Ontology-Based Access to Probabilistic Data with OWL QL

Jean Christoph Jung and Carsten Lutz

Universität Bremen, Germany
{jeanjung,clu}@informatik.uni-bremen.de

Abstract. We propose a framework for querying probabilistic instance data in the presence of an OWL2 QL ontology, arguing that the interplay of probabilities and ontologies is fruitful in many applications such as managing data that was extracted from the web. The prime inference problem is computing answer probabilities, and it can be implemented using standard probabilistic database systems. We establish a PTime vs. #P dichotomy for the data complexity of this problem by lifting a corresponding result from probabilistic databases. We also demonstrate that query rewriting (backwards chaining) is an important tool for our framework, show that non-existence of a rewriting into first-order logic implies #P-hardness, and briefly discuss approximation of answer probabilities.

1 Introduction

There are many applications that require data to be first extracted from the web and then further processed locally, by feeding it into a relational database system (RDBMS). Such web data differs in several crucial aspects from traditional data stored in RDBMSs: on the one hand, it is *uncertain* because of the unreliability of many web data sources and due to the data extraction process, which relies on heuristic decisions and is significantly error prone [23]; on the other hand, web data is often provided without explicit schema information and thus requires *interpretation* based on ontologies and other semantic techniques. This latter aspect is addressed by ontology languages such as OWL2. In particular, the OWL2 QL profile is a popular lightweight ontology language that is tailored towards enriching standard RDBMS query answering with an ontology component, thus allowing the user of semantic technologies to take advantage of the maturity and effiency of such systems [6]. While the current techniques developed around OWL2 QL are well-suited to deal with the interpretation aspect of web data, they are not able to deal with its uncertainty. In this paper, we propose and analyze a framework for data storage and querying that supports ontologies formulated in (a fragment of) OWL2 QL, but also features prominent aspects of probabilistic database models to explicitly represent uncertainty. In a nutshell, our approach relates to probabilistic database systems (PDBMSs) in the same way that traditional OWL2 QL query answering relates to RDBMSs.

In our framework, we adopt data models from the recently very active area of probabilistic databases [7,31], but use an open world assumption as is standard in the context of OWL2 QL. Specifically, we store data in description logic ABoxes enriched with probabilities that are attached to *probabilistic events*, which can either be modeled explicitly (resulting in what we call *pABoxes*) or be implicitly associated with each ABox assertion (resulting in *ipABoxes*). For example, a pABox assertion SoccerPlayer(messi) can be associated with an *event expression* $e_1 \vee e_2$, where e_1 and e_2 represent events such as 'web extraction tool x correctly analyzed webpage y stating that Messi is a soccer player'. We generally assume all events to be probabilistically independent, which results in a straightforward semantics that is similar to well-known probabilistic versions of datalog [27,12]. Ontologies are represented by TBoxes formulated in the description logic DL-Lite, which forms a logical core of the ontology language OWL2 QL. We are then interested in *computing the answer probabilities* to *conjunctive queries (CQs)*; note that probabilities can occur only in the data, but neither in the ontology nor in the query. We believe that this setup is of general interest and potentially useful for a wide range of applications including the management of data extracted from the web, machine translation, and dealing with data that arises from sensor networks. All these applications can potentially benefit from a fruitful interplay between ontologies and probabilities; in particular, we argue that the ontology can help to reduce the uncertainty of the data.

