From Conjunctive Queries to SPARQL Queries in Ontology-Mediated Querying

Cristina Feier[1], Carsten Lutz[1], and Frank Wolter[2]

[1] University of Bremen, Germany
feier@uni-bremen.de clu@uni-bremen.de
[2] University of Liverpool, United Kingdom
wolter@liverpool.ac.uk

Abstract. We consider the rewritability of ontology-mediated queries (OMQs) based on UCQs into OMQs based on (certain kinds of) SPARQL queries. Our focus is on $\mathcal{ALCI}$ as a paradigmatic expressive DL. For rewritability into SPARQL queries that are unions of basic graph patterns, we show that the existence of a rewriting is decidable, based on a suitable characterization; for unary OMQs, this coincides with rewritability into instance queries. For SPARQL queries that additionally admit projection, we make some interesting first observations. In particular, we show that whenever there is a rewriting, then there is one that uses the same TBox as the original OMQ and only $\mathcal{ALCI}$ concepts from a certain finite class of such concepts. We also observe that if the TBox of the original OMQ falls into Horn-$\mathcal{ALCI}$, then a rewriting always exists.

1 Introduction

We study ontology-mediated querying with expressive description logics (DLs), focussing on $\mathcal{ALCI}$ as a paradigmatic such DL. A variety of query languages has been considered in this context, including conjunctive queries (CQs), unions of conjunctive queries (UCQs), instance queries (IQs), and SPARQL queries. The former two are a particularly natural choice because they play a fundamental role in database theory and systems, and in fact they are used in a large number of theoretical studies on ontology-mediated querying [3, 4, 6]. On the practical side, however, there do not seem to be any systems that fully support CQs and UCQs whereas there are systems that support IQs and SPARQL queries, such as Hermit [8, 12]. This raises the question of when and how a (U)CQ can be expressed as an IQ or as a SPARQL query.

The relationship between unary (U)CQs and IQs, which can be seen as a simple form of SPARQL query, has been studied in [10,11,13], and very recently in [7]. While it is easy to see that every tree-shaped CQ can be translated into an equivalent IQ, it is more surprising that also some CQs that are not tree-shaped turn out to be IQ-rewritable when we have an expressive DL at our disposal [13]. 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 IQ $P \to \exists r.P(x)$ (equivalently $\neg P \sqcup \exists r.P(x)$). 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. Kikot and Zolin have identified a large class of CQs that are rewritable into IQs in the context of $\mathcal{ALCI}$, namely that class of unary CQs in which every cycle passes through the answer variable. This was further elaborated in [7] where it is shown that this condition precisely characterizes IQ-rewritability, the condition is generalized to the case of non-empty TBoxes and non-full ABox signatures, and tight complexity bounds for deciding whether an IQ-rewriting exists are obtained, between NP-complete in the case of the empty TBox and full ABox signature and 2NExpTime-complete in the general case.

The purpose of this paper is to present initial results on the rewritability of (U)CQs of arity greater than one, for which IQs are clearly not a suitable target. Kikot and Zolin replace IQ-rewritability by Turing reductions to knowledge base consistency [11]. We prefer to consider rewritability into (two fragments of) SPARQL, namely into unions of basic graph patterns (UBGP), that is, UCQs that contain no quantified variables but potentially have atoms $C(x)$ with C a compound concept, and into PUBGPs, which consist of a projection applied to a UBGP. From an implementation perspective, answering such queries is closely related to knowledge base consistency and other common reasoning tasks which has resulted in support by practical systems, while answering (U)CQs is not. In fact, an imporant difference is that the union in UBGPs and the projection in PUBGPs are operations on query answers while the disjunction of UCQs and the existential quantification of CQs are logical operators on the level of models which makes them more difficult to handle. Note that we disregard many parts of SPARQL such as difference, optional, and the binding of variables to concept and role names. Also, we stick to the OWL 2 direct semantics entailment regime.

After giving preliminaries, in Section 3 we summarize the results from [7] regarding IQ-rewritability in $\mathcal{ALCI}$. We also observe that IQs and unary UBGPs have the same expressive power, which is not entirely trivial, and that unary PUBGPs are more expressive. In Section 4, we then study UBGP-rewritability of OMQs from $(\mathcal{ALCI}, \text{UCQ})$, that is, the TBox is formulated in $\mathcal{ALCI}$, the actual query is a UCQ, and an ABox signature may be imposed. Remarkably, even if the UCQ is full (that is, it has no quantified variables), UBGP-rewritability is not guaranteed. We develop a characterization of UBGP-rewritability and use it to show that this problem is decidable.[3] The characterization is more technical than in the case of IQ-rewritability which we believe to be unavoidable. It also implies that, in UBGP-rewritings, it is always sufficient to use the same TBox as in the original OMQ.

For PUBGP-rewritability, our results are less complete. We start with observing that PUBGP-rewritings always exist when the TBox of the original OMQ is formulated in Horn-$\mathcal{ALCI}$, that otherwise rewritings are not guaranteed to exist and that, when they exist, they might necessarily involve BGPs whose number of

[3] Here and in the subsequent results, we assume that every CQ in the original UCQ is connected in the sense that every variable is reachable from an answer variable.

variables is exponential in the size of the TBox of the original OMQ. We then establish a (non-effective) characterization which shows that PUBGP-rewritability implies rewritability into a particular class of PUBGPs. It follows that it is never necessary to modify the TBox when constructing PUBGP-rewritings and that in atoms of the form $C(x)$, the concept C can be restricted to a certain finite class of concepts. This is a first step towards decidability of PUBGP-rewritability, which remains as an interesting open problem.

A long version of the paper containing full proofs in the appendix is available at http://www.informatik.uni-bremen.de/tdki/research/papers.html.

