Query Expressibility and Verification in Ontology-Based Data Access

Carsten Lutz, Johannes Marti, Leif Sabellek
Department of Computer Science, University of Bremen, Germany

Abstract

In ontology-based data access, multiple data sources are integrated using an ontology and mappings. In practice, this is often achieved by a bootstrapping process, that is, the ontology and mappings are first designed to support only the most important queries over the sources and then gradually extended to enable additional queries. In this paper, we study two reasoning problems that support such an approach. The expressibility problem asks whether a given source query q_s is expressible as a target query (that is, over the ontology's vocabulary) and the verification problem asks, additionally given a candidate target query q_t, whether q_t expresses q_s. We consider (U)CQs as source and target queries and GAV mappings, showing that both problems are Π_2^p-complete in DL-Lite, coNEXPTIME-complete between $\mathcal{EL}$ and $\mathcal{ELHI}$ when source queries are rooted, and 2EXPTIME-complete for unrestricted source queries.

Introduction

Ontology-based data access (OBDA) (Poggi et al. 2008) is an instantiation of the classical data integration scenario, that is, a set of data sources is translated into a unifying global schema by means of mappings. The distinguishing feature of OBDA is that the global schema is formulated in terms of an ontology which provides a rich domain model and can be used to derive additional query answers via logical reasoning. When data sources are numerous such as in large enterprises, data integration is often a considerable investment. OBDA is no exception since the construction of both the mappings and the ontology is non-trivial and labour intensive.

In practice, OBDA is thus often approached in an incremental manner (Trisolini, Lenzerini, and Nardi 1999; Kharlamov et al. 2015; Sequeda and Miranker 2017). One starts with a small set of important source queries (typically hand crafted by experts from the enterprise's IT department) and builds mappings for the involved sources and an initial ontology that support these queries, manually or with the help of extraction tools (Jiménez-Ruiz et al. 2015; Pinkel et al. 2018). The outcome of this first step is then evaluated and, when considered successful, ontology and mappings are extended to support additional queries. This process may proceed for several rounds and in fact forever since new data sources and queries tend to appear as the enterprise develops and existing data sources or the ontology need to be updated (Lembo et al. 2017).

The aim of this paper is to study two reasoning tasks that support such an incremental approach to OBDA. The *expressibility problem* asks whether a given source query q_s is expressible as a target query q_t over the global schema defined by the ontology. Possible reasons for non-expressibility include that the mappings do not transport all data required for answering q_s to the global schema and that the ontology 'blurs' the distinction between different relations from the sources, examples are given in the paper. If q_s is not expressible, one might thus decide to add more mappings or to rework the ontology. The verification problem asks, additionally given a candidate target query q_t, whether q_t expresses q_s. This is useful for example when a complex q_t has been manually constructed and when the ontology, mappings, or source schemas have been updated, with an unclear impact on q_t. The same problems have been considered in the context of open data publishing, there called finding and recognition of s-to-t rewritings (Cima 2017).

We consider UCQs (and sometimes CQs) both for source and target queries, global as view (GAV) mappings, and ontologies that are formulated in DL-Lite or in a description logic (DL) between $\mathcal{EL}$ and $\mathcal{ELHI}$, which are all very common choices in OBDA. It follows from results in (Nash, Segoufin, and Vianu 2010; Afrati 2011) that, even without ontologies, additional source UCQs become expressible when full first-order logic (FO) is admitted for the target query rather than only UCQs. In OBDA, however, going beyond UCQs quickly results in undecidability of query answering (Baader et al. 2017) and thus we stick with UCQs.

The expressibility problem in OBDA is closely related to the problem of query expressibility over views, which has been intensively studied in database theory, see for example (Levy et al. 1995; Duschka and Genesereth 1997; Calvanese et al. 2002; Nash, Segoufin, and Vianu 2010; Afrati 2011) and references therein. The problem has occasionally also been considered in a DL context (Calvanese, De Giacomo, and Lenzerini 2000; Haase and Motik 2005; Beeri, Levy, and Rousset 1997; Calvanese et al. 2012). These papers, however, study setups different from the one we consider, both regarding the rôle of the ontology and the description logics used.

In many classical cases of query expressibility over views, informally stated, q_s is expressible over a set of mappings $\mathbf{M}$ (representing views) if and only if the UCQ $\mathbf{M}^-(\mathbf{M}(q_s))$ is contained in q_s where $\mathbf{M}(q_s)$ is the UCQ obtained from q_s by applying the mappings and $\mathbf{M}^-(\mathbf{M}(q_s))$ is obtained from $\mathbf{M}(q_s)$ by applying the mappings backwards (Nash, Segoufin, and Vianu 2010; Afrati 2011). Our starting point for proving decidability and upper complexity bounds for expressibility in OBDA is the observation that we need to check whether $\mathbf{M}^-(q_r)$ is contained in q_s where q_r is a (potentially infinitary) UCQ-rewriting of the UCQ $\mathbf{M}(q_s)$ under the ontology; note that, here, we mean *rewriting* of an ontology-mediated query into a source query in the classical sense of ontology-mediated querying, see for example (Bienvenu et al. 2016). Verification can be characterized in a very similar way. These characterizations also show that expressibility can be reduced to verification in polynomial time and that if q_s is expressible, then it is expressed by the polynomial size UCQ $\mathbf{M}(q_s)$.

Our main results are that within the setup described above, expressibility and verification are Π^p_2-complete in DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$ and in many other dialects of DL-Lite, CONEXPTIME-complete in DLs between $\mathcal{EL}$ and $\mathcal{ELHI}$ when the source UCQ is rooted (that is, every variable is reachable from an answer variable in the query graph of every CQ), and 2EXPTIME-complete in the unrestricted case. There are some surprises here. First, the Π^p_2 lower bound already applies when the ontology is empty and the source query is a CQ which means that, in the database theory setting, it is Π^p_2-hard to decide the fundamental problem whether a source CQ is expressible as a (U)CQ over a set of UCQ views. For this problem, an NP upper bound was claimed without proof in (Levy et al. 1995). Our results show that the problem is actually Π^p_2-complete. A second surprise is that 2EXPTIME-respectively CONEXPTIME-hardness applies already in the case that the ontology is formulated in $\mathcal{EL}$ (and when queries are UCQs). We are not aware of any other reasoning problem for $\mathcal{EL}$ that has such a high complexity whereas there are several such problems known for $\mathcal{ELI}$, that is, $\mathcal{EL}$ extended with inverse roles (Bienvenu et al. 2016). There is a clear explanation, though: the mappings make it possible to introduce just enough inverse roles in the backwards translation $\mathbf{M}^-$ mentioned above so that hardness proofs can be made work.

Detailed proofs are deferred to the appendix.

Preliminaries

We use a mix of standard DL notation (Baader et al. 2017) and standard notation from database theory.

Databases and Queries. A *schema* $\mathbf{S}$ is a set of relation names with associated arities. An $\mathbf{S}$-*database* D is a set of *facts* $R(a_1,\dots,a_n)$ where $R \in \mathbf{S}$ is a relation name of arity n and $a_1,\dots,a_n$ are constants. We use $\mathsf{adom}(D)$ to denote the set of constants that occur in D.

A conjunctive query (CQ) $q(\mathbf{x})$ *over schema* $\mathbf{S}$ takes the form $\exists \mathbf{y}\, \varphi(\mathbf{x},\mathbf{y})$, where $\mathbf{x}$ are the *answer variables*, $\mathbf{y}$ are the *quantified variables*, and φ is a conjunction of *relational atoms* $R(z_1,\dots,z_n)$ and *equality atoms* $z_1 = z_2$ where $R \in \mathbf{S}$ is of arity n and $z_1,\dots,z_n$ are variables from $\mathbf{x} \cup \mathbf{y}$. Contrary to the usual setup and to avoid dealing with special cases in some technical constructions, we do *not* require that all variables in $\mathbf{x}$ actually occur in $\varphi(\mathbf{x},\mathbf{y})$. We sometimes confuse q with the set of atoms in φ, writing for example $R(x,y,z) \in q$. We use $\mathsf{var}(q)$ to denote $\mathbf{x} \cup \mathbf{y}$. The *arity* of a CQ is the number of variables in $\mathbf{x}$ and q is *Boolean* if it has arity zero. A *homomorphism from* q *to a database* D is a function $h : \mathsf{var}(q) \to \mathsf{adom}(D)$ such that $R(h(\mathbf{x})) \in D$ for every relational atom $R(\mathbf{x}) \in q$ and $h(x) = h(y)$ for every relational atom $x = y \in q$. Note that h needs to be defined also for answer variables that do not occur in $\varphi(\mathbf{x},\mathbf{y})$. A tuple $\mathbf{a} \in \mathsf{adom}(D)$ is an *answer* to q on D if there is a homomorphism h from q to D with $h(\mathbf{x}) = \mathbf{a}$. A *union of conjunctive queries (UCQ)* is a disjunction of CQs that all have the same answer variables. Answers to UCQs are defined in the expected way. We use $\mathsf{ans}_q(D)$ to denote the set of all answers to UCQ q on database D.

Let $q_1(\mathbf{x}_1), q_2(\mathbf{x}_2)$ be UCQs of the same arity and over the same schema $\mathbf{S}$. We say that q_1 is *contained* in q_2, denoted $q_1 \subseteq_{\mathbf{S}} q_2$, if for every $\mathbf{S}$-database D, $\mathsf{ans}_{q_1}(D) \subseteq \mathsf{ans}_{q_2}(D)$. It is well-known that, when q_1, q_2 are CQs, then $q_1 \subseteq_{\mathbf{S}} q_2$ iff there is a homomorphism from q_2 to q_1, that is, a function $h : \mathsf{var}(q_2) \to \mathsf{var}(q_1)$ such that $R(h(\mathbf{x})) \in q_1$ for every relational atom $R(\mathbf{x}) \in q_2$, $(h(x), h(y))$ is in the equivalence relation generated by the equality atoms in q_1 for every $x = y \in q_2$, and $h(\mathbf{x}_2) = \mathbf{x}_1$. We indicate the existence of such a homomorphism with $q_2 \to q_1$. When q_i is a UCQ with disjuncts $q_{i,1},\dots,q_{i,k_i}$, $i \in \{1,2\}$, then $q_1 \subseteq_{\mathbf{S}} q_2$ iff for every $q_{1,i}$, there is a $q_{2,j}$ with $q_{2,j} \to q_{1,i}$.

We shall frequently view CQs as databases, for which we merely need to read variables as constants of the same name and drop equality atoms. Conversely, we shall also view a tuple $(D, \mathbf{a})$ with D a database and $\mathbf{a} = a_1 \cdots a_n \in \mathsf{adom}(D)$ as an n-ary CQ; note that repeated elements are admitted in $\mathbf{a}$. We do this by reserving n fresh answer variables $x_1,\dots,x_n$, viewing D as a CQ by reading all constants (including those in $\mathbf{a}$) as quantified variables, and adding the equality atom $x_i = a_i$ for $1 \leq i \leq n$.

Ontology-Based Data Access. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite set of *concept names* and *role names*. An $\mathcal{ELI}$-*concept* is formed according to the syntax rule

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid \exists r^-.C$$

where A ranges over *concept name* and r over *role names*. An expression r^- is called an *inverse role* and a *role* is either a role name or an inverse role. As usual, we let $(r^-)^-$ denote r. An $\mathcal{ELHI}$-*ontology* is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$, C, D $\mathcal{ELI}$-concepts, and *role inclusions* $r \sqsubseteq s$ and $r \sqsubseteq s^-$. The semantics is defined in terms of interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as usual. An $\mathcal{EL}$-*concept* is an $\mathcal{ELI}$-concept that does not use the constructor $\exists r^-.C$. An $\mathcal{EL}$-*ontology* is an $\mathcal{ELHI}$-ontology that uses only $\mathcal{EL}$-concepts and contains no role inclusions. A *basic concept* is a concept name or of one of the forms $\top$, $\bot$, $\exists r.\top$, and $\exists r^-.\top$. A DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$-ontology is a finite set of statements of form

$$B_1 \sqcap \cdots \sqcap B_n \sqsubseteq B \quad r \sqsubseteq s \quad r \sqsubseteq s^- \quad r_1 \sqcap \cdots \sqcap r_n \sqsubseteq \bot$$

where $B_1, \dots, B_n, B$ range over basic concepts and $r, s, r_1, \dots, r_n$ range over role names.

A *DL schema* is a schema that uses only unary and binary relation names, which we identify with $\mathsf{N_C}$ and $\mathsf{N_R}$ respectively. An $\mathbf{S}$-*ABox* is a database over DL schema $\mathbf{S}$. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$ if $A(a) \in \mathcal{A}$ implies $a \in A^{\mathcal{I}}$ and $r(a,b) \in \mathcal{A}$ implies $(a,b) \in r^{\mathcal{I}}$. In contrast to standard DL terminology, we speak of constants and facts also in the context of ABoxes, instead of individuals and assertions.

An *ontology-mediated query (OMQ)* is a tuple $Q = (\mathcal{O}, \mathbf{S}, q)$ with $\mathcal{O}$ an ontology, $\mathbf{S}$ a DL schema, and q a query such as a CQ. Let $\mathcal{A}$ be an $\mathbf{S}$-ABox. A tuple $\mathbf{a} \in \mathsf{adom}(\mathcal{A})$ is a *certain answer to* Q *on* $\mathcal{A}$ if $\mathbf{a} \in \mathsf{ans}_q(\mathcal{I})$ for every model $\mathcal{I}$ of $\mathcal{A}$ (viewed as a potentially infinite $\mathbf{S}$-database). We use $\mathsf{cert}_Q(\mathcal{A})$ to denote the set of all certain answers to Q on $\mathcal{A}$ and sometimes write $\mathcal{A}, \mathcal{O} \models q(\mathbf{a})$ when $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$. For OMQs $Q_1 = (\mathcal{O}_1, \Sigma, q_1)$ and $Q_2 = (\mathcal{O}_2, \Sigma, q_2)$ of the same arity, we say that Q_1 is *contained* in Q_2, denoted $Q_1 \subseteq Q_2$, if for every Σ-ABox $\mathcal{A}$, $\mathsf{cert}_{Q_1}(\mathcal{A}) \subseteq \mathsf{cert}_{Q_2}(\mathcal{A})$. Containment between an OMQ and a UCQ are defined in the expected way, and so is the converse containment.

A *global as view (GAV) mapping* over a schema $\mathbf{S}$ takes the form $\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \psi(\mathbf{x})$ where $\varphi(\mathbf{x}, \mathbf{y})$ is a conjunction of relational atoms over $\mathbf{S}$ and $\psi(\mathbf{x})$ is of the form $A(x)$, $r(x,y)$, or $r(x,x)$ with A a concept name and r a role name. We call $\varphi(\mathbf{x}, \mathbf{y})$ the *body* of the mapping and $\psi(\mathbf{x})$ its *head*. Every variable that occurs in the head must also occur in the body. Let $\mathbf{M}$ be a set of GAV mappings over a schema $\mathbf{S}$. For every $\mathbf{S}$-database D, the mappings in $\mathbf{M}$ produce an ABox $M(D)$, defined as follows:

$$\{R(\mathbf{a}) \mid D \models \varphi(\mathbf{a}, \mathbf{b}) \text{ and } \varphi(\mathbf{x}, \mathbf{y}) \rightarrow R(\mathbf{x}) \in \mathbf{M}\}.$$

This ABox can be physically materialized or left virtual; we do not make any assumptions regarding this issue.

An *OBDA specification* is a triple $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ where $\mathbf{S}$ is the *source schema*, $\mathbf{M}$ a finite set of mappings over $\mathbf{S}$, and $\mathcal{O}$ an ontology.[1] We use $\mathsf{sch}(\mathbf{M})$ to denote the schema that consists of all relation names that occur in the heads of mappings in $\mathbf{M}$. Informally, $\mathcal{S}$ is addressing source data in schema $\mathbf{S}$, translated into an ABox in schema $\mathsf{sch}(\mathbf{M})$ in terms of the mappings from $\mathbf{M}$ and then evaluated under the ontology $\mathcal{O}$. Note that $\mathcal{O}$ can use the relation names in $\mathsf{sch}(\mathbf{M})$ as well as additional concept and role names, and so can queries that are posed against the ABox.

We use $[\mathcal{L}, \mathcal{M}]$ to denote the set of all OBDA specifications $(\mathcal{O}, \mathbf{M}, \mathbf{S})$ where $\mathcal{O}$ is formulated in the ontology language $\mathcal{L}$ and all mappings in $\mathbf{M}$ are formulated in the mapping language $\mathcal{M}$ and call $[\mathcal{L}, \mathcal{M}]$ an *OBDA language*. An example of an OBDA language is $[\mathcal{ELHI}, \text{GAV}]$. In this paper, we shall concentrate on GAV mappings. While other types of mappings such as LAV and GLAV are also interesting (Poggi et al. 2008; Cima 2017), they are outside the scope of this paper.

Definition 1. Let $\mathcal{Q}_s$ and $\mathcal{Q}_t$ be query languages and $[\mathcal{L}, \mathcal{M}]$ an OBDA language.

1. The $\mathcal{Q}_s$*-to-*$\mathcal{Q}_t$ *verification problem in* $[\mathcal{L}, \mathcal{M}]$ is to decide, given an OBDA specification $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S}) \in [\mathcal{L}, \mathcal{M}]$, a source query $q_s \in \mathcal{Q}_s$, and a target query $q_t \in \mathcal{Q}_t$ of the same arity, whether q_t is a *realization of* q_s *in* $\mathcal{S}$, that is, whether $\mathsf{ans}_{q_s}(D) = \mathsf{cert}_Q(\mathbf{M}(D))$ for all $\mathbf{S}$-databases D, where $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$.
2. The $\mathcal{Q}_s$*-to-*$\mathcal{Q}_t$ *expressibility problem in* $[\mathcal{L}, \mathcal{M}]$ is to decide, given an OBDA specification $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S}) \in [\mathcal{L}, \mathcal{M}]$ and a source query $q_s \in \mathcal{Q}_s$, whether there is a realization q_t of q_s in $\mathcal{Q}_t$. We then say that q_s is $\mathcal{Q}_t$*-expressible in* $\mathcal{S}$.

Note that an alternative definition is obtained by quantifying only over those $\mathbf{S}$-databases D such that $D \cup \mathcal{O}$ is satisfiable. This does not make a difference for most of the setups studied in this paper since the involved DLs cannot express inconsistency.

