Tᵢ → Tj for some j. This implies that Tᵢ → Tj for every i and thus the generalized CSP T is equivalent to the CSP Tj, which finishes the proof.

We next determine the complexity of deciding preservation under disjoint union for MMSNP sentences. For our final aim, we only need the upper bound, but we also observe a lower bound for the sake of completeness.

Theorem 36. Deciding whether an MMSNP sentence is preserved under disjoint union is 2NExpTime-complete.

Proof. For the upper bound, we reduce preservation under disjoint union to a series of (exponentially many) containment checks between MMSNP sentences (of polynomial size) and then invoke the result from [Bourhis and Lutz, 2016] that MMSNP containment can be decided in 2NExpTime.

Let ϕ be an MMSNP sentence over schema S and N the set of nullary predicate symbols in ϕ. We assume w.l.o.g. that the number of first-order variables in ϕ is bounded from below by the largest arity of a predicate in S. For all N, N₁ ⊆ N, we construct an MMSNP sentence ϕN₁, N₂ as follows:

ϕN₁, N₂ has the same quantifiers as ϕ except for two new existentially quantified second-order variables C₁ and C₂:

We do not assume here that I₁ and I₂ contain the same nullary predicate symbols; I₁ ⊕ I₂ contains the union of the nullary symbols in I₁ and I₂.
\[ \varphi_{N_1, N_2} \text{ has the following rules:} \]
- \( \text{true} \rightarrow C_1(x) \lor C_2(x); \)
- \( C_1(x) \lor C_2(x) \rightarrow \text{false}; \)
- \( R(y_1, \ldots, y_n) \land C_i(y_j) \rightarrow C_i(y_k) \) whenever \( R \in S \) is \( n \)-ary, \( i \in \{0, 1\} \), and \( j, k \in \{1, \ldots, n\} \) and where \( y_1, \ldots, y_n \) are the first \( n \) FO variables in \( \varphi \);
- \( C_1(x_1) \land \cdots \land C_2(x_n) \land \text{body} \rightarrow \text{head} \) whenever \( \text{body} \rightarrow \text{head} \) is a rule in \( \varphi \) with no predicate symbol from \( N \setminus N_i \) occurring in \( \text{body} \) and \( i \in \{0, 1\} \) and where \( x_1, \ldots, x_n \) are the FO variables in \( \text{body} \).

Intuitively, an \( S \)-instance \( I \) satisfies \( \varphi_{N_1, N_2} \) iff there is a coloring of \( I \) with the two colors \( C_1 \) and \( C_2 \) such that elements from the same maximal connected components receive the same color and each of the resulting two monochromatic subinstances of \( I \) satisfies \( \varphi \). Note that \( I \) is the disjoint union of \( I_1 \) and \( I_2 \). The sets \( N_1, N_2 \) help to disentangle the nullary predicate symbols: \( N_1 \) contains the predicates true in the monochromatic subinstance colored \( C_1 \) and likewise for \( N_2 \) and \( C_2 \).

**Claim.** \( \varphi \) is preserved under disjoint union iff \( \varphi = \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \).

\( \Rightarrow \). Assume that \( \varphi \) is preserved under disjoint union. We have to show the following inclusions:

- \( \varphi \subseteq \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \).
- \( \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \subseteq \varphi \).

\[ \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \subseteq \varphi \] holds even when \( \varphi \) is not preserved under disjoint union (as shown by the proof above) and thus deciding whether \( \varphi \) is preserved under disjoint union amounts to checking that \( \varphi \supseteq \forall_{N_1, N_2} \varphi_{N_1, N_2} \) for all \( N_1, N_2 \subseteq N \).

\( \leftarrow \). Assume that \( \varphi \subseteq \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \). Let \( I_1 \) and \( I_2 \) be \( S \)-instances with disjoint domain such that \( I_1 \models \varphi \) and \( I_2 \models \varphi \). Then \( I_1 \uplus I_2 \models \varphi_{N_1, N_2} \), where \( N_i \) is the set of nullary predicates true in \( I_i \), \( i \in \{1, 2\} \). Thus, \( I_1 \uplus I_2 \models \forall_{N_1, N_2 \subseteq N} \varphi_{N_1, N_2} \) and from the original assumption \( I_1 \uplus I_2 \models \varphi \). This finishes the proof of the claim.