2 Preliminaries

We assume familiarity with standard DL notation and languages [2] and only introduce notions that are potentially ambiguous. The main DL studied in this paper is $\mathcal{ALCI}$. An *ABox* $\mathcal{A}$ is a finite set of assertions of the form $A(a)$ and $r(a, b)$, where A is a concept name, r a role name, and a, b are individual names. We use $\mathsf{ind}(\mathcal{A})$ to denote the set of all individual names that occur in $\mathcal{A}$. An interpretation is a *model* of an ABox $\mathcal{A}$ if it *satisfies* all assertions in $\mathcal{A}$, that is, $a \in A^{\mathcal{I}}$ when $A(a)$ is in $\mathcal{A}$ and $(a, b) \in r^{\mathcal{I}}$ when $r(a, b)$ is in $\mathcal{A}$. We thus make the standard name assumption. A *signature* Σ is a set of concept and role names. We use $\mathsf{sig}(\mathcal{A})$ to denote the set of concept and role names that occur in the ABox $\mathcal{A}$, and likewise for other syntactic objects such as TBoxes. An ABox $\mathcal{A}$ is a *Σ-ABox* if $\mathsf{sig}(\mathcal{A}) \subseteq \Sigma$.

An *instance query (IQ)* takes the form $C(x)$ where C is a concept from the DL under consideration and x a variable. In this paper, C will always be an $\mathcal{ALCI}$ concept. For an interpretation $\mathcal{I}$, we write $\mathcal{I} \models C(a)$ if $a \in C^{\mathcal{I}}$. A *conjunctive query (CQ)* is of the form $q(\mathbf{x}) = \exists \mathbf{y}\, \varphi(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are tuples of variables and $\varphi(\mathbf{x}, \mathbf{y})$ is a conjunction of *atoms* of the form $A(z)$, $r(z_1, z_2)$, or $z_1 = z_2$, with A a concept name, r a role name, and $z, z_1, z_2 \in \mathbf{x} \cup \mathbf{y}$. When $z_1 = z_2$ occurs in q we assume w.l.o.g. that $z_1, z_2 \in \mathbf{x}$ and there is no other atom in q which contains x_2. A *union of conjunctive queries (UCQ)* $q(\mathbf{x})$ is a formula of the form $\bigvee_i q_i(\mathbf{x})$, where each $q_i(\mathbf{x})$ is a CQ. When compound concepts are admitted in place of concept names in a CQ/UCQ, then we speak about an *extended* CQ/UCQ. For any kind of query, the free variables are called *answer variables*, the *arity* is the number of answer variables, and a query is *Boolean* if it has arity zero. A CQ is *full* if all variables are answer variables and a UCQ is *full* if every disjunct is.

A *homomorphism* from a CQ $q(\mathbf{x})$ to an interpretation $\mathcal{I}$ is a function $h : \mathbf{x} \cup \mathbf{y} \to \Delta^{\mathcal{I}}$ such that $h(z) \in A^{\mathcal{I}}$ for every atom $A(z)$ of $q(\mathbf{x})$, $(h(z_1), h(z_2)) \in r^{\mathcal{I}}$ for every atom $r(z_1, z_2)$ of $q(\mathbf{x})$, and $h(x_1) = h(x_2)$ for every atom $x_1 = x_2$ of $q(\mathbf{x})$. We write $\mathcal{I} \models q(\mathbf{a})$ and call $\mathbf{a}$ an *answer to* $q(\mathbf{x})$ *on* $\mathcal{I}$ if there is a homomorphism from $q(\mathbf{x})$ to $\mathcal{I}$ with $h(\mathbf{x}) = \mathbf{a}$. For a UCQ $\bigvee_i q_i(\mathbf{x})$ and an interpretation $\mathcal{I}$, we write $\mathcal{I} \models q(\mathbf{a})$ if $\mathcal{I} \models q_i(\mathbf{a})$ for some i. The semantics of extended CQ/UCQs is as expected.

We now introduce query languages related to SPARQL [9]. A *basic graph pattern (BGP)* is an extended full CQ. Note that IQs coincide with unary BGPs.

A *union of basic graph patterns (UBGP)* is of the form $\bigcup_i \varphi_i(\mathbf{x})$ where each φ_i is a BGP and a *PUBGP* is of the form $\Pi_{\mathbf{x}}(\bigcup_i \varphi_i(\mathbf{x}, \mathbf{y}_i))$ where each $\varphi_i(\mathbf{x}, \mathbf{y}_i)$ is a BGP. UBGPs correspond to PUBGPs in which all tuples $\mathbf{y}_i$ are empty as in this case the projection to $\mathbf{x}$ denoted by $\Pi_{\mathbf{x}}$ is vacuous. Note that BGPs and UBGPs must have non-zero arity while this is not the case for PUBGPs.

An *ontology-mediated query* (OMQ) takes the form $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ with $\mathcal{T}$ a TBox, $\Sigma \subseteq \mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$ an ABox signature, and $q(\mathbf{x})$ a query. The *arity* of Q is the arity of $q(\mathbf{x})$. When Σ is $\mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$, then for brevity we denote it with Σ_{full} and speak of the *full ABox signature*. Let $\mathcal{A}$ be a Σ-ABox. If $q(\mathbf{x})$ is an IQ or a (possibly extended) UCQ, then $\mathbf{a}$ is an *answer* to Q on $\mathcal{A}$ if $\mathcal{I} \models q(\mathbf{a})$ for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$. This also captures the case where $q(\mathbf{x})$ is a BGP. If $q(\mathbf{x}) = \Pi_{\mathbf{x}}(\bigcup_i \varphi_i(\mathbf{x}, \mathbf{y}_i))$ is a PUBGP, then $\mathbf{a}$ is an *answer* to Q on $\mathcal{A}$ if there exist i and $\mathbf{b}$ such that $\mathbf{ab}$ is an answer to $(\mathcal{T}, \Sigma, \varphi_i(\mathbf{x}, \mathbf{y}_i))$. This also captures the case where $q(\mathbf{x})$ is a UBGP. In either case, we write $\mathcal{A} \models Q(\mathbf{a})$ if $\mathbf{a}$ is an answer to Q on $\mathcal{A}$.