Example 2. Assume that $\mathcal{S}$ contains a binary relation Man with $\mathsf{Man}(m,d)$ meaning that department d is managed by manager m and a ternary relation $\mathsf{Emp}(e,d,o)$ meaning that employee e works for department d in office o. Let $\mathbf{M}$ contain the GAV mappings

$$\begin{aligned} \mathsf{Man}(x,z) \wedge \mathsf{Emp}(y,z,u) &\rightarrow \mathsf{manages}(x,y) \\ \mathsf{Emp}(x,y,z) &\rightarrow \mathsf{Employee}(x) \end{aligned}$$

Then the source query $q_s(x) = \exists y\, \mathsf{Man}(x,y)$ is not expressible because the mappings do not provide sufficient data from the source. It trivially becomes expressible as $q_t(x) = \mathsf{Manager}(x)$ when we add the mapping

$$\mathsf{Man}(x,y) \rightarrow \mathsf{Manager}(x).$$

Next, we further add the following $\mathcal{EL}$-ontology $\mathcal{O}$:

$$\begin{aligned} \mathsf{Manager} &\sqsubseteq \mathsf{Employee} \\ \mathsf{Manager} &\sqsubseteq \exists \mathsf{manages}.\mathsf{Secretary} \end{aligned}$$

Then the source query $q_s(x) = \exists y \exists z\, \mathsf{Emp}(x,y,z)$, which formerly was expressible as $q_t(x) = \mathsf{Employee}(x)$, is no longer expressible due to the first CI in $\mathcal{O}$. Informally, all the required data is there, but it is mixed with other data and we have no way to separate. The source query $q_s(x,y) = \exists z \exists u\, \mathsf{Man}(x,z) \wedge \mathsf{Emp}(y,z,u)$, however, is expressible as $\mathsf{manages}(x,y)$ despite the second CI in $\mathcal{O}$, intuitively because the additional data mixed into $\mathsf{manages}$ by that CI always involves an anonymous constant introduced through the existential quantifier and is thus never returned as a certain answer.

The *size* of any syntactic object X such as a UCQ or an ontology, denoted $|X|$, is the number of symbols needed to write it, with names of concepts, roles, variables, etc. counting as one.

Characterizations

We characterize when a UCQ q_t over $\mathsf{sch}(\mathbf{M})$ is a realization of a UCQ q_s over the source schema $\mathbf{S}$ and then lift this characterization to the expressibility of q_s. This serves as a

[1] For readability, we consider a single data source, only. Multiple source databases can be represented as a single one by assuming that their schemas are disjoint and taking the union.

basis for deciding the expressibility and verification problems later on. The characterization applies the mappings from $\mathbf{M}$ forwards and backwards, as also done in query rewriting under views (Nash, Segoufin, and Vianu 2010; Afrati 2011), and suitably mixes in UCQ-rewritings of certain emerging OMQs.

Let $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ be an OBDA specification and $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q)$ an OMQ. A *rewriting* of Q is a query q_r over $\mathsf{sch}(\mathbf{M})$ of the same arity as q such that for all $\mathsf{sch}(\mathbf{M})$-ABoxes $\mathcal{A}$, $\mathsf{ans}_{q_r}(\mathcal{A}) = \mathsf{cert}_Q(\mathcal{A})$. We speak of a *UCQ rewriting* if q_r is a UCQ, of an *infinitary UCQ rewriting* if q_r is a potentially infinite UCQ, and so on. Note that there always exists a *canonical infinitary UCQ rewriting* that is obtained by taking all $\mathsf{sch}(\mathbf{M})$-ABoxes $\mathcal{A}$ and answers $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ and including $(\mathcal{A}, \mathbf{a})$ viewed as a CQ as a disjunct. This even holds when $\mathcal{O}$ is formulated in FO without equality. In fact, this follows from the definition of rewritings and the fact that OMQs with $\mathcal{O}$ formulated in FO without equality are preserved under homomorphisms (Bienvenu et al. 2014).

Let $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ be an OBDA specification and $\mathcal{A}$ an ABox that uses only concept and role names from $\mathsf{sch}(\mathbf{M})$. We say that a mapping $\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \psi(\mathbf{x})$ from $\mathbf{M}$ is *suitable* for a fact $\alpha \in \mathcal{A}$ if $\psi(\mathbf{x})$ and α are unifiable. We write $\mathbf{M}^{-}(\mathcal{A})$ to denote the set of $\mathbf{S}$-databases obtained from $\mathcal{A}$ as follows: for every fact $\alpha \in \mathcal{A}$, choose a suitable mapping $\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \psi(\mathbf{x})$ from $\mathbf{M}$ and include $R(\sigma(\mathbf{z}))$ in $\mathbf{M}^{-}(\mathcal{A})$ whenever $R(\mathbf{z})$ is an atom in $\varphi(\mathbf{x}, \mathbf{y})$ and where σ is the most general unifier of $\psi(\mathbf{x})$ and α, extended to replace every variable from $\mathbf{y}$ with a fresh constant. For example, for a fact $r(a, a) \in \mathcal{A}$ we can choose a mapping $R(x, y, z) \rightarrow r(x, y)$ and include $R(a, a, b)$ in $\mathbf{M}^{-}(\mathcal{A})$, where b is fresh. Both $\mathbf{M}$ and $\mathbf{M}^{-}$ lift to sets of databases and ABoxes as expected, that is, if S is a set of $\mathbf{S}$-databases, then $\mathbf{M}(S) = \{\mathbf{M}(D) \mid D \in S\}$ and if S is a set of ABoxes over $\mathsf{sch}(\mathbf{M})$, then $\mathbf{M}^{-}(S) = \bigcup_{\mathcal{A} \in S} \mathbf{M}^{-}(\mathcal{A})$.

In what follows, we shall often apply $\mathbf{M}$ to a CQ $q(\mathbf{x})$ viewed as a database, and view the result (which formally is a database) again as a CQ. In this case, the answer variables are again $\mathbf{x}$ and the equality atoms from $q(\mathbf{x})$ are readded to $\mathbf{M}(q)$. The same applies to UCQs and sets of databases, and to $\mathbf{M}^{-}(q)$ (where we also preserve answer variables and readd equality atoms). Note that $\mathbf{M}^{-}(q)$ gives a UCQ even when q was a CQ.

Example 3. Consider the CQ $q(x, y, z) = \exists u\, r(x, y) \wedge s(x, z) \wedge s(z, u) \wedge x = y$ and let $\mathbf{M}$ consist of the single mapping $r(x, y) \rightarrow r(x, y)$, that is, the role name r is simply copied and the role name s is dropped. Then $\mathbf{M}(q)$ viewed as a CQ is $p(x, y, z) = r(x, y) \wedge x = y$. Note that the answer variable z does not occur in an atom.

The following fundamental lemma describes the (non)-effect of applying $\mathbf{M}$ and $\mathbf{M}^{-}$ on query containment. It is explicit or implicit in many papers concerned with query rewriting under views or with query determinacy, see for example (Nash, Segoufin, and Vianu 2010; Afrati 2011).

Lemma 4. Let $\mathbf{M}$ be a set of GAV mappings, q, q_1 and q_2 UCQs over $\mathbf{S}$ and r, r_1 and r_2 UCQs over $\mathsf{sch}(\mathbf{M})$. Then:

1. If $q_1 \subseteq_{\mathbf{S}} q_2$, then $\mathbf{M}(q_1) \subseteq_{\mathsf{sch}(\mathbf{M})} \mathbf{M}(q_2)$.
2. If $r_1 \subseteq_{\mathsf{sch}(\mathbf{M})} r_2$, then $\mathbf{M}^{-}(r_1) \subseteq_{\mathbf{S}} \mathbf{M}^{-}(r_2)$.
3. $q \subseteq_{\mathbf{S}} \mathbf{M}^{-}(r)$ iff $\mathbf{M}(q) \subseteq_{\mathsf{sch}(\mathbf{M})} r$.

The next theorem characterizes realizations in terms of UCQ rewritings and $\mathbf{M}^{-}$. It can thus serve as a basis for deciding the verification problem.

Theorem 5. Let $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ be an OBDA specification from (FO(=), GAV), q_s a UCQ over $\mathbf{S}$, q_t a UCQ over $\mathsf{sch}(\mathbf{M})$, and q_r an infinitary UCQ rewriting of the OMQ $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$. Then q_t is a realization of q_s iff $q_s \equiv_{\mathbf{S}} \mathbf{M}^{-}(q_r)$.

Proof. "if". Assume that $q_s \equiv_{\mathbf{S}} \mathbf{M}^{-}(q_r)$. We have to show that q_t is a realization of q_s. Since q_r is a rewriting of Q, it suffices to prove that $\mathsf{ans}_{q_s}(D) = \mathsf{ans}_{q_r}(\mathbf{M}(D))$ for all $\mathbf{S}$-databases D.

For "$\subseteq$", assume that $\mathbf{a} \in \mathsf{ans}_{q_s}(D)$. Let p be $(D, \mathbf{a})$ viewed as a CQ. From $\mathbf{a} \in \mathsf{ans}_{q_s}(D)$, we obtain $p \subseteq q_s$, and $q_s \subseteq_{\mathbf{S}} \mathbf{M}^{-}(q_r)$ yields $p \subseteq_{\mathbf{S}} \mathbf{M}^{-}(q_r)$. With Point 3 of Lemma 4, it follows that $\mathbf{M}(p) \subseteq_{\mathbf{S}} q_r$, which by construction of p implies $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathbf{M}(D))$.

For "$\supseteq$", assume that $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathbf{M}(D))$. Let p be $(\mathbf{M}(D), \mathbf{a})$ viewed as a UCQ. Then, $p \subseteq_{\mathsf{sch}(\mathbf{M})} q_r$ and Point 3 of Lemma 4 yields $p' \subseteq_{\mathbf{S}} \mathbf{M}^{-}(q_r)$ where p' is $(D, \mathbf{a})$ viewed as a CQ. Together with $\mathbf{M}^{-}(q_r) \subseteq_{\mathbf{S}} q_s$, we obtain $p' \subseteq_{\mathbf{S}} q_s$, which implies that $\mathbf{a} \in \mathsf{ans}_{q_s}(D)$.

For the "only if" direction, assume that q_t is a realization of q_s. We have to show that $q_s \equiv_{\mathbf{S}} \mathbf{M}^{-}(q_r)$. Thus, let D be an $\mathbf{S}$-database and $\mathbf{a}$ a tuple from $\mathsf{adom}(D)$ whose length matches the arity of q_s. Further, let p be $(D, \mathbf{a})$ viewed as a database. Since q_t is a realization of q_s and q_r a rewriting of the OMQ Q, $\mathbf{a} \in \mathsf{ans}_{q_s}(D)$ iff $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathbf{M}(D))$. The latter is the case iff $\mathbf{M}(p) \subseteq_{\mathsf{sch}(\mathbf{M})} q_r$ which by Point 3 of Lemma 4 holds iff $p \subseteq_{\mathbf{S}} \mathbf{M}^{-}(q_r)$. This in turn is the case iff $\mathbf{a} \in \mathsf{ans}_{\mathbf{M}^{-}(q_r)}(D)$. ❏

The next theorem characterizes the expressibility of source queries in an OBDA specification. It has several interesting consequences. First, it implies that the UCQ $\mathbf{M}(q_s)$ is a realization of a UCQ q_s over $\mathbf{S}$ if there is any such realization. This is well known in the case without an ontology (Nash, Segoufin, and Vianu 2010; Afrati 2011) and is implicit in (Cima 2017) for a rather special case of OBDA. Second, the theorem provides a polynomial time reduction of expressibility to verification: q_s is expressible in $\mathcal{S}$ iff $\mathbf{M}(q_s)$ is a realization of q_s in $\mathcal{S}$. And third, it shows that if q_s is a CQ, then CQ-expressibility coincides with UCQ-expressibility. Thus, all lower bounds for CQ-to-CQ expressibility also apply to (U)CQ-to-UCQ expressibility and all upper bounds for UCQ-to-UCQ verification and expressibility also apply to the corresponding CQ-to-(U)CQ case.

Theorem 6. Let $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ be an OBDA specification from (FO(=), GAV), q_s a UCQ over $\mathbf{S}$, and q_r an infinitary UCQ rewriting of the OMQ $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), \mathbf{M}(q_s))$. Then q_s is UCQ-expressible in $\mathcal{S}$ iff $\mathbf{M}^{-}(q_r) \subseteq_{\mathbf{S}} q_s$. Moreover, if this is the case then $\mathbf{M}(q_s)$ is a realization of q_s in $\mathcal{S}$.

Proof. We first observe that

(a) if $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$, then $\mathbf{M}(q_s)$ is a realization of q_s in $\mathcal{S}$.

This actually follows from Theorem 5 because $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_r)$ always holds. In fact, since $\mathbf{M}(q_s)$ is the actual query in Q and since q_r is a rewriting of Q, we have $\mathbf{M}(q_s) \subseteq_{\mathsf{sch}(\mathbf{M})} q_r$; applying Point 3 of Lemma 4 then yields $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_r)$.

Note that (a) establishes the "if" part of Theorem 6. In view of Theorem 5 and by (a), we can prove both the "only if" and the "Moreover" part by showing that if there is any realization q_t of q_s in $\mathcal{S}$, then $\mathbf{M}(q_s)$ is a realization of q_s.

Thus assume that q_t is such a realization and let Q' be the OMQ $(\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$ and q'_r a UCQ-rewriting of Q'. We aim to show that

(b) $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} \mathbf{M}^-(q'_r)$.

This suffices since Theorem 5 yields $\mathbf{M}^-(q'_r) \subseteq_{\mathbf{S}} q_s$ and composing (b) with this containment gives $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$ that yields the desired result because of (a).

To establish (b), by Point 2 of Lemma 4 it suffices to show $q_r \subseteq_{\mathsf{sch}(\mathbf{M})} q'_r$. From Theorem 5, we get $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q'_r)$. Point 3 of Lemma 4 gives $\mathbf{M}(q_s) \subseteq_{\mathsf{sch}(\mathbf{M})} q'_r$. By the semantics of certain answers, this implies $(\mathcal{O}, \mathsf{sch}(\mathbf{M}), \mathbf{M}(q_s)) \subseteq_{\mathsf{sch}(\mathbf{M})} (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q'_r)$. Since the former OMQ is just Q and q_r is a rewriting of Q, we get $q_r \subseteq_{\mathsf{sch}(\mathbf{M})} (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q'_r)$. It thus remains to show $(\mathcal{O}, \mathsf{sch}(\mathbf{M}), q'_r) \subseteq q'_r$, which is exactly the statement of Lemma 24 in the appendix. ❏

The following corollary of Theorem 6 shows that while making the ontology logically stronger might make some source queries inexpressible (see Example 3), it never results in additional such queries becoming expressible.

Corollary 7. Let $\mathcal{S}_i = (\mathcal{O}_i, \mathbf{M}, \mathbf{S})$ $i \in \{1, 2\}$, be OBDA specifications from [FO, GAV] with $\mathcal{O}_1 \models \mathcal{O}_2$, $\mathcal{Q} \in \{\text{CQ}, \text{UCQ}\}$ and q_s from $\mathcal{Q}$. Then $\mathcal{Q}$-expressibility of q_s in $\mathcal{S}_1$ implies $\mathcal{Q}$-expressibility of q_s in $\mathcal{S}_2$.

Proof. Assume that q_s is $\mathcal{Q}$-expressible in $\mathcal{S}_1$. Then Theorem 6 gives that $\mathbf{M}(q_s)$ is a realization, and this query is also from $\mathcal{Q}$. We show that $\mathbf{M}(q_s)$ is also a realization of q_s in $\mathcal{S}_2$. Let $q_{r,i}$ be the canonical infinitary UCQ rewriting of the OMQ $Q_i = (\mathcal{O}_i, \mathsf{sch}(\mathbf{M}), \mathbf{M}(q_s))$, $i \in \{1, 2\}$. By Theorem 5, $q_s \equiv_{\mathbf{S}} \mathbf{M}^-(q_{r,1})$. Since $\mathcal{O}_1 \models \mathcal{O}_2$, we have $Q_2 \subseteq_{\mathsf{sch}(\mathbf{M})} Q_1$. This clearly implies that every CQ in $q_{r,2}$ is also in $q_{r,1}$. Thus $q_s \equiv_{\mathbf{S}} \mathbf{M}^-(q_{r,1})$ implies $q_s \supseteq_{\mathbf{S}} \mathbf{M}^-(q_{r,2})$. It remains to argue that $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_{r,2})$. Since $q_{r,2}$ is a rewriting of Q_2, we have $Q_2 \subseteq_{\mathsf{sch}(\mathbf{M})} q_{r,2}$. By the semantics and definition of Q_2, $\mathbf{M}(q_s) \subseteq_{\mathsf{sch}(\mathbf{M})} Q_2$ and thus $\mathbf{M}(q_s) \subseteq_{\mathsf{sch}(\mathbf{M})} q_{r,2}$. Point 3 of Lemma 4 yields $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_{r,2})$ as desired. ❏

Expressibility and Verification in DL-Lite

We consider OBDA specifications in which the ontology is formulated in a dialect of DL-Lite. The distinguishing feature of logics from this family is that finite UCQ rewritings of OMQs always exist. Therefore, Theorems 5 and 6 immediately imply decidability of the verification and expressibility problem, respectively. It is, however, well known that UCQ rewritings can become exponential in size (Gottlob et al. 2014) and thus optimal complexity bounds are not immediate.

We consider the dialect DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$ as a typical representative of the DL-Lite family of logics. However, our results also apply to many other dialects since their proof rests only on the following properties, established in (Artale et al. 2009).

Theorem 8. In DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$,

1. all OMQs Q have a UCQ-rewriting in which all CQs are of size polynomial in $|Q|$;
2. OMQ evaluation is in NP in combined complexity.

We remark that the results presented in this section are related to those obtained in (Cima 2017), where the DL-Lite$_{\mathcal{A},id}$ dialect of DL-Lite is considered, mappings are GLAV, and queries CQs. A main difference is that Cima's technical results concern rewritings that are complete but not necessarily sound, which corresponds to replacing '$\mathsf{ans}_{q_s}(D) = \mathsf{cert}_Q(\mathbf{M}(D))$' in Definition 1 with '$\mathsf{ans}_{q_s}(D) \subseteq \mathsf{cert}_Q(\mathbf{M}(D))$'. Some of his technical constructions are similar to ours. Note that DL-Lite$_{\mathcal{A},id}$ also satisfies the conditions from Theorem 8 and thus our results apply to [DL-Lite$_{\mathcal{A},id}$, GAV] as well.