For the lower bound, we consider the (polynomial time) reduction from [Bourhis and Lutz, 2016] from a 2NEPTIME-hard torus tiling problem to OMQ containment: there, two OMQs \( Q_1 \) and \( Q_2 \) are constructed, \( Q_1 \) from \( (ACCT, BAQ) \) and \( Q_2 \) from \( (ACCT, BCQ) \), such that a double exponentially large torus can be tiled iff \( Q_1 \subseteq Q_2 \). We sketch a polynomial time reduction from the containment problem for two such OMQs to the preservation under disjoint union of an MMSN sentence \( \varphi \). For the sake of proving the correctness of our reduction, we note that \( Q_1 \) and \( Q_2 \) are such that \( Q_2 \nsubseteq Q_1 \).

It was shown in [Bienvenu et al., 2013] that every OMQ from \( (ACCT, BAQ) \) such as \( Q_1 \) is equivalent to the complement of a CSP \( T \) in the sense that for every \( \Sigma \)-ABox \( A \), \( A \models Q_1 \iff A \not\models \varphi \) where \( \Sigma \) is the signature of the OMQs \( Q_1 \) and \( Q_2 \). Note that this is a variation of Lemma 21 for the case of BAQs. It has also been shown in [Bienvenu et al., 2014] that for every OMQs from \( (ACCT, BCQ) \), one can construct in polynomial time an MMSN sentence whose complement is equivalent to the OMQ and thus for \( Q_1 \) and \( Q_2 \) we find two such sentences \( \varphi_1 \) and \( \varphi_2 \). The size of \( \varphi_1 \) and \( \varphi_2 \) is polynomial in that of \( Q_1 \) and \( Q_2 \). Summing up, the 2-exponent torus can be tiled iff \( \varphi_2 \nsubseteq \varphi_1 \) and \( \varphi_1 \) is equivalent to a CSP, and \( \varphi_1 \nsubseteq \varphi_2 \). We next construct an MMSN sentence \( \varphi \) such that for all \( S \)-instances \( I \):

\[ I \not\models \varphi \text{ iff } I \not\models \varphi_1 \text{ and } I \not\models \varphi_2. \]

Towards constructing \( \varphi \), we start by standardizing apart all FO and SO variables from \( \varphi_1 \) and \( \varphi_2 \). For each \( i \in \{1, 2\} \), let \( \psi_i \) be the quantifier-free part of \( \varphi_i \) with all rules of the form \( \text{body} \rightarrow \text{false} \) removed. Then \( \varphi \) is the MMSN sentence which has as SO/FO variables the union of SO/FO variables from \( \varphi_1 \) and \( \varphi_2 \) and the following rules:

- all rules from \( \psi_1 \) and from \( \psi_2 \),
- all rules of the form \( \text{body}_1 \land \text{body}_2 \rightarrow \text{false} \), where \( \text{body}_1 \rightarrow \text{false} \) is a rule in \( \varphi_1 \) and \( \text{body}_2 \rightarrow \text{false} \) a rule in \( \varphi_2 \).

It can be verified that \( \varphi \) satisfies (1).

**Claim.** \( \varphi_2 \nsubseteq \varphi_1 \) iff \( \varphi \) is preserved under disjoint union.

\( \Rightarrow \). Assume that \( \varphi_2 \nsubseteq \varphi_1 \). Then, \( \varphi \) is equivalent to \( \varphi_1 \) and thus to a CSP, consequently it is preserved under disjoint union.

\( \Leftarrow \). Assume that \( \varphi_2 \nsubseteq \varphi_1 \). Thus there exist instances \( I_1 \) and \( I_2 \) such that \( I_1 \models \varphi \), \( I_1 \nmodels \varphi_2 \), \( I_2 \models \varphi_2 \), \( I_2 \nmodels \varphi_1 \). From (1), we obtain \( I_1 \models \varphi \) and \( I_2 \models \varphi \). We next observe that \( I_1 \nmodels \varphi_1 \) and \( I_2 \nmodels \varphi_2 \) implies \( I_1 \uplus I_2 \nmodels \varphi_1 \) and \( I_1 \uplus I_2 \nmodels \varphi_2 \). Thus \( I_1 \uplus I_2 \nmodels \varphi \) by (1). Consequently, \( I_1 \) and \( I_2 \) witness that \( \varphi \) is not preserved under disjoint union.

We next characterize the equivalence of MMSN sentences to a generalized CSP and analyze the complexity of deciding this property.