In database research, practical feasibility is usually identified with PTime data complexity, where *data complexity* means to treat only the (probabilistic) data as an input while considering both the TBox and the query to be fixed. The main aim of this paper is to study the data complexity of *ontology-based access to probabilistic data (pOBDA)* in the concrete framework described above. As a central tool, we use *query rewriting* (also called *backwards chaining*), which is an important technique for traditional *ontology based data access (OBDA)*, i.e., answering CQs in the presence of a DL-Lite TBox over non-probabilistic data [6]. Specifically, the idea is to rewrite a given CQ q and DL-Lite TBox $\mathcal{T}$ into an SQL (equivalently: first-order) query $q_\mathcal{T}$ such that for any ABox $\mathcal{A}$, the certain answers to q over $\mathcal{A}$ relative to $\mathcal{T}$ are identical with the answers to $q_\mathcal{T}$ over $\mathcal{A}$ viewed as a relational database instance. We set out with observing that rewritings from traditional OBDA are useful also in the context of pOBDA: for any pABox $\mathcal{A}$, the probability that a tuple $\boldsymbol{a}$ is a certain answer to q over $\mathcal{A}$ relative to $\mathcal{T}$ is identical to the probability that $\boldsymbol{a}$ is an answer to $q_\mathcal{T}$ over $\mathcal{A}$ viewed as a probabilistic database. This *lifting* of query rewriting to the probabilistic case immediately implies that one can implement pOBDA based on existing PDBMSs such as MayBMS, Trio, and MystiQ [1,33,5].

Lifting also allows us to carry over the dichotomy between PTime and #P-hardness for computing the probabilities of answers to unions of conjunctive queries (UCQs) over probabilistic databases recently obtained by Dalvi, Suciu, and Schnaitter [8] to our pOBDA framework provided that we restrict ourselves to ipABoxes, which are strictly less expressive than pABoxes. Based on a careful syntactic analysis, we provide a transparent characterization of those CQs q and DL-Lite TBoxes $\mathcal{T}$ for which computing answer probabilities is in PTime. We then proceed to showing that query rewriting is a *complete* tool for proving

PTime data complexity in pOBDA, in the following sense: we replace DL-Lite with the strictly more expressive description logic $\mathcal{ELI}$, which is closely related to the OWL2 profile OWL2 EL and where, in contrast to DL-Lite, rewritings into first-order queries do not exist for every CQ q and TBox $\mathcal{T}$; we then prove that if any $(q, \mathcal{T})$ does *not* have a rewriting, then computing answer probabilities for q relative to $\mathcal{T}$ is #P-hard. Thus, if it is possible at all to prove that some $(q, \mathcal{T})$ has PTime data complexity, then this can always be done using query rewriting. Both in DL-Lite and $\mathcal{ELI}$, the class of queries and TBoxes with PTime data complexity is relatively small, which leads us to also consider the approximation of answer probabilities, based on the notion of a *fully polynomial randomized approximation scheme (FPRAS)*. It is not hard to see that all pairs $(q, \mathcal{T})$ have an FPRAS when $\mathcal{T}$ is formulated in DL-Lite, but this is not the case for more expressive ontology languages such as $\mathcal{ALC}$. Note that these results are in the spirit of the *non-uniform* analysis of data complexity recently initiated in an OBDA context in [26]. We defer some proofs to the appendix of the long version of this paper, available at http://www.informatik.uni-bremen.de/tdki/research/papers.html.

Related Work. The probabilistic ABox formalism studied in this paper is inspired by the probabilistic database models in [9], but can also be viewed as a variation of probabilistic versions of datalog and Prolog, see [27,12] and references therein. There have recently been other approaches to combining ontologies and probabilities for data access [11,14], with a different semantics; the setup considered by Gottlob, Lukasiewicz, and Simari in [14] is close in spirit to the framework studied here, but also allows probabilities in the TBox and has a different, rather intricate semantics based on Markov logic. In fact, we deliberately avoid probabilities in the ontology because (i) this results in a simple and fundamental, yet useful formalism that still admits a very transparent semantics and (ii) it enables the use of standard PDBMSs for query answering. There has also been a large number of proposals for enriching description logic TBoxes (instead of ABoxes) with probabilities, see [24,25] and the references therein. Our running application example is web data extraction, in the spirit of [16] to store extracted web data in a probabilistic database. Note that It has also been proposed to integrate both probabilities and ontologies directly into the data extraction tool [13]. We believe that both approaches can be useful and could even be orchestrated to play together. Finally, we note that the motivation for our framework is somewhat similar to what is done in [30], where the retrieval of top-k-answers in OBDA is considered under a fuzzy logic-like semantics based on 'scoring functions'.