For an OMQ $Q = (\mathcal{O}, \mathbf{S}, q)$, with $\mathcal{O}$ formulated in FO($=$) and q a UCQ, the *canonical UCQ-rewriting of size* n is the UCQ q_c that consists of all pairs $(\mathcal{A}, \mathbf{a})$ viewed as a CQ where $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ and $|\mathcal{A}| \leq n$. The following lemma is interesting in connection with Point 1 of Theorem 8 as it allows us to concentrate on canonical UCQ rewritings of polynomial size.

Lemma 9. Let $Q = (\mathcal{O}, \mathbf{S}, q)$ be an OMQ with $\mathcal{O}$ formulated in FO($=$) and q a UCQ. If Q has a UCQ-rewriting q_r in which all CQs are of size at most n, then the canonical UCQ-rewriting q_c of size n is also a rewriting of Q.

We are now ready to establish the upper bound.

Theorem 10. In [DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$, GAV], the UCQ-to-UCQ expressibility and verification problems are in Π^p_2.

Proof. As remarked before Theorem 6, expressibility polynomially reduces to verification and thus it suffices to consider the latter. Hence let the following be given: an OBDA specification $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ from [DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$, GAV], a UCQ q_s over schema $\mathbf{S}$, and a UCQ q_t over the schema $\mathsf{sch}(\mathbf{M})$. Let n be the size of this input.

Let $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$ and note that the size of Q is polynomial in n. By Point 1 of Theorem 8, we can assume that Q has a UCQ-rewriting in which all CQs are of size $P(n)$, P a polynomial. By Lemma 9, we can even assume that this rewriting is the canonical UCQ-rewriting q_c of size $P(n)$. By Theorem 5, q_t is thus a realization of q_s iff $q_s \equiv_{\mathbf{S}} \mathbf{M}^-(q_c)$. We show that both inclusions of this equivalence can be checked in Π^p_2.

First, consider the inclusion $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_c)$. It holds iff for every q in q_s, there is a p in $\mathbf{M}^-(q_c)$ and a homomorphism $p \to q$. This condition can be checked even in NP:

iterate over all CQs q in q_s (of which there are at most n), guess a disjunct p from $\mathbf{M}^-(q_c)$, and verify in NP that $p \to q$.

To guess a p in $\mathbf{M}^-(q_c)$, it suffices to guess a pair $(\mathcal{A}, \mathbf{a})$ in q_c and suitable mappings from $\mathbf{M}$ for every fact in $\mathcal{A}$, which determine p. Then, p can be computed in polynomial time from $\mathcal{A}$ and these suitable mappings. We guess the pair $(\mathcal{A}, \mathbf{a})$ from q_c by guessing an arbitrary ABox $\mathcal{A}$ of size at most $P(n)$ and then verifying that $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$. By Point 2 of Theorem 8 this verification is possible in NP.

We next consider the inclusion $\mathbf{M}^-(q_c) \subseteq_{\mathbf{S}} q_s$. This holds iff for every p in $\mathbf{M}^-(q_c)$, there is a CQ q in q_s such that $q \to p$. We can thus universally guess a p in $\mathbf{M}^-(q_c)$, then iterate over all CQs q in q_s, and for each such q check in NP whether $q \to p$. For universally guessing p, we actually guess a CQ p of size at most $P'(n)$ and then verify that it is in $\mathbf{M}^-(q_c)$. It has already been argued above that this is possible in NP. Overall, we obtain a Π_2^p-algorithm, as desired. ❏

We next show that the expressibility problem in $[\text{DL-Lite}^{\mathcal{R}}_{\mathsf{horn}}, \text{GAV}]$ is Π_2^p-hard, and thus the same holds for the verification problem. Interestingly, the lower bound already applies when the ontology is empty and the source query is a CQ. As noted in the introduction, this shows that expressibility of a source CQ as a (U)CQ over UCQ views is Π_2^p-hard, and in fact it is Π_2^p-complete by Theorem 10. This corrects a (very likely) erroneous statement of NP-completeness in (Levy et al. 1995).

Theorem 11. The CQ-to-CQ expressibility problem is Π_2^p-hard for GAV mappings and the empty ontology.

The proof is by reduction of validity of $\forall\exists$-QBFs. By Theorem 6, expressibility in the absence of an ontology amounts to checking the containment $\mathbf{M}^-(\mathbf{M}(q_s)) \subseteq q_s$, which is equivalent to the $\forall\exists$-statement that for all $p \in \mathbf{M}^-(\mathbf{M}(q_s))$ there is a homomorphism $q_s \to p$. Hence we encode a $\forall\exists$-quantified Boolean formula such that the outer universal quantifiers correspond to the different choices of mappings when taking a $p \in \mathbf{M}^-(\mathbf{M}(q_s))$, whereas the inner existential quantifiers of the formulas correspond to homomorphisms $q_s \to p$.

Expressibility in $\mathcal{ELHI}$: Upper Bound for Rooted Queries

We show that the expressibility problem in $[\mathcal{ELHI}, \text{GAV}]$ is in CONEXPTIME when the source query is a rooted UCQ. Here, a CQ q is *rooted* or an *rCQ* if every variable from q is reachable from an answer variable in the hypergraph $H_q := (\mathsf{var}(q), \{\{x_1, \dots, x_n\} \mid R(x_1, \dots, x_n) \in q\})$ and a UCQ is *rooted* or an *rUCQ* if every CQ in it is rooted. In practice, many relevant queries are rooted. Our aim is thus to prove the following result.

Theorem 12. In $[\mathcal{ELHI}, \text{GAV}]$, the rUCQ-to-UCQ expressibility problem is in CONEXPTIME.

To prepare for lifting the result from expressibility to verification later, we actually establish a slightly more general result as needed. Note that the following implies Theorem 12 since, by Theorem 6, we can simply use $\mathbf{M}(q_s)$ for q_t.

Theorem 13. Given an OBDA setting $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ from $[\mathcal{ELHI}, \text{GAV}]$, an rUCQ q_s over $\mathbf{S}$, and a UCQ q_t over $\mathsf{sch}(\mathbf{M})$, it is in CONEXPTIME to decide whether $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$, where q_r is an infinitary UCQ-rewriting of the OMQ $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$.

To prove Theorem 13, we now describe a NEXPTIME algorithm for deciding the complement of the problem described there: we want to check whether $\mathbf{M}^-(q_r) \not\subseteq q_s$, that is, whether there is a CQ p in the UCQ $\mathbf{M}^-(q_r)$ such that $q \not\to p$ for all CQs q in q_s. Because all rewritings of Q are equivalent, it suffices to prove the theorem for any particular infinitary UCQ-rewriting q_r of Q. We choose to work with the canonical one introduced at the beginning of the characterizations section. The algorithm is as follows:

1. Guess a $\mathsf{sch}(\mathbf{M})$-ABox $\mathcal{A}$ such that $|\mathsf{adom}(\mathcal{A})| \leq |q_t| + |q_t| \cdot |\mathcal{O}|^{|q_s|+1}$ and a tuple $\mathbf{a}$ in $\mathcal{A}$ of the same arity as q_t.
2. Verify that $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ to make sure that $(\mathcal{A}, \mathbf{a})$ viewed as a CQ is in the UCQ q_r. This can be done by an algorithm that is exponential in $|\mathcal{O}|$ and $|q_t|$, but only polynomial in $|\mathcal{A}|$; see for example (Krisnadhi and Lutz 2007). Hence, the overall running time is single exponential in the size of the original input.
3. Guess a disjunct p from the UCQ $\mathbf{M}^-(\mathcal{A}, \mathbf{a})$ by guessing, for each fact α in $\mathcal{A}$, a suitable mapping from $\mathbf{M}$. Note that both $\mathcal{A}$ and p are of single exponential size.
4. Verify that $q \not\to p$ for all CQs q in the rUCQ q_s. This can be done in single exponential time using brute force.

This is clearly a NEXPTIME algorithm.

Lemma 14. The algorithm decides the complement of the problem in Theorem 13.

It is easy to verify the soundness part: a successful run of the algorithm identifies p as a CQ in $\mathbf{M}^-(q_r)$ such that for any disjunct q of q_s, $q \not\to p$. Completeness is less obvious, mainly because of the magical size bound used in Step 1. We need some preliminaries.

An ABox $\mathcal{A}$ is *tree-shaped* if the undirected graph $G_{\mathcal{A}} = (\mathsf{adom}(\mathcal{A}), \{\{a, b\} \mid r(a, b) \in \mathcal{A}\})$ is acyclic, connected and $r(a, b) \in \mathcal{A}$ implies that $s(a, b) \notin \mathcal{A}$ for all role names $s \neq r$ and that $s(b, a) \notin \mathcal{A}$ for all role names s. An ABox $\mathcal{A}$ is *pseudo tree-shaped with core* $\mathcal{C} \subseteq \mathcal{A}$ if there is a tree-shaped ABox $\mathcal{A}_a$ with root a for every $a \in \mathsf{adom}(\mathcal{C})$, with mutually disjoint domains, such that $\mathcal{A} = \mathcal{C} \cup \bigcup_{a \in \mathsf{adom}(\mathcal{C})} \mathcal{A}_a$. The *outdegree* of $\mathcal{A}$ is the maximal outdegree of the trees underlying any of the $\mathcal{A}_a$.

The following lemma is an adaptation of Proposition 23 in Appendix B of (Bienvenu et al. 2016):

Lemma 15. Consider an $\mathcal{ELHI}$-ontology $\mathcal{O}$, an OMQ $Q = (\mathcal{O}, \mathbf{S}, q)$ with q a UCQ, an $\mathbf{S}$-ABox $\mathcal{A}$, and an answer $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$. Then there is a pseudo tree-shaped $\mathbf{S}$-ABox $\mathcal{A}'$ and a tuple $\mathbf{a}'$ in the core of $\mathcal{A}'$ such that

1. the core of $\mathcal{A}'$ is not larger than $|q|$ and the outdegree of $\mathcal{A}$ is not larger than $|\mathcal{O}|$;
2. $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}')$;
3. there is a homomorphism h from $\mathcal{A}'$ to $\mathcal{A}$ with $h(\mathbf{a}') = \mathbf{a}$.

We are now ready for the completeness part of Lemma 14. Assume that there is a CQ p in $\mathbf{M}^-(q_r)$ such that for any disjunct q of q_s, $q \not\rightarrow p$. Since q_r is the canonical infinitary UCQ-rewriting of Q, p is of the form $\mathbf{M}^-(\mathcal{A}, \mathbf{a})$ where $\mathcal{A}$ is a $\mathsf{sch}(\mathbf{M})$-ABox with $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$. Let $\mathcal{A}'$ be the pseudo tree-shaped ABox and $\mathbf{a}'$ the answer whose existence is guaranteed by Lemma 15. Moreover, let $\mathcal{A}''$ be the result of removing from $\mathcal{A}'$ all facts that contain at least one constant whose distance from the core is larger than $|q_s|$ and adding for all constants a of distance exactly $|q_s|$ from the core both $A(a)$ for all concept names A in $\mathsf{sch}(\mathbf{M})$ and $r(a, a)$ for all role names r in $\mathsf{sch}(\mathbf{M})$. We show in the appendix that $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}'')$ and that there is a CQ p in $\mathbf{M}^-(\mathcal{A}'', \mathbf{a}')$ such that for any disjunct q of q_s, $q \not\rightarrow p$ (this depends on q_s being rooted). The former implies that $(\mathcal{A}'', \mathbf{a}')$, seen as CQ, is a disjunct in q_r and hence Step 2 of the algorithm succeeds. The latter guarantees that Step 4 succeeds. Moreover, $\mathcal{A}''$ satisfies the size bound given in Step 1 of the algorithm.

Expressibility in $\mathcal{ELHI}$: Upper Bound for Unrestricted Queries

We consider the UCQ-to-UCQ expressibility problem in $[\mathcal{ELHI}, \text{GAV}]$, that is, we drop the assumption from the previous section that the source query is rooted. This increases the complexity from coNExpTime to 2ExpTime. Note that similar effects have been observed in the context of different reasoning problems in (Lutz 2008; Bienvenu et al. 2016). In this section, we show the upper bound.

Theorem 16. In $[\mathcal{ELHI}, \text{GAV}]$, the UCQ-to-UCQ expressibility problem is in 2ExpTime.

As in the rooted case, we again prove a slightly more general result that can be reused when studying the verification problem. We can obtain Theorem 16 from the following by setting $q_t = \mathbf{M}(q_s)$ and applying Theorem 6.

Theorem 17. Given an OBDA setting $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ from $[\mathcal{ELHI}, \text{GAV}]$ a UCQ q_s over $\mathbf{S}$, and a UCQ q_t over $\mathsf{sch}(\mathbf{M})$, it is in 2ExpTime to decide whether $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$, where q_r is an infinitary UCQ-rewriting of the OMQ $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$.

To prove Theorem 17, we start by choosing a suitable UCQ-rewriting q_r. Instead of working with the canonical infinitary UCQ-rewriting, here we prefer to use the UCQ that consists of all pairs $(\mathcal{A}, \mathbf{a})$ viewed as a CQ and where $\mathcal{A}$ is a pseudo tree-shaped $\mathsf{sch}(\mathbf{M})$-ABox that satisfies $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ and is of the dimensions stated in Lemma 15, that is, the core of $\mathcal{A}$ is not larger than $|q|$ and the outdegree of $\mathcal{A}$ is not larger than $|\mathcal{O}|$. Due to that lemma, q_r clearly is an infinitary UCQ-rewriting of Q.

We give a decision procedure for the complement of the problem in Theorem 17. We thus have to decide whether there is a pseudo-tree shaped $\mathsf{sch}(\mathbf{M})$-ABox $\mathcal{A}$ of the mentioned dimensions, an $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$, and a CQ p in the UCQ $\mathbf{M}^-(\mathcal{A}, a)$ such that $q \not\rightarrow p$ for all CQs q in q_s. This can be done by constructing a two-way alternating parity tree automaton (TWAPA) $\mathfrak{A}$ on finite trees that accepts exactly those trees that represent a triple $(\mathcal{A}, \mathbf{a}, p)$ with the components as described above, and then testing whether the language accepted by $\mathfrak{A}$ is empty.

In the following, we detail this construction. We reuse some encodings and notation from a TWAPA construction that is employed in (Bienvenu et al. 2016) to decide OMQ containment as this saves us from redoing certain routine work. A *tree* is a non-empty (and potentially infinite) set $T \subseteq \mathbb{N}^*$ closed under prefixes. We say that T is *m-ary* if $T \subseteq \{1, \ldots, m\}^*$ and call the elements of T the *nodes* of the tree and ε its *root*. For an alphabet Γ, a *Γ-labeled tree* is a pair (T, L) with T a tree and $L : T \rightarrow \Gamma$ a node labeling function.

We encode triples $(\mathcal{A}, \mathbf{a}, p)$ as finite $(|\mathcal{O}| \cdot |q_t|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled trees, where Σ_ε is the alphabet used for labeling the root node and Σ_N is for non-root nodes. These alphabets are different because the root of a tree represents the core part of a pseudo tree-shaped ABox whereas each non-root node represents a single constant of the ABox that is outside the core. Let $\mathsf{C}_{\mathsf{core}}$ be a fixed set of $|q_t|$ constants. Formally, the alphabet Σ_ε is the set of all triples $(\mathcal{B}, \mathbf{a}, \mu)$ where $\mathcal{B}$ is a $\mathsf{sch}(\mathbf{M})$-ABox of size at most $|q_t|$ that uses only constants from $\mathsf{C}_{\mathsf{core}}$, $\mathbf{a}$ is a tuple over $\mathsf{C}_{\mathsf{core}}$ whose length matches the arity of q_s, and μ associates every fact α in $\mathcal{B}$ with a mapping $\mu(\alpha) \in \mathbf{M}$ that is suitable for α. The alphabet Σ_N consists of all triples (Θ, M, μ) where $\Theta \subseteq (\mathsf{N_C} \cap \mathsf{sch}(\mathbf{M})) \uplus \{r, r^- \mid r \in \mathsf{N_R} \cap \mathsf{sch}(\mathbf{M})\} \uplus \mathsf{C}_{\mathsf{core}}$ contains exactly one (potentially inverse) role and at most one element of $\mathsf{C}_{\mathsf{core}}$, $M \in \mathbf{M}$ is a mapping suitable for the fact $r(a, b)$ with r the unique role name in Θ, and μ assigns to each $A \in \Theta$ a mapping $\mu(A) \in \mathbf{M}$ suitable for the fact $A(a)$.[2] In the following, a *labeled tree* generally means a $(|\mathcal{O}| \cdot |q_t|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled tree.

A labeled tree is *proper* if (i) the root node is labeled with a symbol from Σ_ε, (ii) each child of the root is labeled with a symbol from Σ_N that contains an element of $\mathsf{C}_{\mathsf{core}}$, (iii) every other non-root node is labeled with a symbol from Σ_N that contains no constant name, and (iv) every non-root node has at most $|\mathcal{O}|$ successors and (v) for every $a \in \mathsf{C}_{\mathsf{core}}$, the root node has at most $|\mathcal{O}|$ successors whose label includes a. A proper labeled tree (T, L) with $L(\varepsilon) = (\mathcal{B}, \mathbf{a}, \mu)$ *encodes* the triple $(\mathcal{A}, \mathbf{a}, p)$ where $\mathcal{A}$ is the ABox

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

$\Theta(x)$ denoting Θ when $L(x) = (\Theta, M, \mu)$ (and undefined otherwise), and where p is the CQ from $\mathbf{M}^-(\mathcal{A})$ that can be obtained by choosing for every fact in $\mathcal{A}$ the suitable mapping from $\mathbf{M}$ assigned to it by L.

The desired TWAPA $\mathfrak{A}$ is obtained as the intersection of two TWAPAs $\mathfrak{A}_1$ and $\overline{\mathfrak{A}}_2$, where $\mathfrak{A}_1$ accepts exactly the proper labeled trees (T, L) that encode a pair $(\mathcal{A}, \mathbf{a}, p)$ with $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ and $\overline{\mathfrak{A}}_2$ is obtained as the complement of an automaton $\mathfrak{A}_2$ that accepts a proper labeled tree (T, L) encoding a pair $(\mathcal{A}, \mathbf{a}, p)$ iff $q \rightarrow p$ for some CQ q in q_s. In fact, the automaton $\mathfrak{A}_1$ is what we can reuse from (Bienvenu et al. 2016), see Point 1 in Proposition 13 there. The only difference is that our trees are decorated in a richer way, so

[2] Here, a and b are arbitrary but fixed constants.

in our case the TWAPA ignores the part of the labeling that is concerned with mappings from $\mathbf{M}$. The number of states of $\mathfrak{A}_1$ is single exponential in $|q_t|$ and $|\mathcal{O}|$.