It is important to note and to keep in mind that, in OMQs, the union in (P)UBGPs behaves differently from the disjunction in UCQs and the projection in PUBGPs behaves differently from the existential quantification in UCQs (also note that the order is reversed). For example, let $\mathcal{T} = \{\exists r.\top \sqsubseteq A \sqcup B\}$ and $\mathcal{A}$ an $\{A, B\}$-ABox. Then the UCQ-based OMQ $(\mathcal{T}, \Sigma_{\mathsf{full}}, A(x) \vee B(x))$ returns every individual in the range of r as an answer whereas the UBGP-based OMQ $(\mathcal{T}, \Sigma_{\mathsf{full}}, A(x) \cup B(x))$ returns only those individuals a from the range of r such that $A(a) \in \mathcal{A}$ or $B(a) \in \mathcal{A}$. In fact, the latter query corresponds to executing the two OMQs $(\mathcal{T}, \Sigma_{\mathsf{full}}, A(x))$ and $(\mathcal{T}, \Sigma_{\mathsf{full}}, B(x))$ independently and then taking the union of their answer sets. Likewise, the projection of PUBGPs is a projection on answer sets.

We use $(\mathcal{L}, \mathcal{Q})$ to refer to the *OMQ language* in which the TBox is formulated in the DL $\mathcal{L}$ and where the actual queries are from the query language $\mathcal{Q}$, such as in $(\mathcal{ALCI}, \mathrm{UCQ})$.

Definition 1. *Let $(\mathcal{L}, \mathcal{Q})$ be an OMQ language. An OMQ $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ is* $(\mathcal{L}, \mathcal{Q})$-rewritable *if there is an OMQ Q' from $(\mathcal{L}, \mathcal{Q})$ such that the answers to Q and to Q' are identical on any Σ-ABox that is consistent with $\mathcal{T}$. In this case, we say that Q* is rewritable *into Q' and call Q' a* rewriting *of Q.*

We are interested in rewriting OMQs from $(\mathcal{ALCI}, \mathrm{UCQ})$ into OMQs based on IQs, UBGPs, and PUBGPs. For brevity, we speak of $\mathcal{Q}$-rewritability instead of $(\mathcal{ALCI}, \mathcal{Q})$-rewritability. For example, IQ-rewritability means rewritability of an OMQ from $(\mathcal{ALCI}, \mathrm{UCQ})$ into an OMQ from $(\mathcal{ALCI}, \mathrm{IQ})$ with the IQ formulated in $\mathcal{ALCI}$, and PUBGP-rewritability means rewritability into an OMQ from $(\mathcal{ALCI}, \mathrm{PUBGP})$ where the PUBGP uses only $\mathcal{ALCI}$ compound concepts. The following example of IQ-rewritability is from [7]. Examples and non-examples of PUBGP-rewritability are given later in this paper.

Example 1. Let $q_1(x) = r(x, x)$. The OMQ $Q_1 = (\emptyset, \{r\}, q_1(x))$ is rewritable into the OMQ $(\emptyset, \{r\}, C(x))$ where C is the $\mathcal{ALCI}$ concept $P \to \exists r.P$. In contrast, let $q_2(x) = \exists y\, r(x, y) \wedge r(y, y)$. It follows from Theorem 1 below that the OMQ $Q_2 = (\emptyset, \{r\}, q_2(x))$ is not rewritable into an OMQ from $(\mathcal{ALCI}, \mathrm{IQ})$.

We close this section with some additional, more technical preliminaries. We say that an OMQ $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ is *empty* if for all Σ-ABoxes $\mathcal{A}$, there is no answer to Q on $\mathcal{A}$. Let Q_1, Q_2 be OMQs, $Q_i = (\mathcal{T}_i, \Sigma, q_i(\mathbf{x}))$ for $i \in \{1, 2\}$. Then Q_1 is *contained* in Q_2, written $Q_1 \subseteq Q_2$, if for all Σ-ABoxes $\mathcal{A}$, every answer to Q_1 on $\mathcal{A}$ is also an answer to Q_2 on $\mathcal{A}$. Further, Q_1 and Q_2 are *equivalent*, written $Q_1 \equiv Q_2$, if $Q_1 \subseteq Q_2$ and $Q_2 \subseteq Q_1$.

Every CQ $q(\mathbf{x}) = \exists \mathbf{y}\, \varphi(\mathbf{x}, \mathbf{y})$ gives rise to an undirecteded graph G_q whose nodes are the elements of $\mathbf{x} \cup \mathbf{y}$ and that contains an edge $\{z_1, z_2\}$ if $\varphi(\mathbf{x}, \mathbf{y})$ contains an atom $r(z_1', z_2')$ or $z_1' = z_2'$ with $\{z_1, z_2\} = \{z_1', z_2'\}$. G_q might contain self loops. We say that $q(\mathbf{x})$ is *connected* if every variable is reachable from an answer variable in G_q. An UCQ is *connected* if every CQ in it is and an OMQ from $(\mathcal{ALCI}, \text{UCQ})$ is connected if the UCQ in it is. A *contraction* of a CQ $q(\mathbf{x})$ is a CQ obtained from $q(\mathbf{x})$ by zero or more variable identifications, where the identification of an answer variable x with any non-answer variable yields x.