2 Preliminaries

We use standard notation for the syntax and semantics of description logics (DLs) and refer to [3] for full details. As usual, $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ denote countably infinite sets of concept names, role names, and individual names, C, D denote (potentially) composite concepts, A, B concept names, r, s role names, R and S

role names or their inverse, and a, b individual names. When $R = r^-$, then as usual R^- denotes r. We consider the following DLs.

In *DL-Lite*, *TBoxes* are finite sets of *concept inclusions (CIs)* $B \sqsubseteq B'$ and $B \sqcap B' \sqsubseteq \bot$ with B and B' concepts of the form $\exists r$, $\exists r^-$, $\top$ or A. Note that there is no nesting of concept constructors in DL-Lite. This version is sometimes called *DL-Lite*$_{\mathsf{core}}$ and includes crucial parts of the OWL2 QL profile; some features of OWL2 QL are omitted in DL-Lite$_{\mathsf{core}}$, mainly to keep the presentation simple.

$\mathcal{ELI}$ is a generalization of DL-Lite without $\bot$ (which we will largely disregard in this paper for reasons explained later on) and offers the concept constructors $\top$, $C \sqcap D$, $\exists r.C$, and $\exists r^-.C$. In $\mathcal{ELI}$, a *TBox* is a finite set of CIs $C \sqsubseteq D$ with C and D (potentially) compound concepts.

In DLs, data is stored in an *ABox*, which is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individual names used in the ABox $\mathcal{A}$ and sometimes write $r^-(a, b) \in \mathcal{A}$ for $r(b, a) \in \mathcal{A}$.

The semantics of DLs is based on interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ as usual [3]. An interpretation is a *model* of a TBox $\mathcal{T}$ (resp. ABox $\mathcal{A}$) if it satisfies all concept inclusions in $\mathcal{T}$ (resp. assertions in $\mathcal{A}$), where satisfaction is defined in the standard way. An ABox $\mathcal{A}$ is *consistent* w.r.t. a TBox $\mathcal{T}$ if $\mathcal{A}$ and $\mathcal{T}$ have a common model. We write $\mathcal{T} \models C \sqsubseteq D$ if for all models $\mathcal{I}$ of $\mathcal{T}$, $C^\mathcal{I} \subseteq D^\mathcal{I}$ and say that C *is subsumed by* D relative to $\mathcal{T}$.

Conjunctive queries (CQs) take the form $\exists \boldsymbol{y}.\varphi(\boldsymbol{x}, \boldsymbol{y})$, with φ a conjunction of atoms of the form $A(t)$ and $r(t, t')$ and where $\boldsymbol{x}, \boldsymbol{y}$ denote (tuples of) variables taken from a set $\mathsf{N_V}$ and t, t' denote *terms*, i.e., a variable or an individual name. We call the variables in $\boldsymbol{x}$ the *answer variables* and those in $\boldsymbol{y}$ the *quantified* variables. The set of all variables in a CQ q is denoted by $\mathsf{var}(q)$ and the set of all terms in q by $\mathsf{term}(q)$. A CQ q is *n-ary* if it has n answer variables and *Boolean* if it is 0-ary. Whenever convenient, we treat a CQ as a *set* of atoms and sometimes write $r^-(t, t') \in q$ instead of $r(t', t) \in q$.