We now sketch the construction of the automaton $\mathfrak{A}_2$ for a single CQ q of q_s (the general case can be dealt with using union). Let $q_1, \dots, q_k$ be the maximal connected components of q. We define automata $\mathfrak{A}_{2,1}, \dots, \mathfrak{A}_{2,k}$ where $\mathfrak{A}_{2,i}$ accepts (T, L) encoding $(\mathcal{A}, \mathbf{a}, p)$ iff $q_i \to p$, and then intersect to obtain $\mathfrak{A}_2$. Let (T, L) be a proper labeled tree. A set $T' \subseteq T$ is a *subtree* of T if for any $s, t \in T'$, all nodes from T that are on the shortest (undirected) path from s to t are in T'. We use (T', L) to denote the restriction of (T, L) to T' and $\mathbf{M}^-(T', L)$ to denote the subquery of p which contains only the atoms in p that can be derived from the part of $\mathcal{A}$ generated by the subtree (T', L) of (T, L).

To define $\mathfrak{A}_{2,i}$, let $\mathcal{C}$ denote the set of all labeled trees (T', L) of size at most $|q_i|$ such that there is a proper labeled tree (T, L) and $q_i \to \mathbf{M}^-(T', L)$ with a homomorphism that only needs to respect the answer variables from q that actually occur in q_i (if there are any, then T' must thus contain the root of T). The automaton $\mathfrak{A}_{2,i}$ is then constructed such that it accepts a proper labeled tree (T, L) iff it contains a subtree from $\mathcal{C}$. It should be clear that such an automaton can be constructed using only single exponentially many states. Moreover, it can be verified that $\mathcal{A}_{2,i}$ accepts exactly the desired trees. We obtain an overall automaton with single exponentially many states which together with the EXPTIME-complete emptiness problem of TWAPAs gives Theorem 17.

Verification in $\mathcal{ELHI}$: Upper Bounds

We show that in $[\mathcal{ELHI}, \text{GAV}]$, the complexity of the verification problem is not higher than the complexity of the expressivity problem both in the rooted and in the general case.

Theorem 18. In $[\mathcal{ELHI}, \text{GAV}]$,

1. the rUCQ-to-UCQ verification problem can be decided in CONEXPTIME.
2. the UCQ-to-UCQ verification problem can be decided in 2EXPTIME.

Recall the characterization of realizations from Theorem 5: a UCQ q_t is a realization of q_s iff $q_s \equiv \mathbf{M}^-(q_r)$, where q_r is a rewriting of the OMQ $(\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$. The inclusion $q_s \supseteq \mathbf{M}^-(q_r)$ is already treated by Theorems 13 and 17 and thus it remains to show that the converse inclusion can be decided in the relevant complexity class. We show that it is actually in EXPTIME even in the unrooted case. We thus aim to prove the following.

Theorem 19. Given an OBDA setting $\mathcal{S} = (\mathcal{O}, \mathbf{M}, \mathbf{S})$ from $[\mathcal{ELHI}, \text{GAV}]$, a UCQ q_s over $\mathbf{S}$, and a UCQ q_t over $\mathsf{sch}(\mathbf{M})$, it is in EXPTIME to decide whether $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_r)$, where q_r is an infinitary UCQ-rewriting of the OMQ $Q = (\mathcal{O}, \mathsf{sch}(\mathbf{M}), q_t)$.

For what follows, it is convenient to assume that the ontology $\mathcal{O}$ is in *normal form*, that is, all CIs in it are of one of the forms $\top \sqsubseteq A$, $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, and $\exists r.A \sqsubseteq B$ where A, B and all A_i range over concept names and r ranges over roles. It is well-known that every $\mathcal{ELHI}$-ontology $\mathcal{O}$ can be converted into an $\mathcal{ELHI}$-ontology $\mathcal{O}'$ in normal form in linear time such that $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$ in the model-theoretic sense (Baader et al. 2017). It is easy to verify that for the verification problem, we can w.l.o.g. assume the involved ontology to be in normal form.

We again start by choosing a suitable concrete infinitary UCQ-rewriting to use for q_r. As in the previous section, we would like to use CQs derived from pseudo tree-shaped ABoxes of certain dimension that entail an answer to the OMQ Q, as sanctioned by Lemma 15. Here, however, we use a slight strengthening of that lemma where Condition 2 is replaced with the following strictly stronger Condition $2'$, where $\mathcal{C}'$ is the core of the pseudo tree-shaped ABox $\mathcal{A}'$:

$$\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{C}' \cup \{A(a) \mid \mathcal{A}', \mathcal{O} \models A(a),\ a \in \mathsf{dom}(\mathcal{C}')\}).$$

This condition essentially says that, in the universal model of $\mathcal{A}'$ and $\mathcal{O}$ (defined in the appendix), there is a homomorphism h from a CQ in the UCQ q in Q that only involves constants from the core and 'anonymous subtrees' (generated by existential quantifiers) below them. This is true only since $\mathcal{O}$ is assumed to be in normal form and it is a consequence of the proof of Lemma 15 where one chooses a homomorphism h from a CQ in q to the universal model of $\mathcal{A}$ and $\mathcal{O}$, selecting as the core $\mathcal{C}'$ of the pseudo-tree ABox $\mathcal{A}'$ to be constructed all constants a from $\mathcal{A}$ that are in the range of h or which root an anonymous subtree that contains an element in the range of h, and then unraveling the rest of $\mathcal{A}$. Summing up, we thus use for q_r the set of all pairs $(\mathcal{A}, \mathbf{a})$ viewed as a CQ where $\mathcal{A}$ is a pseudo tree-shaped $\mathsf{sch}(\mathbf{M})$-ABox with core $\mathcal{C}$ that satisfies

$$\mathbf{a} \in \mathsf{cert}_Q(\mathcal{C} \cup \{A(a) \mid \mathcal{A}, \mathcal{O} \models A(a),\ a \in \mathsf{dom}(\mathcal{C})\})$$

and is of the dimensions stated in Lemma 15.

For deciding $q_s \subseteq_{\mathbf{S}} \mathbf{M}^-(q_r)$, we need to show that for every disjunct q in q_s there is a disjunct p in $\mathbf{M}^-(q_r)$ such that $p \to q$. We can do this for every disjunct q of q_s separately. Hence let q be such a disjunct. To find a CQ p in $\mathbf{M}^-(q_r)$ with $p \to q$, we again aim to utilize TWAPAs. As in the previous section, let $\mathsf{C}_{\mathsf{core}}$ be a fixed set of $|q_t|$ constants. A *homomorphism pattern* for q_t is a function λ that maps every variable y in q_t to a pair $(a, o) \in \mathsf{C}_{\mathsf{core}} \times \{\mathsf{core}, \mathsf{subtree}\}$. Informally, λ is an abstract description of a homomorphism h from q_t to the universal model of a pseudo-tree ABox $\mathcal{A}$ and $\mathcal{O}$ (assume that the core of $\mathcal{A}$ uses only constants from $\mathsf{C}_{\mathsf{core}}$) such that $h(x) = a$ when $\lambda(x) = (a, \mathsf{core})$ and $h(x)$ is an element in the anonymous subtree below a when $\lambda(x) = (a, \mathsf{subtree})$.

We build one TWAPA $\mathfrak{A}^\lambda$ for every homomorphism pattern λ (there are single exponentially many). These TWAPAs again run on $(|\mathcal{O}| \cdot |q_t|)$-ary $\Sigma_\varepsilon \cup \Sigma_N$-labeled trees that encode a triple $(\mathcal{A}, \mathbf{a}, p)$, defined exactly as in the previous section and from now on are only referred to as labeled trees. We also use the same notion of properness as in the previous section. The TWAPA $\mathfrak{A}^\lambda$ will accept exactly those labeled trees (T, L) that are proper and encode a triple $(\mathcal{A}, \mathbf{a}, p)$ such that

1. there is a homomorphism h from $q_t(\mathbf{x})$ to the universal model of $\mathcal{A}$ and $\mathcal{O}$ that satisfies $h(\mathbf{x}) = \mathbf{a}$ and follows the homomorphism pattern λ and
2. $p \rightarrow q$.

Note that it would be sufficient to demand in Point 1 that $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$ and recall that, in the previous section, we have reused an automaton from (Bienvenu et al. 2016) which checks exactly this condition. That automaton, however, has exponentially many states because it is built using a construction known under various names such as query splitting, forest decomposition, and squid decomposition. To attain an ExpTime upper bound, though, the automaton $\mathfrak{A}^\lambda$ can only have polynomially many states. This is in fact the reason why we have the strengthened Condition 2 of Lemma 15.

We construct the automaton $\mathfrak{A}^\lambda$ as the intersection of three automata $\mathfrak{A}_{\mathsf{proper}}$, $\mathfrak{A}_1^\lambda$, and $\mathfrak{A}_2$, where $\mathfrak{A}_{\mathsf{proper}}$ accepts if the input tree is proper, $\mathfrak{A}_1^\lambda$ accepts trees that encode a triple $(\mathcal{A}, \mathbf{a}, p)$ that satisfy Condition 1 for λ and $\mathfrak{A}_2$ accepts trees that encode a triple $(\mathcal{A}, \mathbf{a}, p)$ that satisfy Condition 2. It is easy to build the automaton $\mathfrak{A}_{\mathsf{proper}}$ and we leave out the details. The precise construction of $\mathfrak{A}_1^\lambda$ and $\mathfrak{A}_2$ can be found in the appendix, we only give a brief description here. Automaton $\mathfrak{A}_1^\lambda$ works as follows: it accepts labeled trees (T, L) whose root node label $(\mathcal{B}, \mathbf{a}, \mu)$ is such that the extension of the ABox $\mathcal{B}$ with certain facts of the form $A(a)$ results in a universal model which admits a homomorphism following pattern λ, and it then verifies that the facts $A(a)$ used in the extension can be derived from the ABox encoded by (T, L). The automaton $\mathfrak{A}_2$ checks Condition 2 by traversing the input tree once from the root to the leaves and guessing the homomorphism from p to q along the way. Some care is required since p is represented only implicitly in the input.

Overall, we obtain single exponentially many automata with polynomially many states each and we answer 'yes' if any of the automata recognizes a non-empty language. This gives Theorem 19.

Expressibility and Verification in $\mathcal{EL}$: Lower Bounds

We establish lower bounds that match the upper bounds obtained in the previous three sections and show that they even apply to $[\mathcal{EL}, \text{GAV}]$, that is, inverse roles are not required.

Theorem 20.

1. In $[\mathcal{EL}, \text{GAV}]$, the rUCQ-to-UCQ expressibility and verification problems are coNExpTime-hard.
2. In $[\mathcal{EL}, \text{GAV}]$, the UCQ-to-UCQ expressibility and verification problems are 2ExpTime-hard

By Theorem 6, it suffices to establish the lower bounds for the expressibility problem. We prove both points of Theorem 20 by a reduction from certain OMQ containment problems. For Point 1, we reduce from the following problem.

Theorem 21. (Bienvenu et al. 2016) Containment between OMQs $Q_1 = (\mathcal{O}, \Sigma, q_1)$ and $Q_2 = (\mathcal{O}, \Sigma, q_2)$ with $\mathcal{O}$ an $\mathcal{ELI}$-ontology, q_1 an AQ, and q_2 a rooted UCQ is coNExpTime-hard even when

1. $q_2(x)$ uses only symbols from Σ and
2. no symbol from Σ occurs on the right-hand side of a CI in $\mathcal{O}$.

We first establish Point 1 of Theorem 20 for $[\mathcal{ELI}, \text{GAV}]$ instead of for $[\mathcal{EL}, \text{GAV}]$ and in a second step show how to get rid of inverse roles. To reduce the containment problem in Theorem 21 to rUCQ-to-UCQ expressibility in $[\mathcal{ELI}, \text{GAV}]$, let $Q_1 = (\mathcal{O}, \Sigma, A_0(x))$ and $Q_2 = (\mathcal{O}, \Sigma, q)$ be as in that theorem. We define an OBDA-specification $\mathcal{S} = (\mathcal{O}', \mathbf{M}, \mathbf{S})$ and a query q_s over $\mathbf{S}$ as follows. Let B be a concept name that does not occur in Q_1 and Q_2. Set

$$\begin{aligned} \mathcal{O}' &= \mathcal{O} \cup \{A_0 \sqsubseteq B\} \\ \mathbf{S} &= \Sigma \cup \{B\} \\ q_s(x) &= B(x) \vee q(x) \end{aligned}$$

Note that q_s is a rooted UCQ, as required. Moreover, the set $\mathbf{M}$ of mappings contains $A(x) \rightarrow A(x)$ for all concept names $A \in \mathbf{S}$ and $r(x, y) \rightarrow r(x, y)$ for all role names $r \in \mathbf{S}$. Informally, the CI $A_0 \sqsubseteq B$ 'pollutes' B, potentially preventing the disjunct $B(x)$ of q_s to be expressible, but this is not a problem if (and only if) $Q_1 \not\subseteq Q_2$.

Lemma 22. $Q_1 \subseteq Q_2$ iff q_s is UCQ-expressible in $\mathcal{S}$.

In short, $Q_1 \not\subseteq Q_2$ iff there is a tree-shaped Σ-ABox witnessing this iff such an ABox, viewed as a CQ, is a disjunct of an infinitary UCQ-rewriting q_r of the OMQ $Q = (\mathcal{O}', \mathbf{S}, q_s)$ iff $q_r \not\subseteq q_s$. The latter is the case iff q_s is not UCQ-expressible in $\mathcal{S}$ by Lemma 6 and since $\mathbf{M}(q_s) = q_s$ and $\mathbf{M}^-(q_r) = q_r$.

In the appendix, we describe how to replace the $\mathcal{ELI}$-ontology $\mathcal{O}$ with an $\mathcal{EL}$-ontology. The crucial observation is that the hardness proof from (Bienvenu et al. 2016) uses only a single symmetric role S implemented as a composition $r_0^-; r_0$ with r_0 a normal role name, and that it is possible to replace this composition with a normal role name r in $\mathcal{O}$ when reintroducing it in $\mathbf{M}^-(q_r)$ via mappings $r_0(x, y) \wedge r_0(y, z) \rightarrow r(x, z)$ where q_r is an infinitary UCQ-rewriting of the OMQ Q mentioned above.

The 2ExpTime lower bound in Point 2 of Theorem 20 is proved similarly, using 2ExpTime-hardness of a different containment problem also studied in (Bienvenu et al. 2016).

Conclusion

We believe that several interesting questions remain. For example, our lower bounds only apply when the source query is a UCQ and it would be interesting to see whether the complexity drops when source queries are CQs. It would also be interesting to consider ontologies formulated in more expressive DLs such as $\mathcal{ALC}$. As a first observation in this direction, we note the following undecidability result, where $\mathcal{ALCF}$ is $\mathcal{ALC}$ extended with (globally) functional roles. It is proved by a reduction from the emptiness of AQs w.r.t. $\mathcal{ALCF}$-ontologies (Baader et al. 2016).

Theorem 23. In $[\mathcal{ALCF}, \text{GAV}]$, the AQ-to-$\mathcal{Q}$ expressibility and verification problems are undecidable for any $\mathcal{Q} \in \{\text{AQ}, \text{CQ}, \text{UCQ}\}$.

Regarding the expressibility problem, we note that the realization $\mathbf{M}(q_s)$ identified by Theorem 6 does not use any symbols introduced by the ontology and, in fact, is also a realization regarding the empty ontology. It would be interesting to understand how to obtain realizations that make better use of the ontology and to study setups where it can be unavoidable to exploit the ontology in realizations. This is the case, for example, when source queries are formulated in Datalog, the ontology is formulated in (some extension of) $\mathcal{EL}$, and target queries are UCQs. Finally, we note that it would be natural to study maximally contained realizations instead of exact ones and to take into account constraints over the source databases.

Acknowledgments

We thank the anonymous reviewers for their helpful remarks. This research was supported by ERC consolidator grant 647289 CODA.

References

Afrati, F. N. 2011. Determinacy and query rewriting for conjunctive queries and views. *Theor. Comput. Sci.* 412(11):1005–1021.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The dl-lite family and relations. *J. Artif. Intell. Res.* 36:1–69.

Baader, F.; Bienvenu, M.; Lutz, C.; and Wolter, F. 2016. Query and predicate emptiness in ontology-based data access. *J. Artif. Intell. Res.* 56:1–59.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Beeri, C.; Levy, A. Y.; and Rousset, M. 1997. Rewriting queries using views in description logics. In *Proc. of PODS*, 99–108. ACM Press.

Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.* 39(4):33:1–33:44.

Bienvenu, M.; Hansen, P.; Lutz, C.; and Wolter, F. 2016. First order-rewritability and containment of conjunctive queries in Horn description logics. In *Proc. of IJCAI2016*, 965–971.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Vardi, M. Y. 2002. Lossless regular views. In *Proc. of PODS*, 247–258. ACM.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2012. View-based query answering in description logics: Semantics and complexity. *J. Comput. Syst. Sci.* 78(1):26–46.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 2000. Answering queries using views over description logics knowledge bases. In *Proc. of IAAI*, 386–391. AAAI Press / The MIT Press.

Cima, G. 2017. Preliminary results on ontology-based open data publishing. In *Proc. of DL*, volume 1879 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Duschka, O. M., and Genesereth, M. R. 1997. Answering recursive queries using views. In *Proc. of PODS*, 109–116. ACM Press.

Gottlob, G.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; Schwentick, T.; and Zakharyaschev, M. 2014. The price of query rewriting in ontology-based data access. *Artif. Intell.* 213:42–59.

Haase, P., and Motik, B. 2005. A mapping system for the integration of OWL-DL ontologies. In *Proc. of IHIS'05*, 9–16. ACM.

Jiménez-Ruiz, E.; Kharlamov, E.; Zheleznyakov, D.; Horrocks, I.; Pinkel, C.; Skjæveland, M. G.; Thorstensen, E.; and Mora, J. 2015. Bootox: Practical mapping of rdbs to owl 2. In *Proc. of ISWC*, volume 9367 of *LNCS*, 113–132. Springer.