For an MMSN sentence \( \varphi \), let \( \varphi_{\text{acyc}} \) be the MMSN sentence with the same quantifier prefix that contains all rules which have an acyclic body and can be obtained from a rule in \( \varphi \) by zero or more identifications of variables.

**Theorem 37.** An MMSN sentence \( \varphi \) is definable by a generalized CSP iff \( \varphi \equiv \varphi_{\text{acyc}} \).

**Proof.** "only if". Assume that \( \varphi \) is definable by a generalized CSP \( T = \{T_1, \ldots, T_n\} \). Using the construction of \( \varphi_{\text{acyc}} \), it can be verified that \( \varphi \subseteq \varphi_{\text{acyc}} \). It thus remains to be shown that \( \varphi_{\text{acyc}} \subseteq \varphi \). If the body of each rule in \( \varphi \) is acyclic, then this is clearly the case. Otherwise, let \( y \) be the maximum girth of a cyclic rule body from \( \varphi \). Take an \( S \)-instance \( I \) such that \( I \not\models \varphi \). We have to show that \( I \not\models \varphi_{\text{acyc}} \). Since \( \varphi \) is
equivalent to $T$, $I \not\equiv T_i$ for $1 \leq i \leq n$. From Lemma 19, we obtain an $S$-instance $I^g$ of girth exceeding $g$ such that $I^g \rightarrow I$ and $I^g \not\rightarrow T_i$ for $1 \leq i \leq n$. Thus $I^g \not\models \varphi$. As the girth of $I^g$ is higher than the girth of every cyclic rule body in $\varphi$, it follows that $I^g \not\models \varphi_{acyc}$. Since $I^g \rightarrow I$, $I \not\models \varphi_{acyc}$.

"if". Assume that $\varphi \equiv \varphi_{acyc}$. Since the rule bodies in $\varphi_{acyc}$ are acyclic, it is easy to convert $\varphi_{acyc}$ into an equivalent MMSNP sentence in which each rule body contains at most one atom that uses a predicate symbol from $S$; see [Feder and Vardi, 1998]. It is implicit in that paper (see also [Bienvenu et al., 2014]) that MMSNP sentences of this kind have the same expressive power as generalized CSPs. Thus, $\varphi$ is equivalent to a generalized CSP. 

Before showing the main complexity result of this section, we state a slightly refined version of a theorem from [Bourhis and Lutz, 2016] regarding the complexity of MMSNP containment. It emphasizes that the complexity of containment is double exponential only in the size of the rules, but not in their number. This only requires a careful analysis of the constructions in [Bourhis and Lutz, 2016].

**Theorem 38.** Containment between MMSNP sentences is in 2NE$\times$PTIME. More precisely, for MMSNP sentences $\varphi_1$ and $\varphi_2$ where $\varphi_i$ has $n_i$ rules and rule size $r_i$, $i \in \{1, 2\}$, it can be decided in time $2^{2^{O(r_1 + r_2 + \log n_1 + \log n_2)}}$.

**Theorem 15.** It is 2NE$\times$PTIME-complete to decide whether a given MMSNP-sentence is equivalent to a CSP.

**Proof.** We start with the upper bound. Lemma 35 and Theorem 37 suggest an algorithm for deciding CSP-definability of an MMSNP sentence $\varphi$: check whether $\varphi$ is preserved under disjoint union and $\varphi \equiv \varphi_{acyc}$. The first condition can be decided in 2NE$\times$PTIME according to Theorem 36. As for the second check, we note that the size of $\varphi_{acyc}$ might be exponential in the size of $\varphi$ so we cannot apply the MMSNP containment result from [Bourhis and Lutz, 2016] straightaway. However, the rule size of $\varphi_{acyc}$ is polynomial in the size of $\varphi$ and thus by Theorem 38 the second condition can be decided in 2NE$\times$PTIME as well.

For showing that CSP-definability of MMSNP sentences is 2NE$\times$PTIME-hard, we can apply the same reduction as in the proof of Theorem 36: the MMSNP sentences $\varphi_1$, $\varphi_2$, and $\varphi$, constructed in the reduction are such that $\varphi_2 \subseteq \varphi_1$ if $\varphi$ is equivalent to a CSP. 

---

Note that the lemma in its original formulation in [Feder and Vardi, 1998] applies to instances over schemas of any arity, not just to ABoxes.