Let $\mathcal{I}$ be an interpretation and q a CQ with answer variables $x_1, \dots, x_k$. For $\boldsymbol{a} = a_1 \cdots a_k \in {\mathsf{N_I}}^k$, an *$\boldsymbol{a}$-match* for q in $\mathcal{I}$ is a mapping $\pi : \mathsf{term}(q) \to \Delta^\mathcal{I}$ such that $\pi(x_i) = a_i$ for $1 \leq i \leq k$, $\pi(a) = a^\mathcal{I}$ for all $a \in \mathsf{term}(q) \cap \mathsf{N_I}$, $\pi(t) \in A^\mathcal{I}$ for all $A(t) \in q$, and $(\pi(t_1), \pi(t_2)) \in r^\mathcal{I}$ for all $r(t_1, t_2) \in q$. We write $\mathcal{I} \models q[\boldsymbol{a}]$ if there is an $\boldsymbol{a}$-match of q in $\mathcal{I}$. For a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, we write $\mathcal{T}, \mathcal{A} \models q[\boldsymbol{a}]$ if $\mathcal{I} \models q[\boldsymbol{a}]$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. In this case and when all elements of $\boldsymbol{a}$ are from $\mathsf{Ind}(\mathcal{A})$, $\boldsymbol{a}$ is a *certain answer* to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$. We use $\mathsf{cert}_\mathcal{T}(q, \mathcal{A})$ to denote the set of all certain answers to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$.

As done often in the context of OBDA, we adopt the unique name assumption (UNA), which requires that $a^\mathcal{I} \neq b^\mathcal{I}$ for all interpretations $\mathcal{I}$ and all $a, b \in \mathsf{N_I}$ with $a \neq b$. Note that, in all logics studied here, query answers with and without UNA actually coincide, so the assumption of the UNA is without loss of generality.

3 Probabilistic OBDA

We introduce our framework for probabilistic OBDA, starting with the definition of a rather general, probabilistic version of ABoxes. Let $\mathcal{E}$ be a countably infinite set of *atomic (probabilistic) events*. An *event expression* is built up from atomic events using the Boolean operators $\neg$, $\wedge$, $\vee$. We use $\mathsf{expr}(\mathcal{E})$ to denote the set of all event expressions over $\mathcal{E}$. A *probability assignment for* E is a map $E \to [0, 1]$.

Definition 1 (pABox). *A* probabilistic ABox (pABox) *is of the form* $(\mathcal{A}, e, p)$ *with* $\mathcal{A}$ *an ABox,* e *a map* $\mathcal{A} \to \mathsf{expr}(\mathcal{E})$*, and* p *a probability assignment for* $E_{\mathcal{A}}$*, the atomic events in* $\mathcal{A}$*.*

Example 1. We consider as a running example a (fictitious) information extraction tool that is gathering data from the web, see [16] for a similar setup. Assume we are gathering data about soccer players and the clubs they play for in the current 2012 season, and we want to represent the result as a pABox.

(1) The tool processes a newspaper article stating that ‘Messi is the soul of the Argentinian national soccer team’. Because the exact meaning of this phrase is unclear (it could refer to a soccer player, a coach, a mascot), it generates the assertion $\mathsf{Player}(\mathsf{messi})$ associated with the atomic event expression e_1 with $p(e_1) = 0.7$. The event e_1 represents that the phrase was interpreted correctly.

(2) The tool finds the Wikipedia page on Lionel Messi, which states that he is a soccer player. Since Wikipedia is typically reliable and up to date, but not *always* correct, it updates the expression associated with $\mathsf{Player}(\mathsf{messi})$ to $e_1 \vee e_2$ and associates e_2 with $p(e_2) = 0.95$.

(3) The tool finds an HTML table on the homepage of FC Barcelona saying that the top scorers of the season are Messi, Villa, and Pedro. It is not stated whether the table refers to the 2011 or the 2012 season, and consequently we generate the ABox assertions $\mathsf{playsfor}(x, \mathsf{FCbarca})$ for $x \in \{\mathsf{messi}, \mathsf{villa}, \mathsf{pedro}\}$ all associated with the same atomic event expression e_3 with $p(e_3) = 0.5$. Intuitively, the event e_3 is that the table refers to 2012.