Kharlamov, E.; Hovland, D.; Jiménez-Ruiz, E.; Lanti, D.; Lie, H.; Pinkel, C.; Rezk, M.; Skjæveland, M. G.; Thorstensen, E.; Xiao, G.; Zheleznyakov, D.; and Horrocks, I. 2015. Ontology based access to exploration data at statoil. In *Proc. of ISWC*, volume 9367 of *LNCS*, 93–112. Springer.

Krisnadhi, A., and Lutz, C. 2007. Data complexity in the *EL* family of description logics. In *Proc. of LPAR*, volume 4790 of *LNCS*, 333–347. Springer.

Lembo, D.; Rosati, R.; Santarelli, V.; Savo, D. F.; and Thorstensen, E. 2017. Mapping repair in ontology-based data access evolving systems. In *Proc. of IJCAI*, 1160–1166. ijcai.org.

Levy, A. Y.; Mendelzon, A. O.; Sagiv, Y.; and Srivastava, D. 1995. Answering queries using views. In *Proc of PODS*, 95–104.

Lutz, C. 2008. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR2008*, volume 5195 of *LNCS*, 179–193. Springer.

Nash, A.; Segoufin, L.; and Vianu, V. 2010. Views and queries: Determinacy and rewriting. *ACM Trans. Database Syst.* 35(3):21:1–21:41.

Pinkel, C.; Binnig, C.; Jiménez-Ruiz, E.; Kharlamov, E.; May, W.; Nikolov, A.; Bastinos, A. S.; Skjæveland, M. G.; Solimando, A.; Taheriyan, M.; Heupel, C.; and Horrocks, I. 2018. RODI: benchmarking relational-to-ontology mapping generation quality. *Semantic Web* 9(1):25–52.

Poggi, A.; Lembo, D.; Calvanese, D.; Giacomo, G. D.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *Journal on Data Semantics* 10:133–173.

Sequeda, J. F., and Miranker, D. P. 2017. A pay-as-you-go methodology for ontology-based data access. *IEEE Internet Computing* 21(2):92–96.

Trisolini, S.; Lenzerini, M.; and Nardi, D. 1999. Data integration and warehousing in telecom italia. In *Proc. of SIGMOD*, 538–539. ACM Press.

Appendix

Proof of Lemma 4

Lemma 4. Let $\mathbf{M}$ be a set of GAV mappings, q, q_1 and q_2 UCQs over $\mathbf{S}$ and r, r_1 and r_2 UCQs over $\mathsf{sch}(\mathbf{M})$. Then:

1. If $q_1 \subseteq_{\mathbf{S}} q_2$, then $\mathbf{M}(q_1) \subseteq_{\mathsf{sch}(\mathbf{M})} \mathbf{M}(q_2)$.
2. If $r_1 \subseteq_{\mathsf{sch}(\mathbf{M})} r_2$, then $\mathbf{M}^-(r_1) \subseteq_{\mathbf{S}} \mathbf{M}^-(r_2)$.
3. $q \subseteq_{\mathbf{S}} \mathbf{M}^-(r)$ iff $\mathbf{M}(q) \subseteq_{\mathsf{sch}(\mathbf{M})} r$.

Proof. We prove all statements for CQs, it is straightforward to generalize to UCQs: one only needs to employ the characterization of UCQ in terms of homomorphisms instead of the one for CQs.

1. Since $q_1 \subseteq q_2$, we have $q_2 \rightarrow q_1$. Let h be a homomorphism witnessing this. It can be verified that restricting h to the variables in the CQ $\mathbf{M}(q_2)$[3] yields a homomorphism from CQ $\mathbf{M}(q_2)$ to CQ $\mathbf{M}(q_1)$, thus $\mathbf{M}(q_1) \subseteq_{\mathsf{sch}(\mathbf{M})} \mathbf{M}(q_2)$. In fact, let $R(\mathbf{x})$ be a relational atom in $\mathbf{M}(q_2)$. Then $R(\mathbf{x})$ was produced by some mapping $\varphi(\mathbf{y}) \rightarrow R(\mathbf{z}) \in \mathbf{M}$ and homomorphism h' from $\varphi(\mathbf{y})$ to q_2 with $h'(\mathbf{z}) = \mathbf{x}$. Composing h' with h enables an application of the same mapping in q_1 that delivers $R(h(\mathbf{x})) \in \mathbf{M}(q_1)$, as required. Moreover, every equality atom $x = y$ in $\mathbf{M}(q_2)$ must also be in q_2 by construction of $\mathbf{M}(q_2)$. Thus $h(x) = h(y) \in q_1$ or $h(x) = h(y)$. In the former case, $h(x) = h(y) \in \mathbf{M}(q_1)$ by construction of $\mathbf{M}(q_1)$.
2. From $r_1 \subseteq_{\mathsf{sch}(\mathbf{M})} r_2$, we obtain $r_2 \rightarrow r_1$. Let h be a witnessing homomorphism. We need to show that for each CQ p_1 in the UCQ $\mathbf{M}^-(r_1)$, there is a CQ p_2 in the UCQ $\mathbf{M}^-(r_2)$ such that $p_2 \rightarrow p_1$. Thus let p_1 be from $\mathbf{M}^-(r_1)$. By construction of $\mathbf{M}^-(r_1)$, p_1 is obtained from r_1 by choosing a mapping from $\mathbf{M}$ for each relational atom in r_1, 'applying it backwards', and then readding all equality atoms. We identify p_2 by choosing for every atom $R(\mathbf{y})$ of r_2 the mapping chosen for $R(h(\mathbf{y})) \in r_1$ in the construction of p_1. We can then straightforwardly define a homomorphism h' from p_2 to p_1 by extending h to the fresh variables in p_2.
3. For the "only if" direction, assume $p \rightarrow q$ for some CQ p in the UCQ $\mathbf{M}^-(r)$ and let h be a witnessing homomorphism. Let h' be the restriction of h to the variables in r. It can be verified that h' is a homomorphism from r to $\mathbf{M}(q)$. In fact, let $R(\mathbf{x})$ be a relational atom in r. Then by construction of $\mathbf{M}^-(r)$ there is a mapping $\varphi(\mathbf{y}) \rightarrow R(\mathbf{z}) \in \mathbf{M}$ that was 'applied backwards' in the construction of p and thus there is a homomorphism g from $\varphi(\mathbf{y})$ to $\mathbf{M}^-(r)$ with $g(\mathbf{z}) = \mathbf{x}$. Composing g with h enables an application of the same mapping in q that delivers $R(h(\mathbf{x})) \in \mathbf{M}(q)$, as required. The equality atoms in r must also be satisfied since they are the same as in $\mathbf{M}^-(r)$.

   For the "if" direction, assume that there is a homomorphism h from r to $\mathbf{M}(q)$. We need to show that there is a CQ p in the UCQ $\mathbf{M}^-(r)$ such that $p \rightarrow q$. By construction, every atom $R(\mathbf{x})$ in $\mathbf{M}(q)$ is produced by some mapping $\varphi(\mathbf{y}) \rightarrow R(\mathbf{z}) \in \mathbf{M}$ and homomorphism g from $\varphi(\mathbf{y})$ to q with $g(\mathbf{z}) = \mathbf{x}$. We identify p by choosing for every atom $R(\mathbf{y})$ of r the mapping that produced $R(h(\mathbf{y})) \in \mathbf{M}(q)$. We can then straightforwardly define a homomorphism h' from p to q by extending h to the fresh variables in p.

❏

Lemma 24. Let $Q = (\mathcal{O}, \mathbf{S}, q)$ be an OMQ with $\mathcal{O}$ formulated in FO($=$), q a UCQ, and q_r an infinitary UCQ rewriting of Q. Then $(\mathcal{O}, \mathbf{S}, q_r) \subseteq_{\mathbf{S}} q_r$.

Proof. For brevity, let $Q' = (\mathcal{O}, \mathbf{S}, q_r)$. Take an $\mathbf{S}$-ABox $\mathcal{A}$ and an $\mathbf{a} \in \mathsf{adom}(\mathcal{A})$ such that $\mathbf{a} \in \mathsf{cert}_{Q'}(\mathcal{A})$. We show that $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$, which implies $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathcal{A})$ as desired since q_r is a rewriting of Q. Let $\mathcal{I}$ be a model of $\mathcal{A}$ and $\mathcal{O}$. We have to show that $\mathbf{a} \in \mathsf{ans}_q(\mathcal{I})$.

From $\mathbf{a} \in \mathsf{cert}_{Q'}(\mathcal{A})$, we obtain $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathcal{I})$. This clearly implies that there is an interpretation $\mathcal{I}_f$ that is obtained by restricting $\mathcal{I}$ to a finite subset of the domain and satisfies $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathcal{I}_f)$. Let $\mathcal{A}_{\mathcal{I}}$ be $\mathcal{I}_f$ viewed as an ABox, restricted to the symbols in $\mathbf{S}$. Since q_r uses only symbols from $\mathbf{S}$, $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathcal{A}_{\mathcal{I}})$. As q_r is a rewriting of Q, $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A}_{\mathcal{I}})$. Observe that $\mathcal{I}$ is a model of $\mathcal{O}$ and $\mathcal{A}_{\mathcal{I}}$, and thus it follows that $\mathbf{a} \in \mathsf{ans}_q(\mathcal{I})$, as required. ❏

Proof Details for DL-Lite

Lemma 9. Let $Q = (\mathcal{O}, \mathbf{S}, q)$ be an OMQ with $\mathcal{O}$ formulated in FO($=$) and q a UCQ. If Q has a UCQ-rewriting q_r in which all CQs are of size at most n, then the canonical UCQ-rewriting q_c of size n is also a rewriting of Q.

Proof. We show that $q_c \equiv_{\mathsf{sch}(\mathbf{M})} q_r$. Since q_r is a UCQ-rewriting of Q, it then follows that q_c is also a UCQ-rewriting of $\mathcal{O}$. We consider the two directions of the equivalence separately.

$q_c \supseteq_{\mathsf{sch}(\mathbf{M})} q_r$. This holds because every CQ q in q_r is also an element in q_c. In fact, $q(\mathbf{x})$ being in q_r means that it is of size at most n. Viewing q as an ABox and $\mathbf{x}$ as a candidate answer, we trivially have $\mathbf{x} \in \mathsf{ans}_{q_r}(q)$ and thus $\mathbf{x} \in \mathsf{cert}_Q(q)$ because q_r is a rewriting of Q. As a consequence, the pair $(q, \mathbf{x})$ gives rise to a CQ in q_c.

$q_c \subseteq_{\mathsf{sch}(\mathbf{M})} q_r$. Let q in q_c. We have to show that there is a CQ q' in q_r such that $q' \rightarrow q$. By definition of q_c, q is a pair $(\mathcal{A}, \mathbf{a})$ viewed as a CQ such that $\mathbf{a} \in \mathsf{cert}_Q(\mathcal{A})$. Because q_r is a rewriting of Q, it follows from the latter that $\mathbf{a} \in \mathsf{ans}_{q_r}(\mathcal{A})$. This means that there is some q' in q_r with a homomorphism h from q' to $\mathcal{A}$ that maps the answer variables of q' to $\mathbf{a}$. As q is just $\mathcal{A}$ with $\mathbf{a}$ as the answer variables, h shows $q' \rightarrow q$. ❏

Theorem 11. The CQ-to-CQ expressibility problem is Π^p_2-hard for GAV mappings and the empty ontology.

In preparation for the proof of Theorem 11, let us recall the standard representation of 3SAT as a constraint satisfaction problem (CSP) and sketch how this can be used to show that the CQ-to-CQ expressibility problem is NP-hard for GAV mappings and the empty ontology. Let $\varphi(y_1, \ldots, y_m)$

[3] Including all the answer variables, which are the answer variables from q_2, even when they do not occur in any atom of $\mathbf{M}(q_2)$.

be a propositional logic formula in 3CNF. Let $\mathbf{S}$ be the schema that consists of all ternary relation names $C_{u_1u_2u_3}$ with $u_1u_2u_3 \in \{\mathsf{n}, \mathsf{p}\}^3$. Every clause in φ can be viewed as a fact over signature S by letting the u_i represent the polarities of the variables in the clause and using the variables from φ as constants. For example, $\neg y_2 \vee y_1 \vee \neg y_3$ gives the fact $C_{\mathsf{n},\mathsf{p},\mathsf{n}}(y_2, y_1, y_3)$. Thus, φ can be viewed as a database D_φ. What's more, we can build a database D that is independent of φ and such that $D_\varphi \to D$ if and only if φ is satisfiable. In CSP, D is called the *template for 3SAT*. It is actually easy to find D: use two constants 0 and 1 that represent truth values and add the fact $C_{u_1u_2u_3}(t_1, t_2, t_3)$ if the truth assignment t_1, t_2, t_3 to the three variables of a clause with polarities $u_1u_2u_3$ makes the clause true. For example, $C_{\mathsf{n},\mathsf{p},\mathsf{n}}(t_1, t_2, t_3)$ is added for any $t_1t_2t_3 \in \{0, 1\}^3$ except 101.

How does this relate to the expressibility problem? Let q_s be D_φ viewed as a Boolean CQ and take the mappings $D_\varphi \to A(x)$ and $D \to A(x)$ where D_φ and D are viewed as CQs in which all variables are answer variables with x an arbitrary but fixed such variable. Then $\mathbf{M}^-(\mathbf{M}(q_s))$ is (equivalent to) $D_\varphi \vee D$ with D_φ and D viewed as a Boolean CQs and thus it remains to apply Theorem 6 where q_r is now simply $\mathbf{M}(q_s)$ since the ontology is empty.

We now lift this simple reduction to ∀∃-3SAT. Thus let

$$\varphi = \forall x_0 \cdots \forall x_n \exists y_0 \cdots \exists y_m \psi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

be a quantified Boolean formula with ψ in 3CNF. We construct an OBDA specification $(\emptyset, \mathbf{M}, \mathbf{S})$ with $\mathbf{M}$ a set of GAV mappings as well as a Boolean CQ q_s over schema $\mathbf{S}$ such that φ is true iff q_s is CQ-expressible in $(\emptyset, \mathbf{M}, \mathbf{S})$.

The universally quantified variables have a different status in the reduction as, unlike the existentially quantified variables, they are not represented by variables in q_s. For example, the clause $(y_1 \vee \neg x_0 \vee \neg y_1)$ gives rise to the atom $C_{\mathsf{p}\neg x_0\mathsf{n}}(y_1, y_1)$. This is compensated by constructing the mappings in $\mathbf{M}$ so that $\mathbf{M}^-(\mathbf{M}(q_s))$ is now essentially a disjunction of (q_s and) exponentially many versions of the template D_φ, one for every truth assignment to the universally quantified variables. For example, such a template includes $C_{\mathsf{p}\neg x_0\mathsf{n}}(t_1t_2)$ for *all* $t_1t_2 \in \{0, 1\}^2$ if it represents a truth assignment that makes x_0 true and otherwise it includes $C_{\mathsf{p}\neg x_0\mathsf{n}}(t_1t_2)$ for all $t_1t_2 \in \{0, 1\}^2$ except 01. To achieve this, we use binary relations in the head of mappings instead of unary ones. In fact, we want $\mathbf{M}(q_s)$ to be of the form $\mathbf{M}(q_s) = \bigwedge_{i=0}^n r_i(y_0, y_1)$ and there will be two ways to translate each $r_i(y_0, y_1)$ backwards in the construction of $\mathbf{M}^-(\mathbf{M}(q_s))$, corresponding to the two possible truth values of the universally quantified variable x_i.

We now make the reduction precise. Let $U = \{x_0, \ldots, x_n, \neg x_0, \ldots, \neg x_n, \mathsf{n}, \mathsf{p}\}$. For every triple $(u_1, u_2, u_3) \in U^3$ we include a relation $C_{u_1u_2u_3}$ in $\mathbf{S}$. The *arity* of $C_{u_1u_2u_3}$ is the number of positions in (u_1, u_2, u_3) that are n or p. Additionally, $\mathbf{S}$ contains a binary relation Z which helps us to achieve that $\mathbf{M}(q_s)$ is of the intended form even when q_s admits non-trivial automorphisms.

We define q_s to encode φ. The existentially quantified variables of q_s are $y_0, \ldots, y_m$. For every clause $\ell_1 \vee \ell_2 \vee \ell_3$ in ψ, we introduce an atom in q_s with the symbol $C_{u_1u_2u_3}$, where $u_i = \ell_i$ if ℓ_i contains a universally quantified variable, $u_i = \mathsf{p}$ if ℓ_i is a positive literal with an existentially quantified variable and $u_i = \mathsf{n}$ if ℓ_i is a negative literal with an existentially quantified variable. The variables that occur in this atom are the existentially qualified variables of the clause in the order of their appearance in the clause, see above for an example. Moreover, we add the atom $Z(y_0, y_1)$ to q_s, assuming w.l.o.g. that there are at least two existentially quantified variables in φ.

We now construct the GAV mappings in $\mathbf{M}$. For every universally quantified variable x_i of φ, we introduce three mappings with the same head $r_i(z_0, z_1)$:

1. In the first mapping $q_s'(z_0, z_1) \to r_i(z_0, z_1)$, the body is q_s with y_0 and y_1 renamed to z_0, z_1 (and all existential quantifiers removed).
2. The body of the second mapping $\tau_i^0(z_0, z_1) \to r_i(z_0, z_1)$ generates the part of the template that must be there when x_i is assigned truth value 0. The variables z_0 and z_1 represent the two elements of the template.

   The body τ_i^0 only contains the variables z_0 and z_1. For every $u_1u_2u_3 \in U^3$ and sequence $\overline{v} = v_1v_2\cdots$ over $\{z_0, z_1\}$ of the same length as the arity of $C_{u_1u_2u_3}$, we add the atom $C_{u_1u_2u_3}(\overline{v})$ to τ_i^0 if at least one of the following holds:

   (a) $\neg x_i$ is among u_1, u_2 and u_3,

   (b) $v_i = z_0$ and the i-th appearance of n or p in (u_1, u_2, u_3) is n for some i,

   (c) $v_i = z_1$ and the i-th appearance of n or p in (u_1, u_2, u_3) is p for some i.

   We also add the atoms $Z(z_0, z_0)$, $Z(z_0, z_1)$, $Z(z_1, z_0)$ and $Z(z_1, z_1)$ to τ_i^0.