3 IQ-rewritability

We give an overview of the results on IQ-rewritability in $(\mathcal{ALCI}, \text{UCQ})$ from [7]. In that paper, several other DLs are considered as well, including $\mathcal{ALC}$ and variations of $\mathcal{ALC}$ and $\mathcal{ALCI}$ that support role hierarchies, the universal role, and functional roles. In some cases, the characterization of OMQ rewritability and the complexity of deciding IQ-rewritability are rather different from the $\mathcal{ALCI}$ case, which we briefly comment on. We also show that unary OMQs from $(\mathcal{ALCI}, \text{IQ})$, $(\mathcal{ALCI}, \text{BGP})$, and $(\mathcal{ALCI}, \text{UBGP})$ have the same expressive power while unary OMQs from $(\mathcal{ALCI}, \text{PUBGP})$ are more expressive.

We start with a fundamental characterization of OMQs from $(\mathcal{ALCI}, \text{UCQ})$ that are IQ-rewritable, first giving some preliminaries. Let $q(x)$ be a unary CQ. We can assume w.l.o.g. that, being unary, $q(x)$ contains no equality atoms. 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 $q(x)$, $n \geq 1$, where[4]

1. $r_0, \ldots, r_{n-1}$ are (potentially inverse) roles,
2. $x_i \neq x_j$ for $0 \leq i < j < n$, and $x_0 = x_n$.

We say that $q(x)$ is *x-acyclic* if every cycle in it passes through x. A UCQ is *x-acyclic* if every CQ in it is.

Let $q(x)$ be a UCQ. We use $q_{\mathsf{acyc}}(x)$ to denote the UCQ that consists of all contractions of a CQ from $q(x)$ that are x-acyclic and $q_{\mathsf{acyc}}^{\mathsf{con}}(x)$ to denote the UCQ obtained from $q_{\mathsf{acyc}}(x)$ by restricting every CQ in it to atoms that only use variables reachable in G_q from the answer variable x. The following theorem is the announced characterization [7].

Theorem 1. *Let $Q = (\mathcal{T}, \Sigma, q(x))$ be a unary OMQ from $(\mathcal{ALCI}, UCQ)$ that is non-empty. Then the following are equivalent:*

[4] 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).

1. *Q is IQ-rewritable;*
2. *Q is rewritable into an OMQ $Q' = (\mathcal{T}, \Sigma, C(x))$ from $(\mathcal{ALCI}, IQ)$;*
3. *$Q \equiv (\mathcal{T}, \Sigma, q^{\mathsf{con}}_{\mathsf{acyc}}(x))$.*

Theorem 1 excludes empty OMQs, but these are trivially IQ-rewritable; moreover, emptiness of OMQs from $(\mathcal{ALCI}, \text{UCQ})$ is decidable (and 2ExpTime-complete) [1]. Equivalence of Points 1 and 2 implies that it is not necessary to modify the TBox when constructing an IQ-rewriting. The latter is not the case, for example, when constructing IQ-rewritings of OMQs from $(\mathcal{ALC}, \text{UCQ})$ assuming that the constructed IQ must be an $\mathcal{ALC}$ concept. In that case, Point 1 and 3 of Theorem 1 are still equivalent, but equivalence with Point 2 is not as it might be necessary to (mildly) extend the TBox [7].

In the direction "3 $\Rightarrow$ 2", we construct actual rewritings, extending a construction due to Kikot and Zolin [11] to accomodate TBoxes, ABox signatures, and UCQs (instead of CQs). This extension yields the following.

Lemma 1. *Let $Q = (\mathcal{T}, \Sigma, q(x))$ be an OMQ from $(\mathcal{ALCI}, UCQ)$. If $q(x)$ is x-acyclic and connected, then Q is rewritable into an OMQ $(\mathcal{T}, \Sigma, C(x))$ with $C(x)$ an IQ. The size of the IQ $C(x)$ is polynomial in the size of $q(x)$.*

The IQ $C(x)$ constructed in the proof of Lemma 1 takes the form $P \to (C_1 \sqcup \cdots \sqcup C_n)$ where P is a concept name or $\top$ and $C_1, \ldots, C_n$ are $\mathcal{ELI}$-concepts. Based on Theorem 1 and variations thereof and using careful reductions to OMQ containment [5], one can establish the following complexity results [7].

Theorem 2. *Deciding IQ-rewritability of a unary OMQ from $(\mathcal{ALCI}, UCQ)$ is*

1. *2*NExpTime*-complete in the general case;*
2. *2*ExpTime*-complete for OMQs based on the full ABox signature;*
3. NP*-complete for OMQs based on the empty TBox.*

Different complexities are obtained for other DLs. For example, IQ-rewritability is undecidable in $(\mathcal{ALCF}, \text{CQ})$ and between ExpTime and coNExpTime in $(\mathcal{ALC}, \text{UCQ})$ when the ABox signature is full. As observed in [7], IQ-rewritings for OMQs based on the empty TBox can be viewed as underapproximations for OMQs with non-empty TBoxes in the sense that an IQ-rewriting for $(\emptyset, \Sigma, q(x))$ with $q(x)$ a UCQ is also an IQ-rewriting for $(\mathcal{T}, \Sigma, q(x))$ for any $\mathcal{ALCI}$ TBox $\mathcal{T}$.