(4) Still processing the table, the tool applies the background knowledge that top scorers are typically strikers. It generates three assertions $\mathsf{Striker}(x)$ with $x \in \{\mathsf{messi}, \mathsf{villa}, \mathsf{pedro}\}$, associated with atomic events e_4, e_4', and e_4''. It sets $p(e_4) = p(e_4') = p(e_4'') = 0.8$. The probability is higher than in (3) since being a striker is a more stable property than playing for a certain club, thus this information does not depend so much on whether the table is from 2011 or 2012.

(5) The tool processes the twitter message ‘Villa was the only one to score a goal in the match between Barca and Real’. It infers that Villa plays either for Barcelona or for Madrid, generating the assertions $\mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca})$ and $\mathsf{playsfor}(\mathsf{villa}, \mathsf{realmadrid})$. The first assertion is associated with the event e_5, the second one with $\neg e_5$. It sets $p(e_5) = 0.5$.

Now for the semantics of pABoxes $(\mathcal{A}, e, p)$. Each $E \subseteq E_{\mathcal{A}}$ can be viewed as a truth assignment that makes all events in E true and all events in $E_{\mathcal{A}} \setminus E$ false.

Definition 2. *Let* $(\mathcal{A}, e, p)$ *be a pABox. For each* $E \subseteq E_{\mathcal{A}}$*, define a corresponding non-probabilistic ABox* $\mathcal{A}_E := \{\alpha \in \mathcal{A} \mid E \models e(\alpha)\}$*. The function* p *represents a probability distribution on* $2^{E_{\mathcal{A}}}$*, by setting for each* $E \subseteq E_{\mathcal{A}}$*:*

$$p(E) = \prod_{e \in E} p(e) \cdot \prod_{e \in E_{\mathcal{A}} \setminus E} (1 - p(e)).$$

The probability of an answer $\boldsymbol{a} \in \mathsf{Ind}(\mathcal{A})^n$ *to an n-ary conjunctive query q over a pABox $\mathcal{A}$ and TBox $\mathcal{T}$ is*

$$p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) = \sum_{E \subseteq E_{\mathcal{A}} \,:\, \boldsymbol{a} \in \mathsf{cert}_{\mathcal{T}}(q,\mathcal{A}_E)} p(E).$$

For Boolean CQs q, we write $p(\mathcal{A},\mathcal{T} \models q)$ instead of $p_{\mathcal{A},\mathcal{T}}(() \in q)$, where $()$ denotes the empty tuple.

Example 2. Consider again the web data extraction example discussed above. To illustrate how ontologies can help to reduce uncertainty, we use the DL-Lite TBox

$$\begin{aligned} \mathcal{T} = \{ \quad & \exists \mathsf{playsfor} \sqsubseteq \mathsf{Player} & \mathsf{Player} \sqsubseteq \exists \mathsf{playsfor} \\ & \exists \mathsf{playsfor}^- \sqsubseteq \mathsf{SoccerClub} & \mathsf{Striker} \sqsubseteq \mathsf{Player} \quad \} \end{aligned}$$

Consider the following subcases considered above.

(1) + (3) The resulting pABox comprises the following assertions with associated event expressions:

$$\begin{array}{rr} \mathsf{Player}(\mathsf{messi}) \rightsquigarrow e_1 & \mathsf{playsfor}(\mathsf{messi}, \mathsf{FCbarca}) \rightsquigarrow e_3 \\ \mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca}) \rightsquigarrow e_3 & \mathsf{playsfor}(\mathsf{pedro}, \mathsf{FCbarca}) \rightsquigarrow e_3 \end{array}$$

with $p(e_1) = 0.7$ and $p(e_3) = 0.5$. Because of the statement $\exists \mathsf{playsfor} \sqsubseteq \mathsf{Player}$, using $\mathcal{T}$ (instead of an empty TBox) increases the probability of messi to be an answer to the query $\mathsf{Player}(x)$ from 0.7 to 0.85.