   (For example in τ_3^0 we add the atoms $C_{\mathsf{p}x_3\mathsf{n}}(z_0, z_0)$, $C_{\mathsf{p}x_3\mathsf{n}}(z_1, z_0)$, and $C_{\mathsf{p}x_3\mathsf{n}}(z_1, z_1)$, but not $C_{\mathsf{p}x_3\mathsf{n}}(z_0, z_1)$. We add all $C_{\mathsf{p}\neg x_3\mathsf{n}}(z_0, z_0)$, $C_{\mathsf{p}\neg x_3\mathsf{n}}(z_0, z_1)$, $C_{\mathsf{p}\neg x_3\mathsf{n}}(z_1, z_0)$ and $C_{\mathsf{p}\neg x_3\mathsf{n}}(z_1, z_1)$. Also, we add $C_{x_2x_3\mathsf{p}}(z_1)$ but not $C_{x_2x_3\mathsf{p}}(z_0)$.)
3. The body $\tau_i^1(z_0, z_1)$ of the third mapping $\tau_i^1(z_0, z_1) \to r_i(z_0, z_1)$ is dual to $\tau_i^0(z_0, z_1)$ in that it encodes the case where x_i is true. This means the definition is as above with the difference that in Condition 2a, the literal x_i, and not $\neg x_i$, is required to be among u_1, u_2 and u_3.

Lemma 25. φ is true iff q_s is CQ-expressible in $(\emptyset, \mathbf{M}, \mathbf{S})$.

Proof. By Theorem 6, it suffices to show that φ is true iff $\mathbf{M}^-(\mathbf{M}(q_s)) \subseteq q_s$, that is, iff there is a homomorphism from q_s to every disjunct of $\mathbf{M}^-(\mathbf{M}(q_s))$.

We first describe the UCQ $\mathbf{M}^-(\mathbf{M}(q_s))$. First observe that $\mathbf{M}(q_s)$ is indeed $\bigwedge_{i=0}^n r_i(y_0, y_1)$: The mapping $q_s' \to r_i(z_0, z_1)$ has a match at (y_0, y_1) for every i and it has no other matches since $Z(z_0, z_1)$ is in q_s' and $Z(y_0, y_1)$ is the only atom in q_s the contains Z. The mappings $\tau_i^v(z_0, z_1) \to r_i(z_0, z_1)$ do not match anywhere in q_s for $v \in \{0, 1\}$, since the atoms $Z(z_0, z_0)$, $Z(z_0, z_1)$, $Z(z_1, z_0)$ and $Z(z_1, z_1)$ all appear in the body of these mappings. There are 3^{n+1} disjuncts in $\mathbf{M}^-(\mathbf{M}(q_s))$, one for every combination of $n + 1$

choices of the three different mappings with head $r_i(z_0, z_1)$, for every $i \in \{0, \dots, n\}$. We now prove the lemma.

"⇒". Assume that φ is true. We want to show that there is a homomorphism from q_s into every CQ in $\mathbf{M}^-(\mathbf{M}(q_s))$. Pick an arbitrary CQ p in $\mathbf{M}^-(\mathbf{M}(q_s))$. Such a disjunct corresponds of a choice of one of the bodies q'_s, τ^0_i, or τ^1_i for each $i \in \{0, \dots, n\}$.

First consider the case where for some i we choose q'_s. In that case p contains an isomorphic copy of q_s and thus we are done.

Now consider the case for no i we choose a mapping with body q'_s, that is, for every i we choose τ^0_i or τ^1_i. This corresponds to an assignment t of the truth values 0 and 1 to the variables $x_0, \dots, x_n$. Because $\varphi = \forall x_0, \dots, x_n \exists y_0, \dots, y_m \psi$ is true we can extend t to an assignment for $x_0, \dots, x_n, y_0, \dots, y_m$ that makes ψ true. We define the homomorphism h from q_s to p such that $h(y_j) = y_{t(y_j)}$ for all $j \in \{0, \dots, m\}$. We need to verify that h is a homomorphism. All atoms with the symbol Z are preserved as by definition of the τ^0_i and τ^1_i, there is a Z atom in p for any pair over $\{y_0, y_1\}$. Consider then any atom $C_{u_1u_2u_3}(\overline{y})$ from q_s. There is a corresponding clause $\ell_1 \vee \ell_2 \vee \ell_3$ in ψ that is true under the assignment t. It follows that one of the literals ℓ_1, ℓ_2, ℓ_3 is true under t. We make a case distinction:

If $\ell_k = x_i$ is a universally quantified variable, then $t(x_i) = 1$ and hence p contains τ^1_i. By (the implicit) Condition 3a, from the definition of $\mathbf{M}$, it follows that τ^1_i contains the atom $C_{u_1u_2u_3}(y_{t(\overline{y})})$.

In the case where $\ell_k = \neg x_i$, we use the same argument and Condition 2a.

If $\ell_k = y_j$ is an existentially quantified variable, then $u_k = \mathsf{p}$ and $t(y_j) = 1$. Hence, $h(y_j) = y_1$ and from Condition (2c) or (3c), the atom $C_{u_1u_2u_3}$ is in τ^0_0 and in τ^1_0, thus in p (since we can assume w.l.o.g. that there is at least one universally quantified variable).

The case where $\ell_k = \neg y_j$ is analogous, using Conditions (2b) or (3b).

"⇐". Assume that there is a homomorphism from q_s into every disjunct of $\mathbf{M}^-(\mathbf{M}(q_s))$. We want to show that φ is true. So take any assignment t for $x_0, \dots, x_n$. We need to extend t to an assignment t' for $x_0, \dots, x_n, y_0, \dots, y_m$ that makes ψ true. Consider the disjunct p_t of $\mathbf{M}^-(\mathbf{M}(q_s))$ that arises from choosing the mapping $\tau^{t(x_i)}_i(z_0, z_1) \rightarrow r_i(z_0, z_1)$ for each $i = 0, \dots, n$. By assumption, there is a homomorphism h from q_s to p_t. Clearly, p_t contains only the variables y_0 and y_1. We define t' such that $t'(x_i) = t(x_i)$, $t'(y_j) = 0$ if $h(y_j) = y_0$, and $t'(y_j) = 1$ if $h(y_j) = y_1$.

It remains to be shown that ψ is true under t'. Take an arbitrary clause $\ell_1 \vee \ell_2 \vee \ell_3$ in ψ. Consider the corresponding atom $C_{u_1u_2u_3}(\overline{y})$ in q_s. As h is a homomorphism, $C_{u_1u_2u_3}(h(\overline{y}))$ is in p_t. By the definition of p_t this means that $C_{u_1u_2u_3}(h(\overline{y}))$ is contained in $\tau^{t(x_i)}_i$ for some i. Hence one of the conditions (a), (b) or (c) from Point 2 or 3 of the construction of $\mathbf{M}$ is satisfied.

If (a) is satisfied and $t(x_i) = 0$, then $\neg x_i$ is among u_1, u_2, u_3. It follows that then $\neg x_i$ is a literal in $\ell_1 \vee \ell_2 \vee \ell_3$ and hence the clause is true under t' because $t'(x_i) = t(x_i)$. In case (a) is satisfied and $t(x_i) = 1$, we can reason analogously.

If (b) is satisfied, then $h(y_i) = y_0$ for some y_i in $\overline{y}$, and the i-th appearance of either n or p in (u_1, u_2, u_3) is n. Hence $\neg y_i$ is a literal in $\ell_1 \vee \ell_2 \vee \ell_3$ which makes the clause true because $t'(y_1) = 0$ since $h(y_i) = y_0$. We reason analogously in the case where (c) is satisfied. ❑

Preliminary: Two-way alternating parity automata (TWAPA)

We introduce two-way alternating parity automata on finite trees (TWAPAs).

A *tree* is a non-empty (and potentially infinite) set $T \subseteq \mathbb{N}^*$ closed under prefixes. We say that T is *m-ary* if $T \subseteq \{1, \dots, m\}^*$. For an alphabet Γ, a *Γ-labeled tree* is a pair (T, L) with T a tree and $L : T \rightarrow \Gamma$ a node labeling function.

For any set X, let $\mathcal{B}^+(X)$ denote the set of all positive Boolean formulas over X, i.e., formulas built using conjunction and disjunction over the elements of X used as propositional variables, and where the special formulas true and false are allowed as well. An *infinite path P* of a tree T is a prefix-closed set $P \subseteq T$ such that for every $i \geq 0$, there is a unique $x \in P$ with $|x| = i$.

Definition 26 (TWAPA)**.** A *two-way alternating parity automaton (TWAPA) on finite m-ary trees* is a tuple $\mathfrak{A} = (S, \Gamma, \delta, s_0, c)$ where S is a finite set of *states*, Γ is a finite alphabet, $\delta : S \times \Gamma \rightarrow \mathcal{B}^+(\mathsf{tran}(\mathfrak{A}))$ is the *transition function* with $\mathsf{tran}(\mathfrak{A}) = \{\langle i \rangle s,\ [i]s \mid -1 \leq i \leq m \text{ and } s \in S\}$ the set of *transitions* of $\mathfrak{A}$, $s_0 \in S$ is the *initial state*, and $c : S \rightarrow \mathbb{N}$ is the *parity condition* that assigns to each state a *priority*.

Intuitively, a transition $\langle i \rangle s$ with $i > 0$ means that a copy of the automaton in state s is sent to the i-th successor of the current node, which is then required to exist. Similarly, $\langle 0 \rangle s$ means that the automaton stays at the current node and switches to state s, and $\langle -1 \rangle s$ indicates moving to the predecessor of the current node, which is then required to exist. Transitions $[i]s$ mean that a copy of the automaton in state s is sent on the relevant successor if that successor exists (which is not required).

Definition 27 (Run, Acceptance)**.** A *run* of a TWAPA $\mathfrak{A} = (S, \Gamma, \delta, s_0, c)$ on a finite Γ-labeled tree (T, L) is a $T \times S$-labeled tree (T_r, r) such that the following conditions are satisfied:

1. $r(\varepsilon) = (\varepsilon, s_0)$;
2. if $y \in T_r$, $r(y) = (x, s)$, and $\delta(s, L(x)) = \varphi$, then there is a (possibly empty) set $S \subseteq \mathsf{tran}(\mathfrak{A})$ such that S (viewed as a propositional valuation) satisfies φ as well as the following conditions:
   (a) if $\langle i \rangle s' \in S$, then $x \cdot i \in T$ and there is a node $y \cdot j \in T_r$ such that $r(y \cdot j) = (x \cdot i, s')$;
   (b) if $[i]s' \in S$ and $x \cdot i \in T$, then there is a node $y \cdot j \in T_r$ such that $r(y \cdot j) = (x \cdot i, s')$.

We say that (T_r, r) is *accepting* if on all infinite paths $\varepsilon = y_1 y_2 \cdots$ of T_r, the maximum priority that appears infinitely often is even. A finite Γ-labeled tree (T, L) is *accepted* by $\mathfrak{A}$ if there is an accepting run of $\mathfrak{A}$ on (T, L). We use $L(\mathfrak{A})$ to denote the set of all finite Γ-labeled tree accepted by $\mathfrak{A}$.

It is known (and easy to see) that TWAPAs are closed under complementation and intersection, and that these constructions involve only a polynomial blowup. It is also known that their emptiness problem can be solved in time single exponential in the number of states and polynomial in all other components of the automaton. In what follows, we shall generally only explicitly analyze the number of states of a TWAPA, but only implicitly take care that all other components are of the allowed size for the complexity result that we aim to obtain.

Preliminary: Universal models

We introduce the universal model $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ of an ABox $\mathcal{A}$ and an ontology $\mathcal{O}$ in $\mathcal{ELHI}$. The main properties of $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ are:

- $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ is a model of $\mathcal{A}$ and $\mathcal{O}$;
- for every model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$ there exists a homomorphism from $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ to $\mathcal{I}$ that maps each $a \in \mathsf{adom}(\mathcal{A})$ to itself.

$\mathcal{I}_{\mathcal{A},\mathcal{O}}$ is constructed using a standard chase procedure. We assume that $\mathcal{O}$ is in normal form.

We start by defining the universal model $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ of $\mathcal{A}$ and $\mathcal{O}$. It is convenient to use ABox notation when constructing $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ and so we will construct a (possibly infinite) ABox $\mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$ and define $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ as the interpretation corresponding to $\mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$.

Thus assume that $\mathcal{A}$ and $\mathcal{O}$ are given. The *full completion sequence of $\mathcal{A}$ w.r.t. $\mathcal{O}$* is the sequence of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \dots$ defined by setting $\mathcal{A}_0 = \mathcal{A}$ and defining $\mathcal{A}_{i+1}$ to be $\mathcal{A}_i$ extended as follows (recall that we abbreviate $r(a, b)$ by $r^-(b, a)$ and that r ranges over roles):

(i) If $\exists r.B \sqsubseteq A \in \mathcal{O}$ and $r(a, b), B(b) \in \mathcal{A}_i$, then add $A(a)$ to $\mathcal{A}_{i+1}$;

(ii) if $\top \sqsubseteq A \in \mathcal{O}$ and $a \in \mathsf{adom}(\mathcal{A}_i)$, then add $A(a)$ to $\mathcal{A}_{i+1}$;

(iii) if $B_1 \sqcap B_2 \sqsubseteq A \in \mathcal{O}$ and $B_1(a), B_2(a) \in \mathcal{A}_i$, then add $A(a)$ to $\mathcal{A}_{i+1}$;

(iv) if $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $A(a) \in \mathcal{A}_i$ then take a fresh individual b and add $r(a, b)$ and $B(b)$ to $\mathcal{A}_{i+1}$;

(v) if $r \sqsubseteq s \in \mathcal{O}$ and $r(a, b) \in \mathcal{A}_i$, then add $s(a, b)$ to $\mathcal{A}_{i+1}$.

Now let $\mathcal{A}^{\mathsf{uni}}_{\mathcal{O}} = \bigcup_{i \geq 0} \mathcal{A}_i$ and let $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ be the interpretation corresponding to $\mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$. It is straightforward to prove the following properties of $\mathcal{I}_{\mathcal{A},\mathcal{O}}$.

Lemma 28. Assume $\mathcal{O}$ is in normal form. Then

- $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ is a model of $\mathcal{A}$ and $\mathcal{O}$;
- for every model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$ there exists a homomorphism from $\mathcal{I}_{\mathcal{A},\mathcal{O}}$ to $\mathcal{I}$ that maps each $a \in \mathsf{adom}(\mathcal{A})$ to itself.

Note that the ABox $\mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$ can contain additional constants and can even be infinite.

The proof of the following is straightforward.

Lemma 29. For all facts $A(a)$ and $r(a, b)$ with $a, b \in \mathsf{adom}(\mathcal{A})$:

- $\mathcal{A}, \mathcal{T} \models A(a)$ iff $A(a) \in \mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$;
- $\mathcal{A}, \mathcal{T} \models r(a, b)$ iff $r(a, b) \in \mathcal{A}^{\mathsf{uni}}_{\mathcal{O}}$.

Preliminary: Derivation Trees

In the next sections we use TWAPA to obtain ExpTime-decision procedures for entailment of atomic queries. The construction of these automata relies on a characterization of entailment of AQs in term of derivation trees.

Fix an $\mathcal{ELHI}$ ontology $\mathcal{O}$ in normal form and an ABox $\mathcal{A}$. A *derivation tree* for a fact $A_0(a_0)$ in $\mathcal{A}$, with $A_0 \in \mathsf{N_C}$ and $a_0 \in \mathsf{adom}(\mathcal{A})$, is a finite $\mathsf{adom}(\mathcal{A}) \times \mathsf{N_C}$-labeled tree (T, V) that satisfies the following conditions:

1. $V(\varepsilon) = (a_0, A_0)$;
2. if $V(x) = (a, A)$ and neither $A(a) \notin \mathcal{A}$ nor $\top \sqsubseteq A \in \mathcal{O}$, then one of the following holds:
    - x has successors $y_1, \dots, y_k$, $k \geq 1$ with $V(y_i) = (a, B_i)$ for $1 \leq i \leq k$ and $\mathcal{O} \models B_1 \sqcap \dots \sqcap B_k \sqsubseteq A$;
    - x has a single successor y with $V(y) = (b, B)$ and there is an $\exists s.B \sqsubseteq A \in \mathcal{O}$ and an $r(a, b) \in \mathcal{A}$ such that $\mathcal{O} \models r \sqsubseteq s$.

Note that the first item of Point 2 above requires $\mathcal{O} \models A_1 \sqcap \dots \sqcap A_n \sqsubseteq A$ instead of $A_1 \sqcap A_2 \sqsubseteq A \in \mathcal{O}$ to 'shortcut' anonymous parts of the universal model. In fact, the derivation of A from $A_1 \sqcap \dots \sqcap A_n$ by $\mathcal{O}$ can involve the introduction of anonymous elements.

Lemma 30. Let $A \in \mathsf{N_C}$ and let a be a constant in $\mathcal{A}$. We have $\mathcal{A}, \mathcal{O} \models A(a)$ iff there is a derivation tree for $A(a)$ in $\mathcal{A}$.

The proof is a straightforward variation of an analogous result for the stronger logic $\mathcal{ELIHF}_{\bot}^{\sqcap-\mathsf{lhs}}$ in (Bienvenu et al. 2016). Details are omitted.

Expressibility in $\mathcal{ELHI}$: Upper Bound for Rooted Queries, Missing Details

Lemma 14. The algorithm decides the complement of the problem in Theorem 13.

Proof. As explained in the main text, the soundness part is easy.

For the completeness direction assume that $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$ is false. This means there is an ABox $\mathcal{A}$ and tuple $\mathbf{a}$ such that $(\mathcal{A}, \mathbf{a})$ is in the canonical rewriting q_r of Q and there is p in $\mathbf{M}^-(\mathcal{A}, \mathbf{a})$ such that for all q in q_s we have $q \not\to p$. We show how to obtain an ABox $\mathcal{A}''$ that the algorithm can choose in step 1 in order to accept.

By Lemma 15 there exists an pseudo tree-shaped ABox $\mathcal{A}'$ of core-size $|q_t|$ and branching degree at most $|\mathcal{O}|$ and tuple $\mathbf{a}'$ in the core of $\mathcal{A}'$ such that $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}')$ and there is a homomorphism $(\mathcal{A}', \mathbf{a}') \to (\mathcal{A}, \mathbf{a})$.

Define $\mathcal{A}''$ to be obtained from the pseudo tree-shaped ABox $\mathcal{A}'$ by removing all facts that contain at least one constant that has a distance larger than $|q_s|$ from the core. Furthermore, we add all facts $A(b)$ and $r(b, b)$ for all A and r

in $\mathsf{sch}(\mathbf{M})$ and for all constants b that have exactly distance $|q_s|$ from the core.