We now compare OMQs based on IQs to unary OMQs based on BGPs, UBGPs, and PUBGPs. It is obvious from the definitions that OMQs from $(\mathcal{ALCI}, \text{IQ})$ and $(\mathcal{ALCI}, \text{BGP})$ have the same expressive power. The following is less trivial to show than one might think as converting an OMQ from $(\mathcal{ALCI}, \text{UBGP})$ into an OMQ from $(\mathcal{ALCI}, \text{IQ})$ requires a careful modification of the TBox to bridge the semantic gap between unions in UBGPs and disjunction in IQs. A proof is in the long version of the paper.

Theorem 3. *The language $(\mathcal{ALCI}, IQ)$ has the same expressive power as unary $(\mathcal{ALCI}, UBGP)$.*

The next example shows that unary $(\mathcal{ALCI}, \text{PUBGP})$ is more expressive than $(\mathcal{ALCI}, \text{IQ})$.

Example 2. Let $Q = (\emptyset, \{r\}, q(x))$ be the OMQ where $q(x)$ is the PUBGP $\Pi_x \varphi(x, y)$ and $\varphi(x, y)$ is the BGP $r(x, y) \wedge (P \to \exists r.P)(y)$. It can be verified that Q is equivalent to the OMQ Q_2 from Example 1, which is not expressible in $(\mathcal{ALCI}, \mathrm{IQ})$.

4 UBGP-Rewritability

We study UBGP-rewritability in $(\mathcal{ALCI}, \mathrm{UCQ})$. We first present an example and a characterization of UBGP-rewritability, and then use it to show that this property is decidable. The example illustrates that as a result of disjunction in UCQs behaving differently from union in UBGPs, OMQs based on full UCQs are not always UBGP-rewritable. Note that this is in contrast to the CQ case since every OMQ based on a full CQ is by definition also an OMQ based on a BGP.

Example 3. Take the UCQ

$$q(x, y) = (A(x) \wedge r(x, y)) \vee (r(x, y) \wedge A(y)).$$

We first consider the OMQ $Q_0 = (\emptyset, \Sigma_{\mathsf{full}}, q(x, y))$ based on the empty TBox. Clearly, Q_0 is rewritable into the OMQ $Q_0' = (\emptyset, \Sigma_{\mathsf{full}}, q'(x, y))$ from the language $(\mathcal{ALCI}, \mathrm{UBGP})$ in which the disjunction from the UCQ $q(x, y)$ is replaced by BGP union:

$$q'(x, y) = (A(x) \wedge r(x, y)) \cup (r(x, y) \wedge A(y)).$$

Now consider the TBox $\mathcal{T} = \{M \sqsubseteq \forall s.A \sqcup \forall t.A\}$ and let $Q_1 = (\mathcal{T}, \Sigma_{\mathsf{full}}, q(x, y))$. Then $Q_1' = (\mathcal{T}, \Sigma_{\mathsf{full}}, q'(x, y))$ is not a rewriting of Q_1. To see this, consider the ABox

$$\mathcal{A} = \{r(a, b), M(c), s(c, a), t(c, b)\}.$$

Then $\mathcal{A} \models Q_1(a, b)$, but $\mathcal{A} \not\models Q_1'(a, b)$. In fact, Q_1 is not UBGP-rewritable. We sketch the proof. Assume to the contrary that there is a rewriting $Q = (\mathcal{T}', \Sigma, \bigcup_i \varphi_i(x, y))$ of Q_1 from $(\mathcal{ALCI}, \mathrm{UBGP})$. Then $\mathcal{A} \models Q(a, b)$. Consider the ABox $\mathcal{A}'$ obtained from $\mathcal{A}$ by dropping all assertions that use c, taking the union with two copies $\mathcal{A}_a$ and $\mathcal{A}_b$ of $\mathcal{A}$ that do not share individuals with $\mathcal{A}$ or with each other, and then identifying the copy of a in $\mathcal{A}_a$ with a, and the copy of b in $\mathcal{A}_b$ with b. Thus,

$$\mathcal{A}' = \{r(a, b), M(c_1), s(c_1, a), t(c_1, b'), M(c_2), s(c_2, a'), s(c_2, b), t(a', b)\}.$$

We then have $\mathcal{A}' \not\models Q_1(a, b)$, and so $\mathcal{A}' \not\models Q(a, b)$. By construction, there is a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$ that maps a to a and also a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$ that maps b to b. As entailment of $\mathcal{ALCI}$ concepts is preserved under ABox homomorphisms, for every $\mathcal{ALCI}$ concept C, $\mathcal{A} \models (\mathcal{T}', \Sigma_{\mathsf{full}}, C(x))(a)$ implies $\mathcal{A}' \models (\mathcal{T}', \Sigma_{\mathsf{full}}, C(x))(a)$ and the same holds for b. We obtain $\mathcal{A}' \models Q(a, b)$ from $\mathcal{A} \models Q(a, b)$ and have derived a contradiction.

We now give the characterization of UBGP-rewritability, starting with some technical preliminaries. We first identify a set of concepts to be used in UBGP-rewritings and, later on, also in PUBGP-rewritings.

Let $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ be a non-Boolean OMQ from $(\mathcal{ALCI}, \text{UCQ})$. Let $\mathbb{Q}_{q(\mathbf{x})}$ be the set of unary CQs $p(x)$ that are connected and x-acyclic and can be obtained from a contraction of a CQ in $q(\mathbf{x})$ by dropping all equality atoms, then (potentially) taking a subquery, and finally choosing any variable x as the new answer variable and existentially quantifying all other variables. For every such $p(x)$, take an $\mathcal{ALCI}$ concept C_p such that the CQ-based OMQ $(\mathcal{T}, \Sigma, p(x))$ is equivalent to the IQ-based OMQ $(\mathcal{T}, \Sigma, C_p(x))$, as constructed in the proof of Lemma 1. Let $\mathsf{sub}(Q)$ be the closure under subconcepts of concepts C_p, $p(x) \in \mathbb{Q}_{q(\mathbf{x})}$, and concepts used in $\mathcal{T}$. A *Q-type* is a minimal set of concepts that contains C or $\neg C$ for every $C \in \mathsf{sub}(Q)$. For a Q-type τ, set $C_\tau = \bigsqcap_{C \in \tau} C$. Also let $\mathbb{C}_Q$ be the set of concepts of the form $C_{\tau_1} \sqcup \cdots \sqcup C_{\tau_\ell}$ with each τ_i a Q-type.