(5) The resulting pABox is

$$\mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca}) \rightsquigarrow e_5 \quad \mathsf{playsfor}(\mathsf{villa}, \mathsf{realmadrid}) \rightsquigarrow \neg e_5$$

with $p(e_5) = 0.5$. Although Player(villa) does not occur in the data, the probability of villa to be an answer to the query $\mathsf{Player}(x)$ is 1 (again by the TBox-statement $\exists \mathsf{playsfor} \sqsubseteq \mathsf{Player}$).

(3)+(4) This results in the pABox

$$\begin{array}{rr} \mathsf{playsfor}(\mathsf{messi}, \mathsf{FCbarca}) \rightsquigarrow e_3 & \mathsf{Striker}(\mathsf{messi}) \rightsquigarrow e_4 \\ \mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca}) \rightsquigarrow e_3 & \mathsf{Striker}(\mathsf{villa}) \rightsquigarrow e_4' \\ \mathsf{playsfor}(\mathsf{pedro}, \mathsf{FCbarca}) \rightsquigarrow e_3 & \mathsf{Striker}(\mathsf{pedro}) \rightsquigarrow e_4'' \end{array}$$

with $p(e_3) = 0.5$ and $p(e_4) = p(e_4') = p(e_4'') = 0.8$. Due to the last three CIs in $\mathcal{T}$, each of messi, villa, pedro is an answer to the CQ $\exists y.\mathsf{playsfor}(x, y) \wedge \mathsf{SoccerClub}(y)$ with probability 0.9.

Related Models in Probabilistic Databases. Our pABoxes can be viewed as an open world version of the probabilistic data model studied by Dalvi and Suciu in [9]. It is as a less succinct version of *c-tables*, a traditional data model for probabilistic databases due to Imielinski and Lipski [18]. Nowadays, there is an abundance of probabilistic data models, see [15,29,2] and the references therein.

All these models provide a compact representation of distributions over potentially large sets of *possible worlds*. Since we are working with an open world semantics, pABoxes instead represent a distribution over *possible world descriptions*. Each such description may have any number of models. Note that our semantics is similar to the semantics of ("type 2") probabilistic first-order and description logics [17,25].

Dealing with Inconsistencies. Of course, some of the ABoxes $\mathcal{A}_E$ might be inconsistent w.r.t. the TBox $\mathcal{T}$ used. In this case, it may be undesirable to let them contribute to the probabilities of answers. For example, if we use the pABox

$$\mathsf{Striker}(\mathsf{messi}) \rightsquigarrow e_1 \quad \mathsf{Goalie}(\mathsf{messi}) \rightsquigarrow e_2$$

with $p(e_1) = 0.8$ and $p(e_2) = 0.3$ and the TBox $\mathsf{Goalie} \sqcap \mathsf{Striker} \sqsubseteq \bot$, then messi is an answer to the query $\mathsf{SoccerClub}(x)$ with probability 0.24 while one would probably expect it to be zero (which is the result when the empty TBox is used). We follow Antova, Koch, and Olteanu and advocate a pragmatic solution based on *rescaling* [2]. More specifically, we remove those ABoxes $\mathcal{A}_E$ that are inconsistent w.r.t. $\mathcal{T}$ and rescale the remaining set of ABoxes so that they sum up to probability one. In other words, we set

$$\widehat{p}_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) = \frac{p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) - p(\mathcal{A},\mathcal{T} \models \bot)}{1 - p(\mathcal{A},\mathcal{T} \models \bot)}$$

where $\bot$ is a Boolean query that is entailed exactly by those ABoxes $\mathcal{A}$ that are inconsistent w.r.t. $\mathcal{T}$. The rescaled probability $\widehat{p}_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q)$ can be computed in PTime when this is the case both for $p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q)$ and $p(\mathcal{A},\mathcal{T} \models \bot)$. Note that rescaling results in some effects that might be unexpected such as reducing the probability of messi to be an answer to $\mathsf{Striker}(x)$ from 0.8 to ≈0.74 when the above TBox is added.

In the remainder of the paper, for simplicity we will only admit TBoxes $\mathcal{T}$ such that all ABoxes $\mathcal{A}$ are consistent w.r.t. $\mathcal{T}$.