The resulting size of $\mathcal{A}''$ is at most $|q_t| + |q_t| \cdot |\mathcal{O}|^{|q_s|}$, so $\mathcal{A}''$ can be chosen in Step 1.

Claim: Step 2 succeeds, that is, $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}'')$.

Proof of claim: Define a homomorphism $h : \mathcal{A}' \to \mathcal{A}''$ that is the identity on constants of distance at most $|q_s|$ from the core and maps a constant b of distance more than $|q_s|$ from the core to the unique b' of distance precisely $|q_s|$ such that b is in the subtree rooted at b'. This indeed defines a homomorphism because in $\mathcal{A}''$ all symbols in $\mathsf{sch}(\mathbf{M})$ are true at each constant with distance $|q_s|$. Now $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}'')$ follows from $\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{A}')$ because h is a homomorphism that fixes the tuple $\mathbf{a}'$.

We then describe the query p' in $\mathbf{M}^-(\mathcal{A}'')$ that can be chosen in step 3. For every fact in $\mathcal{A}''$ that is also in $\mathcal{A}'$ choose the same mapping that was chosen to obtain p from $\mathcal{A}$. For all other facts we choose an arbitrary suitable mapping.

Claim: Step 4 succeeds, that is, $q \not\to p'$ for all q in q_s.

Proof of claim: Assume towards contradiction that there is a homomorphism $g' : q \to p'$ for some q in q_s. Because $\mathcal{A}''$ and $\mathcal{A}'$ are equal when restricted to variables of distance less than n from $\mathbf{a}'$ it follows that p and p' are equal when restricted to variables of distance less than n from $\mathbf{a}'$. This is due to the fact that the distance of two variables from $\mathcal{A}$ cannot decrease in p in $\mathbf{M}^-(\mathcal{A})$, and similarly for $\mathcal{A}''$ and p'.

Because $|q| \leq |q_s|$ and q is rooted it follows that all constants in the image of $g' : q \to p'$ have distance less than $|q_s|$ from $\mathbf{a}$ in p. Since $\mathbf{a}$ lies in the core, these constants have also distance less than $|q_s|$ from the core. Because p and p' are equal when restricted to constants of distance less than $|q_s|$ from the core, there is a homomorphism $g : q \to p$, which contradicts our assumption on p.

❏

Verification in $\mathcal{ELHI}$: Upper Bound, Missing Details

We describe the construction of the TWAPA $\mathfrak{A}^\lambda$. Recall that $\mathfrak{A}^\lambda$ will get as input (an encoding of) a triple $(\mathcal{A}, \mathbf{a}, p)$ consisting of a pseudo tree ABox $\mathcal{A}$ with a core $\mathcal{C}$ of size at most $|q_t|$, a tuple $\mathbf{a}$ of the same arity as q_t and a CQ $p \in \mathbf{M}^-(\mathcal{A})$, while λ is a homomorphism pattern that maps every variable y in q_t to a tuple $(c, o) \in \mathsf{adom}(\mathcal{C}) \times \{\mathsf{core}, \mathsf{subtree}\}$. It will accept this triple iff

1. there is a homomorphism h from q_t to the universal model of $\mathcal{A}$ and $\mathcal{O}$ that maps answer variables of q_t to $\mathbf{a}$ and $\lambda(y) = (h(y), \mathsf{core})$ if $h(y) \in \mathsf{C}_{\mathsf{core}}$ and $\lambda(y) = (c, \mathsf{subtree})$, if $h(y)$ maps into a fresh subtree in the universal model of $\mathcal{A}$ and $\mathcal{O}$ whose root is $c \in \mathsf{C}_{\mathsf{core}}$ and
2. $p \to q$.

It is easy to see that Condition 1 holds for some λ if and only if Condition $2'$ from the definition of the UCQ-rewriting q_r in the main part of the paper holds. Recall that Condition $2'$ is the following requirement:

$$\mathbf{a}' \in \mathsf{cert}_Q(\mathcal{C}' \cup \{A(a) \mid \mathcal{A}', \mathcal{O} \models A(a),\ a \in \mathsf{dom}(\mathcal{C}')\}).$$

Hence Condition 1 guarantees that $(\mathcal{A}, \mathbf{a})$ is in the rewriting q_r. Together with the second point and the conditions on the encoded tuples $(\mathcal{A}, \mathbf{a}, p)$ this then means that there is a λ such that $\mathfrak{A}^\lambda$ accepts a tree encoding $(\mathcal{A}, \mathbf{a}, p)$ iff p is a disjunct in $\mathbf{M}^-(q_r)$ such that $p \to q$.

The automaton $\mathfrak{A}^\lambda$ is constructed as the intersection of three automata $\mathfrak{A}_{\mathsf{proper}}$, $\mathfrak{A}_1^\lambda$ and $\mathfrak{A}_2$, where the first checks if the input tree describes a proper tree, $\mathfrak{A}_1^\lambda$ accepts trees that encode a triple that satisfies Condition 1 and $\mathfrak{A}_2$ accepts trees that encode a triple that satisfies Condition 2. It is easy to define the automaton $\mathfrak{A}_{\mathsf{proper}}$, we leave out the details.

Definition of $\mathfrak{A}_1^\lambda$. To define $\mathfrak{A}_1^\lambda$, we first introduce some notation. Let ROL be the set of roles that appear in $\mathcal{O}$ or $\mathsf{sch}(\mathbf{M})$. Let CN the set of concept names that appear in $\mathcal{O}$ or $\mathsf{sch}(\mathbf{M})$ and let $\mathsf{tp} = 2^{\mathsf{CN}}$. Let U be the set of all partial functions from the variables in q_t to the set $\{\mathsf{core}, \mathsf{subtree}\}$. We define a relation $R \subseteq \mathsf{tp} \times U$ such that $(t, f) \in R$ iff there is a homomorphism g from q_t restricted to variables in the domain of f to the universal model of $\mathcal{O}$ and $\{A(a) \mid A \in t\}$ such that $g(y) = a$ iff $f(y) = \mathsf{core}$. The relation R can be computed in exponential time: Given a pair (t, f), compute the universal model of $\{A(a) \mid A \in t\}$ and $\mathcal{O}$ up to depth $|q_t|$ and check for the existence of a suitable homomorphism g using brute force.

Let $\mathfrak{A}_1 = (S, \delta, \Sigma_\varepsilon \uplus \Sigma_N, s_0, c)$ where

$$\begin{aligned} S = \{s_0\} &\uplus \{s_b^A \mid A \in \mathsf{CN} \text{ and } b \in \mathsf{C}_{\mathsf{core}}\} \\ &\uplus \{s_{r,b}^A \mid A \in \mathsf{CN} \text{ and } r \in \mathsf{ROL} \text{ and } b \in \mathsf{C}_{\mathsf{core}}\} \\ &\uplus \{s_r^A \mid A \in \mathsf{CN} \text{ and } r \in \mathsf{ROL}\} \\ &\uplus \{s^A \mid A \in \mathsf{CN}\} \end{aligned}$$

and $c(s) = 1$ for every $s \in S$, i.e. precisely the finite runs are accepting. All states besides s_0 are used to check whether a certain fact $A(a)$ is entailed in the universal model of $\mathcal{A}$ and $\mathcal{O}$. Following Lemma 30, this can be done by checking the existence of an appropriate derivation tree. The state s_b^A checks whether $\mathcal{A}, \mathcal{O} \models A(b)$. The state $s_{r,b}^A$ checks whether we are in an r-child d of b such that $\mathcal{A}, \mathcal{O} \models A(d)$. The state s_r^A checks whether the current node d is an r-child such that $\mathcal{A}, \mathcal{O} \models A(d)$. The state s^A checks whether for the current node d we have $\mathcal{A}, \mathcal{O} \models A(d)$.

We now describe the definition of the transition function δ. To define $\delta(s_0, l)$, where $l = (\mathcal{B}, \mathbf{a}, \mu) \in \Sigma_\varepsilon$ we distinguish cases depending on whether the $\mathcal{B}$ and the $\mathbf{a}$ encoded in l is compatible with the homomorphism pattern λ. That is we check whether the following two conditions are fulfilled:

- λ maps the i-th answer variable x_i from q_t to (a_i, core), where a_i is the i-the element of $\mathbf{a}$.
- For every role atom $r(z_1, z_2)$ in q_t such that $\lambda(z_i) = (b_i, \mathsf{core})$ we have that $r(b_1, b_2) \in \mathcal{B}$.

If these conditions are not fulfilled then we set $\delta(s_0, l) = \mathsf{false}$, meaning that the automaton $\mathfrak{A}_1^\lambda$ immediately rejects the input tree. If these conditions are fulfilled then define $\delta(s_0, l)$ to be:

$$\bigwedge_{\substack{z \in \mathsf{var}(q_t) \\ \lambda(z)=(b,\mathsf{core})}} \bigwedge_{A(z) \in q_t} \langle 0 \rangle s_b^A \quad \wedge \tag{1}$$

$$\bigwedge_{\substack{Z \subseteq \mathsf{var}(q_t) \\ Z=\lambda^{-1}(\{b\}\times\{\mathsf{core},\mathsf{subtree}\}) \\ Z \neq \emptyset}} \bigvee_{\substack{t \in \mathsf{tp} \\ (t,f) \in R \\ f=\pi_2 \circ \lambda|_Z}} \bigwedge_{A \in t} \langle 0 \rangle s_b^A \tag{2}$$

Here the $\lambda|_Z$ denotes the restriction of λ to the variables in Z and π_2 denotes the projection to the second component. The conjunction in the first line makes sure that the concept names needed for variables of q_t that are mapped to $\mathsf{C}_{\mathsf{core}}$ are derived. The second line assures that for all variables of q_t that are mapped to a fresh subtree generated in the universal model of $\mathcal{A}$ and $\mathcal{O}$, there is actually a type t derived at the root $b \in \mathsf{C}_{\mathsf{core}}$ that generates a suitable tree.

The following transitions are then used to check for derivations of concept names.

For $l \in \Sigma_\varepsilon$, let:

$$\delta(s_b^A, l) = \bigvee_{\mathcal{O} \models B_1 \sqcap \cdots \sqcap B_n \sqsubseteq A} \langle 0 \rangle q_b^{B_1} \wedge \ldots \wedge \langle 0 \rangle s_b^{B_n} \quad \vee$$

$$\bigvee_{\substack{\exists s.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models r \sqsubseteq s}} \bigvee_{\substack{b' \in \mathsf{C}_{\mathsf{core}} \\ r(b,b') \in \mathcal{B}}} \langle 0 \rangle s_{b'}^B \quad \vee$$

$$\bigvee_{\substack{\exists s.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models r \sqsubseteq s}} \bigvee_{1 \leq i \leq |\mathcal{O}| \cdot |q_t|} \langle i \rangle s_{r,b}^B$$

For $l \in \Sigma_N$ and $\{r, b\} \subseteq l$ let:

$$\delta(s_{r,b}^A, l) = \langle 0 \rangle s^A.$$

For $l \in \Sigma_N$ and $r \in l$ let

$$\delta(s_r^A, l) = \langle 0 \rangle s^A.$$

For $l \in \Sigma_N$ with role $r \in l$ such that l contains no constant of $\mathsf{C}_{\mathsf{core}}$, let:

$$\delta(s^A, l) = \bigvee_{\mathcal{O} \models B_1 \sqcap \cdots \sqcap B_n \sqsubseteq A} \langle 0 \rangle s^{B_1} \wedge \ldots \wedge \langle 0 \rangle s^{B_n} \quad \vee$$

$$\bigvee_{\substack{\exists s^-.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models r \sqsubseteq s}} \langle -1 \rangle s^B \quad \vee$$

$$\bigvee_{\substack{\exists s.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models u \sqsubseteq s}} \langle 1 \rangle s_u^B \vee \ldots \vee \langle |\mathcal{O}| \rangle s_u^B$$

For $l \in \Sigma_N$ with $\{r, b\} \subseteq l$ for a role r and $b \in \mathsf{C}_{\mathsf{core}}$, let:

$$\delta(s^A, l) = \bigvee_{\mathcal{O} \models B_1 \sqcap \cdots \sqcap B_n \sqsubseteq A} \langle 0 \rangle s^{B_1} \wedge \ldots \wedge \langle 0 \rangle s^{B_n} \quad \vee$$

$$\bigvee_{\substack{\exists s^-.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models r \sqsubseteq s}} \langle -1 \rangle s_b^B \quad \vee$$

$$\bigvee_{\substack{\exists s.B \sqsubseteq A \in \mathcal{O} \\ \mathcal{O} \models u \sqsubseteq s}} \langle 1 \rangle s_u^B \vee \ldots \vee \langle |\mathcal{O}| \rangle s_u^B$$

All pairs $(s, l) \in S \times \Sigma_\varepsilon \uplus \Sigma_N$ that have not been mentioned yet will never occur in a run on a proper tree, so for those we just define $\delta(s, l) = \mathsf{false}$.

Correctness of $\mathfrak{A}_1^\lambda$. We now argue that $\mathfrak{A}_1^\lambda$ accepts a tree if and only if it encodes a tuple $(\mathcal{A}, \mathbf{a}, p)$ such that Condition 1 is fulfilled.

Let Condition 1 be fulfilled. We show that the automation accepts. This entails that the homomorphism pattern λ is compatible with the $\mathcal{B}$ and $\mathbf{a}$ that encoded in the label l at the root node of the tree. Hence $\mathfrak{A}_1^\lambda$ does not reject immediately. Let h be the homomorphism from Condition 1. Since h is a homomorphism for all variables z of q_t that are mapped to the core of $\mathcal{A}$ and all atoms $A(z)$ from q_t, we have that $\mathcal{A}, \mathcal{O} \models A(h(z))$, so the conjunction (1) will succeed. For the second conjunction, consider a set of variables $Z \subseteq \mathsf{var}(q_t)$ described by the first conjunction when considering a core constant b. Let t be the set of concept names derived at b in the universal model of $\mathcal{O}$ and $\mathcal{A}$. We argue that that $(t, f) \in R$: By the definition of λ, the homomorphism h maps every variable from Z either to b or to the subtree below b. Since $\mathcal{O}$ is assumed to be in normal form, the subtree generated below b only depends on t and we can define g to be the restriction of h to Z. This function g witnesses that $(t, f) \in R$. For this set t, the last conjunction will of course succeed, since t was chosen to be the set of all concept names derived at b.

For the other direction, let (T, L) be a proper tree that is accepted by $\mathfrak{A}_1^\lambda$ and that encodes the tuple $(\mathcal{A}, \mathbf{a}, p)$. We need to construct the homomorphism h such that Condition 1 is fulfilled. Since the automaton does not reject immediately and the conjunction in (2) is fulfilled, there are $b_1, \ldots, b_n \in \mathsf{C}_{\mathsf{core}}$ such that the sets $Z_i = \lambda^{-1}(\{b_i\} \times \{\mathsf{core}, \mathsf{subtree}\}) \neq \emptyset$ form a partition of $\mathsf{var}(q_t)$ and for every i there is a type t_i derived at b_i in the universal model of $\mathcal{O}$ and $\mathcal{A}$ such that $(t_i, \pi_2 \circ \lambda|_{Z_i}) \in R$. The latter means that there is a homomorphism g_i from q_t restricted to Z_i to the tree that is generated below b_i in the universal model of $\mathcal{O}$ and $\mathcal{A}$. The homomorphism h is obtained by combining the g_i for all i. The second condition in the definition of Σ_ε guarantees that h also preserves roles between variables that lie in different Z_i.

Definition of $\mathfrak{A}_2$. To define $\mathfrak{A}_2$, we precompute three relations R, R' and R''. Let R be a relation between ABoxes $\mathcal{B}$ over $\mathsf{C}_{\mathsf{core}}$, disjuncts $d \in \mathbf{M}^-(\mathcal{B})$ and functions f from $\mathsf{adom}(\mathcal{B})$ to $\mathsf{var}(q)$. A triple $(\mathcal{B}, d, f)$ is in R if and only if there exists a homomorphism $h : d \to q$ such that $h(b) = f(b)$ for all $b \in \mathsf{adom}(\mathcal{B})$. Let R' be a relation between unary mappings from $\mathbf{M}$ and variables from q. A pair $(\varphi(x) \to A(x), y)$ is in R' if and only if there is a homomorphism $h : \varphi(x) \to q$ such that $h(x) = y$. Let R'' be a relation between binary mappings from $\mathbf{M}$ and pairs of variables from q. A triple $(\varphi(x, x') \to r(x, x'), y, y')$ is in R'' if there is a homomorphism $h : \varphi(x, x') \to q$ such that $h(x) = y$ and $h(x') = y'$. All three relations can be computed in ExpTime, since they all only check for homomorphisms between structures of polynomial size.

Let $\mathfrak{A}_2 = (S, \delta, \Sigma_\varepsilon \uplus \Sigma_N, s_0, c)$ where

$$S = \{s_0\} \uplus \{s^b_y \mid b \in \mathsf{C}_{\mathsf{core}} \text{ and } y \in \mathsf{var}(q)\} \\ \uplus \{s_y \mid y \in \mathsf{var}(q)\}$$

and $c(s) = 0$ for every $s \in S$, but the actual value of $c(s)$ does not matter since all runs of $\mathfrak{A}_2$ on proper trees will be finite.

For $l \in \Sigma_\varepsilon$ and $\mathcal{B} \in l$ and $d \in \mathbf{M}^-(\mathcal{B})$ the disjunct defined by the mappings in l, we define:

$$\delta(s_0, l) = \bigvee_{\substack{f:\mathsf{adom}(\mathcal{B})\to\mathsf{var}(q)\\ (\mathcal{B},d,f)\in R}} \bigwedge_{b\in\mathsf{adom}(\mathcal{B})} \bigwedge_{i\in\{1,\dots,|\mathsf{C}_{\mathsf{core}}|\cdot|\mathcal{O}|\}} [i]s^b_{f(b)}$$

For $l \in \Sigma_N$ we define:

$$\delta(s^b_y, l) = \begin{cases} \mathsf{true} & \text{if } b \notin l \\ \langle 0\rangle s_y & \text{if } b \in l \end{cases}$$

For $l = (\Theta, M, \mu) \in \Sigma_N$ and $y \in \mathsf{var}(q)$ let Y^y_l be the set of all $y' \in \mathsf{var}(q)$ such that $(\varphi(x, x') \to r(x, x'), y, y') \in R''$, where $\varphi(x, x') \to r(x, x')$ is the mapping M corresponding to the unique role in Θ, and such that $(\varphi(x) \to A(x), y') \in R'$ for every concept name $A \in \Theta$, where $\varphi(x) \to A(x)$ is the mapping $\mu(A)$. Then we define:

$$\delta(s_y, l) = \bigvee_{y'\in Y^y_l} \bigwedge_{i\in\{1,\dots,|\mathcal{O}|\}} [i]s_{y'}$$

Correctness of $\mathfrak{A}_2$. We now argue that $\mathfrak{A}_2$ accepts a tree (T, L) if and only if it encodes a tuple $(\mathcal{A}, \mathbf{a}, p)$ such that $p \to q$.