We next define an approximation of Q from below that is formulated in $(\mathcal{ALCI}, \text{UBGP})$ and based on the TBox $\mathcal{T}$ from Q and the concepts from $\mathbb{C}_Q$. More precisely, Q_{UBGP} is the OMQ $(\mathcal{T}, \Sigma, q'(\mathbf{x}))$ where $q'(\mathbf{x})$ is the UBGP that is the union of all BGPs $\varphi(\mathbf{x})$ such that

1. for every $x \in \mathbf{x}$, there is a unique atom $D(x) \in \varphi(\mathbf{x})$, and $D \in \mathbb{C}_Q$ and
2. $(\mathcal{T}, \Sigma, \varphi(\mathbf{x})) \subseteq Q$.

It is easy to see that there are only finitely many such BGPs. Note that, by construction, $Q_{\text{UBGP}} \subseteq Q$.

Theorem 4. *Let $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ be an OMQ from $(\mathcal{ALCI}, UCQ)$ that is connected, of arity at least one, and non-empty. Then the following are equivalent:*

1. *Q is UBGP-rewritable;*
2. *Q_{UBGP} is a rewriting of Q;*
3. *$Q \subseteq Q_{UBGP}$.*

Since Q_{UBGP} uses the same TBox as Q, Theorem 4 implies that it is not necessary to modify the TBox when constructing UBGP-rewritings. The proof of Theorem 4 is a minor variation of the proof of Theorem 7 below, given after Lemma 2.

The next theorem can be obtained by a reduction of UBGP-rewritability to containment in monadic disjunctive datalog [5], based on Theorem 4.

Theorem 5. *UBGP-rewritability in $(\mathcal{ALCI}, UCQ)$ is decidable.*

A 2NExpTime lower bound follows from Theorems 2 and 3. The described reduction only yields a 4NExpTime upper bound.

5 PUBGP-Rewritability

We study PUBGP-rewritability in $(\mathcal{ALCI}, \text{UCQ})$, first observing that these always exist when the TBox of the original OMQ is formulated in the fragment Horn-$\mathcal{ALCI}$ of $\mathcal{ALCI}$. This is essentially a consequence of the fact that Horn-$\mathcal{ALCI}$ has universal models, more details are given in the long version.

Theorem 6. *Every OMQ from (Horn-$\mathcal{ALCI}$, UCQ) is PUBGP-rewritable.*

We next give an example showing that PUBGP-rewritings are not guaranteed to exist in the non-Horn case. This is already true for OMQs from $(\mathcal{ALC}, \mathrm{CQ})$ even when $\mathcal{ALCI}$ concepts are admitted in PUBGPs.

Example 4. Let $Q = (\mathcal{T}, \Sigma_{\mathsf{full}}, q(x))$ with

$$\mathcal{T} = \{M \sqsubseteq M_1 \sqcup M_2, M_1 \sqsubseteq \forall s.A, M_2 \sqsubseteq \forall t.A, A \sqsubseteq \forall t.A\}$$

and $q(x) = \exists y\ r(x,y) \wedge A(y) \wedge r(y,y)$. Then Q is not PUBGP-rewritable. For a proof by contradiction, assume that $Q' = (\mathcal{T}', \Sigma_{\mathsf{full}}, q'(x))$ is a PUBGP-rewriting. For every $n \geq 1$, let

$$\begin{aligned}\mathcal{A}_n = \{&r(a,b_1), r(a,b_2), r(b_1,b_1), r(b_2,b_2),\\ &M(c_1), s(c_1,b_1), t(c_1,c_2), \ldots, t(c_n,b_2)\}.\end{aligned}$$

It can be verified that $\mathcal{A}_n \models Q(a)$ and thus $\mathcal{A}_n \models Q'(a)$, for every n. Choose $n > 0$ that exceeds the number of variables in any BGP in q'. As $\mathcal{A}_n \models Q'(a)$, there must be a BGP $\varphi(x, \mathbf{y})$ in q' and a $\mathbf{b}$ such that $\mathcal{A}_n \models Q_\varphi(a, \mathbf{b})$ for the OMQ $Q_\varphi = (\mathcal{T}, \Sigma_{\mathsf{full}}, \varphi(x, \mathbf{y}))$. Further, there must be a c_i in $\mathcal{A}_n$ that is not in $\mathbf{b}$. Let $\mathsf{ind}^- = \mathsf{ind}(\mathcal{A}_n) \setminus \{c_i\}$ and for each $b \in \mathsf{ind}^-$, let $\mathcal{A}_n^b$ denote the disjoint copy of $\mathcal{A}_n$ obtained by renaming every individual c to c^b. Now consider the ABox $\mathcal{A}_n'$ obtained from $\mathcal{A}_n$ by removing all assertions that use c_i, taking the disjoint union with all $\mathcal{A}_n^b$, $b \in \mathsf{ind}^-$, and then identifying each individual $b \in \mathsf{ind}^-$ with the individual b^b from $\mathcal{A}_n^b$. Note that this construction is similar to the construction of $\mathcal{A}'$ in Example 3. We have $\mathcal{A}_n' \models Q_\varphi(a, \mathbf{b})$ since for every $b \in \mathsf{ind}^-$, there is a homomorphism h from $\mathcal{A}_n$ to $\mathcal{A}_n'$ with $h(b) = b$ and for all $b, b' \in \mathsf{ind}^-$ and role names v, $v(b,b') \in \mathcal{A}_n$ iff $v(b,b') \in \mathcal{A}_n'$. On the other hand, it is easy to see that $\mathcal{A}_n' \not\models Q(a)$ due to the removal of c_i.

Next example shows that when constructing PUBGP-rewritings of OMQs from $(\mathcal{ALC}, \mathrm{CQ})$, it can be necessary to use BGPs whose topology is different from that of the CQs in the original OMQ. In fact, exponentially many variables (in the size of the TBox of the original OMQ) can be required in BGPs. We conjecture that the example can even be improved to show a double exponential lower bound on the number of variables. As for the previous example, the same is true for $(\mathcal{ALCI}, \mathrm{CQ})$, that is, when $\mathcal{ALCI}$ concepts are admitted in PUBGPs.

Example 5. Let $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ be the OMQ with

$$\begin{aligned}\mathcal{T} &= \{A_i \sqsubseteq \forall r.A_{i+1} \sqcup \forall s.A_{i+1} \mid 0 \leq i < n\} \cup \{A_n \sqsubseteq A\}\\ \Sigma &= \{A_0, r, s, t\}\\ q(x) &= \exists y\, t(x,y) \wedge A(y) \wedge t(y,y).\end{aligned}$$

Q is rewritable into the OMQ $Q' = (\emptyset, \Sigma, q'(x))$ where $q'(x)$ is the PUBGP $\Pi_x \varphi(x, \mathbf{y})$ with $\varphi(x, \mathbf{y})$ as depicted in Figure 5. Observe that the topmost part is a full binary tree of depth n. We aim to show that there is no rewriting of Q that uses less than 2^n variables.

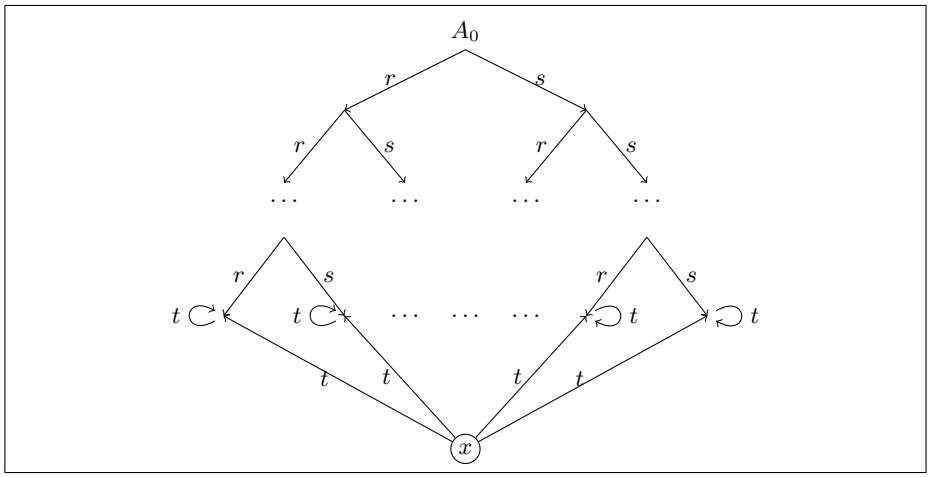

Fig. 1. PUBGP-rewriting for the OMQ Q in Example 5

Assume for a proof by contradiction that there is such a rewriting $Q' = (\mathcal{T}', \Sigma, q'(x))$. Let $\mathcal{A}_\varphi$ be φ viewed as an ABox. Clearly, $\mathcal{A}_\varphi \models Q(x)$ and thus there must be a BGP $\varphi'(x, \mathbf{y}')$ in q' and a homomorphism h such that $\mathcal{A}_\varphi \models Q_\varphi(x, h(\mathbf{y}'))$ where $Q_\varphi = (\mathcal{T}', \Sigma, \varphi'(x, \mathbf{y}'))$. Some leaf node y from the binary tree in φ must be outside the range of h. Now consider the ABox $\mathcal{A}'_\varphi$ obtained from $\mathcal{A}_\varphi$ by dropping all assertions that use y, adding a disjoint copy $\mathcal{A}^z_\varphi$ of $\mathcal{A}_\varphi$ for every $z \in \mathsf{ind}(\mathcal{A}_\varphi)$ except y, and then identifying each z from $\mathcal{A}_\varphi$ with the copy of z in $\mathcal{A}^z_\varphi$. Note that this construction is similar to the construction of $\mathcal{A}'$ in Example 3. Then, as in Example 3, $\mathcal{A}'_\varphi \models Q'(x, h(\mathbf{y}'))$ follows from the fact that $\varphi'(x, \mathbf{y}')$ is a BGP in q'. On the other hand, $\mathcal{A}'_\varphi \not\models Q(x)$ and we have derived a contradiction.

We now show that PUBGP-rewritability implies rewritability into a more controlled class of PUBGPs, based on the set of concepts $\mathbb{C}_Q$ defined in Section 4. A $\mathbb{C}_Q$-*PUBGP* is a PUBGP that uses only concepts from $\mathbb{C}_Q$.