Assume there is homomorphism $h : p \to q$. We use h to describe a run of $\mathfrak{A}_2$ on (T, L) that traverses the tree once from the root to the leaves. At the root, the automaton chooses as f the restriction of h to $\mathsf{adom}(\mathcal{B})$, which is possible because $(\mathcal{B}, d, f) \in R$. If the run is at a position (t, s_y), where $t \in T$ corresponding to a constant a in $\mathcal{A}$, then we choose $h(a)$ as y'. Because h is a homomorphism, one can check that $y' \in Y^y_{L(t)}$. Thus, $\mathcal{A}_2$ accepts (T, L).

For the other direction, let (T, L) be a tree encoding a tuple $(\mathcal{A}, \mathbf{a}, p)$ that is accepted by $\mathfrak{A}_2$. Let ρ be an accepting run of $\mathfrak{A}_2$ on (T, L). We obtain a homomorphism $h : p \to q$ by gluing together all of the following homomorphisms:

- The homomorphism from p restricted to facts generated by facts in $\mathcal{B}$ to q that exists by the choice of f in the root node.
- All homomorphisms h' obtained as follows: Consider any non-core fact α from $\mathcal{A}$. This fact appears in the label $L(t)$ of some t in (T, L). Since ρ is accepting, it will visit t in some state of the form s_y and chooses $y' \in Y^y_{L(t)}$. Because α is encoded in $L(t)$, it follows by the definition of $Y^y_{L(t)}$ that there is a homomorphism from the body of the mapping corresponding to α to q, which we choose as h'.

The information that is passed along in the states of the automaton guarantees that all these homomorphisms can be glued together to obtain a single homomorphism $h : p \to q$.

Lower Bounds for $\mathcal{EL}$, Missing Details

We first give some justification of Theorem 21.

Theorem 21. (Bienvenu et al. 2016) Containment between OMQs $Q_1 = (\mathcal{O}, \Sigma, q_1)$ and $Q_2 = (\mathcal{O}, \Sigma, q_2)$ with $\mathcal{O}$ an $\mathcal{ELI}$-ontology, q_1 an AQ, and q_2 a rooted UCQ is coNExpTime-hard even when

1. $q_2(x)$ uses only symbols from Σ and
2. no symbol from Σ occurs on the right-hand side of a CI in $\mathcal{O}$.

The hardness proof in (Bienvenu et al. 2016) actually produces as q_2 a rooted CQ, but does not satisfy Condition 2. However, the only exception to Condition 2 are CIs of the form $D \sqsubseteq C_q$ where C_q is a specially crafted $\mathcal{ELI}$-concept that uses only symbols from Σ and is designed to 'make the query q_2 true', that is, whenever $d \in C^{\mathcal{I}}_q$ in an interpretation $\mathcal{I}$, then $\mathcal{I} \models q(d)$. It can be verified that the reduction in (Bienvenu et al. 2016) still works when replacing the rooted CQ q_2 with the rooted UCQ $q_2 \vee \bigvee_{D\sqsubseteq C_q\in\mathcal{O}} q_D$, where q_D is the concept D viewed as a unary CQ in the obvious way, and then deleting all CIs of the form $D \sqsubseteq C_q$ from $\mathcal{O}$. Arguably, this modification even yields the more natural reduction, avoided in (Bienvenu et al. 2016) to ensure that q_2 is a CQ rather than a UCQ.

Towards a proof of Lemma 22, consider the OMQ $Q = (\mathcal{O}', \mathbf{S}, q_s)$ and the infinitary UCQ q_r that consists of the following CQs:

1. $S(x)$,
2. each CQ from $q(x)$,
3. for every tree-shaped Σ-ABox $\mathcal{A}$ with root $a_0 \in \mathsf{cert}_{Q_1}(\mathcal{A})$, $(\mathcal{A}, a)$ viewed as a CQ.

The following is straightforward to verify. In particular, the restriction to tree-shaped ABoxes in Point 3 is sanctioned by Lemma 15 and it is important that $q(x)$ uses only symbols from Σ which cannot be derived using the ontology.

Lemma 31. q_r is a rewriting of Q.

The next lemma is the core ingredient to the proof of Lemma 22. In fact, by Theorem 6 and since $\mathbf{M}(q_s) = q_s$ and $\mathbf{M}^-(q_r) = q_r$, q_s is UCQ-expressible in $\mathcal{S}$ iff $q_r \subseteq_{\mathbf{S}} q_s$. The following lemma states that this is the case iff $Q_1 \subseteq Q_2$.

Lemma 32. $Q_1 \subseteq Q_2$ iff for every CQ p in q_r, there is a CQ $p' \in q_s$ with $p' \to p$.

Proof. "if". Assume that $Q_1 \not\subseteq Q_2$. Then there is a Σ-ABox $\mathcal{A}$ and an $a \in \mathsf{adom}(\mathcal{A})$ such that $a \in \mathsf{cert}_{Q_1}(\mathcal{A})$, but $a \notin \mathsf{cert}_{Q_2}(\mathcal{A})$. By Lemma 15 and as already observed in (Bienvenu et al. 2016), we can assume that $\mathcal{A}$ is tree-shaped with root a. Let $q_{\mathcal{A}}$ be $(\mathcal{A}, a)$ viewed as a CQ. Clearly $a \in \mathsf{cert}_Q(\mathcal{A})$ and thus $q_{\mathcal{A}}$ is a CQ in q_r. However, from none of the CQs in the UCQ q_s there is a homomorphism to $q_{\mathcal{A}}$. This is true for $S(x)$ since S does not occur in $q_{\mathcal{A}}$. It is also true for $q(x)$ since $\mathcal{A} \not\models Q_2(a)$ and q does not use symbols that occur on the right-hand side of CIs in $\mathcal{O}$.

"only if". Assume that $Q_1 \subseteq Q_2$. We have to show that for every CQ p in q_r, there is a CQ $p' \in q_s$ with $p' \to p$. This is clear for the CQs from Points 1 and 2 of the construction of q_r since all of them appear as a CQ also in $q_s(x)$.

For Point 3, let $\mathcal{A}$ be a tree-shaped Σ-ABox with root $a_0 \in \mathsf{cert}_{Q_1}(\mathcal{A})$. From $Q_1 \subseteq Q_2$, we obtain $a_0 \in \mathsf{cert}_{Q_2}(\mathcal{A})$. Let p be $(\mathcal{A}, a)$ viewed as a CQ. By Points 1 and 2 from Theorem 21 , $a_0 \in \mathsf{cert}_{Q_2}(\mathcal{A})$ implies $a_0 \in \mathsf{ans}_q(\mathcal{A})$ and thus $q \to p$ and we are done. ❑

We now describe how the reduction can be improved to work for $\mathcal{EL}$, that is, how the $\mathcal{ELI}$-ontology $\mathcal{O}'$ can be replaced with an $\mathcal{EL}$-ontology. It can be verified that query Q_1 from Theorem 21 is 'one-way', that is, $\mathcal{O}$ verifies the existence of a (homomorphic image of a) certain tree-shaped sub-ABox *from the bottom up* and then Q_1 makes $q_1 = A_0(x)$ true at the root when the tree-shaped ABox was found. This one-way behaviour can be made formal in terms of derivations of A_0 by $\mathcal{O}$, see (Bienvenu et al. 2016).

Assume w.l.o.g. that $\mathcal{O}$ is in *normal form*, that is, it only contains CIs of the forms $\top \sqsubseteq A$, $A \sqsubseteq \bot$, $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, and $\exists r.A \sqsubseteq B$ where $A, B, A_1, \ldots, A_n$ are concept names. A *derivation* for a fact $A_0(a_0)$ in an ABox $\mathcal{A}$ with $A_0 \in \mathsf{N_C} \cup \{\bot\}$ is a finite $\mathsf{adom}(\mathcal{A}) \times (\mathsf{N_C} \cup \{\bot\})$-labeled tree (T, V) that satisfies the following conditions:

1. $V(\varepsilon) = (a_0, A_0)$;
2. if $V(x) = (a, A)$ and neither $A(a) \notin \mathcal{A}$ nor $\top \sqsubseteq A \in \mathcal{O}$, then one of the following holds:
   - x has successors $y_1, \ldots, y_k$, $k \geq 1$ with $V(y_i) = (a, A_i)$ for $1 \leq i \leq k$ and $\mathcal{O} \models A_1 \sqcap \cdots \sqcap A_k \sqsubseteq A$;
   - x has a single successor y with $V(y) = (b, B)$ and there is an $\exists R.B \sqsubseteq A \in \mathcal{O}$ and a $R'(a, b) \in \mathcal{A}$ such that $\mathcal{O} \models R' \sqsubseteq R$;
   - x has a single successor y with $V(y) = (b, B)$ and there is a $B \sqsubseteq \exists r.A \in \mathcal{O}$ such that $r(b, a) \in \mathcal{A}$ and $\mathsf{func}(r) \in \mathcal{O}$.

Now if $\mathcal{A}$ is tree-shaped, then the derivation is *bottom-up* if the following condition is satisfied: if y is a successor of x in T, $V(x) = (a_x, A_x)$, and $V(y) = (a_y, A_y)$, then $a_x = a_y$ or a_y is a successor (but not a predecessor) or a_x in $\mathcal{A}$, that is, a_y is further away from the root of $\mathcal{A}$ than a_x is. The OMQ Q_1 is one-way in the sense that if $\mathcal{A}$ is a tree-shaped Σ-ABox with root $a_0 \in \mathsf{cert}_{Q_1}(\mathcal{A})$, then all derivations of $A_0(a_0)$ in $\mathcal{A}$ are bottom-up. Note that a corresponding statement for Q_2 makes little sense since by Conditions 1 and 2 of Theorem 21, answers to Q_2 on an ABox $\mathcal{A}$ are independent of $\mathcal{O}$.

We can exploit the one-way property of Q_1 as follows. In the hardness proof in (Bienvenu et al. 2016), all involved ontologies, signatures, and queries use only a single role name S that is interpreted as a *symmetric role*, and in fact represented via the composition $r_0^-; r_0$ where r_0 is a fixed 'standard' (non-symmetric) role name. We can replace S with a standard role name r in $\mathcal{O}$ and in Σ, turning the $\mathcal{ELI}$-ontology $\mathcal{O}$ into an $\mathcal{EL}$-ontology. The mapping $r_0(x, y) \to r_0(x, y)$ from $\mathbf{M}$ in the original reduction is then replaced with $r_0(x, y) \land r_0(x, z) \to r(y, z)$; note that, in q_s, we keep the composition $r_0^-; r_0$ from the original reduction. We claim that, again, the following holds.

Lemma 33. $Q_1 \subseteq Q_2$ iff q_s is UCQ-expressible in $\mathcal{S}$.

Proof. (sketch) Let the UCQ q_r be defined as before except that the CQs from q are replaced with those from $\mathbf{M}(q)$. It can be verified that q_r is a rewriting of $Q = (\mathcal{O}', \mathbf{S}, \mathbf{M}(q_s))$. Moreover, it is easy to see that $\mathbf{M}^-(\mathbf{M}(q)) = q$. This and the fact that Q_1 is one-way can be used to verify that $\mathbf{M}^-(q_r)$ is identical to the query q_r from the original reduction (the one-way property is needed to see that the CQs from Point 3 of the definition of q_r are identical in both cases, except that S is replaced with r). We therefore get from Lemma 32 that

($*$) $Q_1 \subseteq Q_2$ iff for every CQ p in $\mathbf{M}^-(q_r)$, there is a CQ $p' \in q_s$ with $p' \to p$.

By Theorem 6 q_s is UCQ-expressible in $\mathcal{S}$ iff $\mathbf{M}^-(q_r) \subseteq_{\mathbf{S}} q_s$. By ($*$), this is the case iff $Q_1 \subseteq Q_2$. ❑

Now for Point 2 of Theorem 20. We identify a suitable containment problem proved 2ExpTime in (Bienvenu et al. 2016) and then proceed very similarly to the case of Point 1.

Theorem 34. (Bienvenu et al. 2016) Containment between OMQs $Q_1 = (\mathcal{O}, \Sigma, q_1)$ and $Q_2 = (\mathcal{O}, \Sigma, q_2)$ with $\mathcal{O}$ an $\mathcal{ELI}$-ontology, q_1 of the form $\exists x\, A_0(x)$, and q_2 a UCQ is 2ExpTime-hard even when

1. $q_2(x)$ uses only symbols from Σ
2. no symbol from Σ occurs on the right-hand side of a CI in $\mathcal{O}$;
3. all occurrences of $\bot$ in $\mathcal{O}$ are of the form $C \sqsubseteq \bot$ where C is an $\mathcal{ELI}$-concept in signature Σ.

Again, the actual hardness proof from (Bienvenu et al. 2016) needs to be slightly modified to actually achieve what is stated in Theorem 34. In particular, CIs of the form $D \sqsubseteq C_q$ again have to be turned into additional disjuncts of the UCQ q_2. This requires an additional modification of the reduction since there are CIs of the form $D \sqsubseteq C_q$ where D uses symbols that are not from Σ and occur on the right-hand side of CIs in $\mathcal{O}$. In a nutshell and speaking in terms of the notation from (Bienvenu et al. 2016), the concept names H and W' need to be added to Σ and their presence at the intended places must be 'verified in the input' rather than 'enforced by the ontology'. After this is done, the only (minor) problem remaining is that the concept name G is used (exactly twice) in a (single) CQ p in q_2, but it does occur on the right-hand side of two CIs which are $G_1 \sqsubseteq G$ and $G_2 \sqsubseteq G$. Here, G_1, G_2 are from Σ and do not occur on the right-hand side of a CI. This can be fixed by replacing p with four CQs in the UCQ q_2, replacing the two occurrences of G with G_1 or G_2 in all possible ways.

Point 2 of Theorem 20 is again first proved for $[\mathcal{ELI}, \mathrm{GAV}]$ instead of for $[\mathcal{EL}, \mathrm{GAV}]$. This is done by reduction from the containment problem in Theorem 34. Let $Q_1 = (\mathcal{O}, \Sigma, A_0(x))$ and $Q_2 = (\mathcal{O}, \Sigma, q)$. We define an OBDA-specification $\mathcal{S} = (\mathcal{O}', \mathbf{M}, \mathbf{S})$ and a query $q_s()$ over $\mathbf{S}$ as follows. Let B be a concept name that does not occur in Q_1 and Q_2. Set

$$\begin{aligned}
\mathcal{O}' &= \mathcal{O} \cup \{A_0 \sqsubseteq B\} \\
\mathbf{S} &= \Sigma \cup \{B\} \\
q_s() &= \exists x\, B(x) \lor q()
\end{aligned}$$

The set $\mathbf{M}$ of mappings again contains $A(x) \rightarrow A(x)$ for all concept names $A \in \mathbf{S}$ and $r(x,y) \rightarrow r(x,y)$ for all role names $r \in \mathbf{S}$. The proof of the following is essentially identical to the coNExpTime case and omitted.

Lemma 35. $Q_1 \subseteq Q_2$ iff q_s is UCQ-expressible in $\mathcal{S}$.

The approach to eliminating inverse roles is also exactly identical to the coNExpTime case. In fact, the OMQ $(\mathcal{O}, \Sigma, A_0(x))$ is again one-way and all involved ontologies, signatures and queries again only use the single symmetric role name S represented as the composition $r_0^-; r_0$. Consequently, the same arguments apply.

Proof of Undecidability Result

Theorem 23. In $[\mathcal{ALCF}, \text{GAV}]$, the AQ-to-$\mathcal{Q}$ expressibility and verification problems are undecidable for any $\mathcal{Q} \in \{\text{AQ}, \text{CQ}, \text{UCQ}\}$.

Proof. We provide a reduction from the emptiness of AQs w.r.t. $\mathcal{ALCF}$-ontologies, which is undecidable (Baader et al. 2016). Thus let $(\mathcal{O}, \mathbf{S}, A_0)$ be an OMQ with $\mathcal{O}$ in $\mathcal{ALCF}$ and $A_0(x)$ an AQ. Let B_0 be a fresh concept name and define an OBDA-specification $\mathcal{S} = (\mathcal{O}', \mathbf{M}, \mathbf{S}')$ where $\mathcal{O}' = \mathcal{O} \cup \{A_0 \sqsubseteq B_0\}$, $\mathbf{S}' = \mathbf{S} \cup \{B_0\}$, and $\mathbf{M}$ consists of the mappings $A(x) \rightarrow A(x)$ for all concept names A in $\mathbf{S}'$ and $r(x,y) \rightarrow r(x,y)$ for all role names r in $\mathbf{S}'$. We consider expressibility of the AQ $B_0(x)$. In fact, it is possible to verify the following:

1. if A_0 is empty w.r.t. $\mathcal{O}$, then $B_0(x)$ is a realization of $B_0(x)$ in $\mathcal{S}$;
2. if A_0 is non-empty w.r.t. $\mathcal{O}$, then there is a $\mathbf{S}$-database D and a constant $a \in \mathsf{adom}(D)$ such that $a \in \mathsf{cert}_Q(D)$. Define the $\mathbf{S}'$-database $D' = D \cup \{B_0(a)\}$. Now, $B_0(x)$ is not determined in $\mathcal{S}$ in the sense that $a \in \mathsf{ans}_{B_0}(D') \setminus \mathsf{ans}_{B_0}(D)$ but every model of $\mathbf{M}(D)$ and $\mathcal{O}'$ is also a model of $\mathbf{M}(D')$ and $\mathcal{O}'$, and vice versa. Consequently, $B_0(x)$ is not $\mathcal{Q}$-expressible in $\mathcal{S}$ for any $\mathcal{Q} \in \{\text{AQ}, \text{CQ}, \text{UCQ}\}$ or, in fact, any other query language.

❑