Theorem 7. *Let $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ be an OMQ from $(\mathcal{ALCI}, UCQ)$ that is connected, of arity at least one, and non-empty. Then the following are equivalent:*

1. *Q is PUBGP-rewritable;*
2. *Q is rewritable into an OMQ $Q' = (\mathcal{T}, \Sigma, q'(\mathbf{x}))$ with $q'(\mathbf{x})$ a $\mathbb{C}_Q$-PUBGP.*

Theorem 7 clearly implies that it is not necessary to modify the TBox when constructing PUBGP-rewritings. It also implies that it is not necessary to use concepts from outside $\mathbb{C}_Q$. Unlike Theorem 4, however, it does not immediately give rise to a decision procedure for PUBGP-rewritability. Note that an approximation Q_{PUBGP} of Q from below, formulated in $(\mathcal{ALCI}, \text{PUBGP})$ and constructed in exact analogy with the query Q_{UBGP} from Theorem 4, is not guaranteed to be finite. In fact, the main challenge in showing decidability of PUBGP-rewritability is to give a bound on the number of variables used in PUBGP-rewritings. An

additional complication is that there are no existing approaches for deciding containment or equivalence between an OMQ from $(\mathcal{ALCI}, \text{UCQ})$ and an OMQ from $(\mathcal{ALCI}, \text{PUBGP})$. Regarding the converse direction, it is possible to prove that every OMQ from $(\mathcal{ALCI}, \text{PUBGP})$ can be rewritten into $(\mathcal{ALCI}, \text{UCQ})$, the construction being similar to that used in the proof of Theorem 3.

To prove Theorems 7 and 4, we introduce a certain normalization of PUBGP-rewritings. Let $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ be an OMQ from $(\mathcal{ALCI}, \text{UCQ})$ and let $Q_R = (\mathcal{T}_R, \Sigma, q_R(\mathbf{x}))$ be a PUBGP-rewriting of Q with $q_R(\mathbf{x}) = \Pi_{\mathbf{x}}(\bigcup_i \varphi_i(\mathbf{x}, \mathbf{y}_i))$. We may assume w.l.o.g. that the set of BGPs $\varphi_i(\mathbf{x}, \mathbf{y}_i)$ in $q_R(\mathbf{x})$ is closed under contraction.[5] From each BGP $\varphi_i(\mathbf{x}, \mathbf{y}_i)$, we construct all BGPs $\varphi_{i,j}(\mathbf{x}, \mathbf{y}_i)$ that satisfy the following conditions:

1. $r(z_1, z_2) \in \varphi_{i,j}$ iff $r(z_1, z_2) \in \varphi_i$;
2. $z_1 = z_2 \in \varphi_{i,j}$ iff $z_1 = z_2 \in \varphi_i$;
3. for each $z \in \mathbf{x}\mathbf{y}_i$, there is a unique atom $C(z) \in \varphi_{i,j}$, and $C \in \mathbb{C}_Q$;
4. $(\mathcal{T}, \Sigma, \Pi_{\mathbf{x}}\varphi_{i,j}(\mathbf{x}, \mathbf{y}_i)) \subseteq Q$.

We call the OMQ $Q'_R = (\mathcal{T}, \Sigma, \Pi_{\mathbf{x}}(\bigcup_{i,j} \varphi_{i,j}(\mathbf{x}, \mathbf{y}_i)))$ from $(\mathcal{ALCI}, \text{PUBGP})$ the *normalization* of Q_R. In the long version of the paper, we show the following, which clearly yields Theorem 7.

Lemma 2. *If Q_R is a PUBGP-rewriting of Q, then so is its normalization Q'_R.*

We note that Lemma 2 also gives Theorem 4. In fact, assume that an OMQ $Q = (\mathcal{T}, \Sigma, q(\mathbf{x}))$ from $(\mathcal{ALCI}, \text{UCQ})$ is rewritable into an OMQ $Q_R = (\mathcal{T}', \Sigma, q'(\mathbf{x}))$ from $(\mathcal{ALCI}, \text{UBGP})$, a subclass of $(\mathcal{ALCI}, \text{PUBGP})$. Then, the rewriting Q'_R constructed above is also from $(\mathcal{ALCI}, \text{UBGP})$. Recall the OMQ Q_{UBGP} constructed before Theorem 4. By construction, $Q_{\text{UBGP}} \subseteq Q$ and Q_{UBGP} contains every BGP from Q_R. Thus, we also have $Q \subseteq Q_{\text{UBGP}}$, as required.

6 Conclusion

The main problem left open in this paper is whether PUBGP-rewritability is decidable in $(\mathcal{ALCI}, \text{UCQ})$ and related OMQ languages. As a precursor, it would be interesting to show that equivalence of an OMQ from $(\mathcal{ALCI}, \text{UCQ})$ and an OMQ from $(\mathcal{ALCI}, \text{PUBGP})$ is decidable. Another interesting question is whether rewritability into PUBGPs is actually different from rewritability into (more) complete SPARQL, that is, whether features such as difference make it possible to rewrite additional queries. Finally, it would be interesting to investigate the consequence of admitting constants in the original query and nominals in the rewriting.

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.

[5] Meaning that we treat $\varphi_i(\mathbf{x}, \mathbf{y}_i)$ as a CQ with answer variables $\mathbf{x}$.

References

1. 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.
2. Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
3. 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.
4. 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.
5. 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.
6. 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.
7. Cristina Feier, Carsten Lutz, and Frank Wolter. From conjunctive queries to instance queries in ontology-mediated querying. In *Proc. of IJCAI*, pages 1810–1816. ijcai.org, 2018.
8. 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.
9. Birte Glimm and Chimezie Ogbuji, editors. *SPARQL 1.1 Entailment Regimes*. W3C Recommendation, 21 March 2013. Available at https://www.w3.org/TR/sparql11-entailment/.
10. 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.
11. 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.
12. Ilianna Kollia, Birte Glimm, and Ian Horrocks. SPARQL query answering over OWL ontologies. In *Proc. of ESWC*, volume 6643 of *LNCS*, pages 382–396. Springer, 2011.
13. 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.