4 Query Rewriting

The main computational problem in traditional OBDA is, given an ABox $\mathcal{A}$, query q, and TBox $\mathcal{T}$, to produce the certain answers to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$. In the context of lightweight DLs such as DL-Lite, a prominent approach to address this problem is to use *FO-rewriting*, which yields a reduction to query answering in relational databases. The aim of this section is to show that this approach is fruitful also in the case of computing answer probabilities in probabilistic OBDA. In particular, we use it to lift the PTime vs. #P dichotomy result on probabilistic databases recently obtained by Dalvi, Suciu, and Schnaitter [8] to probabilistic OBDA in DL-Lite.

4.1 Lifting FO-Rewritings to probabilistic OBDA

We first describe the query rewriting approach to traditional OBDA. A *first-order query (FOQ)* is a first-order formula $q(\boldsymbol{x})$ constructed from atoms $A(x)$ and $r(x,y)$ using negation, conjunction, disjunction, and existential quantification. The free variables $\boldsymbol{x}$ are the *answer variables* of $q(\boldsymbol{x})$. For an interpretation $\mathcal{I}$, we write $\mathsf{ans}(q,\mathcal{I})$ to denote the *answers to* q *in* $\mathcal{I}$, i.e., the set of all tuples $\boldsymbol{a}$ such that $\mathcal{I} \models q[\boldsymbol{a}]$. In what follows, we use $\mathcal{I}_\mathcal{A}$ to denote the ABox $\mathcal{A}$ viewed as an interpretation (in the obvious way). A *first-order (FO) TBox* is a finite set of first-order sentences.

Definition 3 (FO-rewritable). *A CQ q is* FO-rewritable relative to an FO TBox $\mathcal{T}$ *if one can effectively construct a FOQ $q_\mathcal{T}$ such that $\mathsf{cert}_\mathcal{T}(q,\mathcal{A}) = \mathsf{ans}(q_\mathcal{T},\mathcal{I}_\mathcal{A})$ for every ABox $\mathcal{A}$. In this case, $q_\mathcal{T}$ is a* rewriting *of q relative to $\mathcal{T}$.*

For computing the answers to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$ in traditional OBDA, one can thus construct $q_\mathcal{T}$ and then hand it over for execution to a database system that stores $\mathcal{A}$.

The following observation states that FO-rewritings from traditional OBDA are also useful in probabilistic OBDA. We use $p^d_\mathcal{A}(\boldsymbol{a} \in q)$ to denote the probability that $\boldsymbol{a}$ is an answer to the query q given the pABox $\mathcal{A}$ viewed as a probabilistic database in the sense of Dalvi and Suciu [8]. More specifically,

$$p^d_\mathcal{A}(\boldsymbol{a} \in q) = \sum_{E \subseteq E_\mathcal{A} \mid \boldsymbol{a} \in \mathsf{ans}(q,\mathcal{I}_{\mathcal{A}_E})} p(E)$$

The following is immediate from the definitions.

Theorem 1 (Lifting). *Let $\mathcal{T}$ be an FO TBox, $\mathcal{A}$ a pABox, q an n-ary CQ, $\boldsymbol{a} \in \mathsf{Ind}(\mathcal{A})^n$ a candidate answer for q, and $q_\mathcal{T}$ an FO-rewriting of q relative to $\mathcal{T}$. Then $p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) = p^d_\mathcal{A}(\boldsymbol{a} \in q_\mathcal{T})$.*

From an application perspective, Theorem 1 enables the use of probabilistic database systems such as MayBMS, Trio, and MystiQ for implementing probabilistic OBDA [1,33,5]. Note that it might be necessary to adapt pABoxes in an appropriate way in order to match the data models of these systems. However, such modifications do not impair applicability of Theorem 1.

From a theoretical viewpoint, Theorem 1 establishes query rewriting as a useful tool for analyzing data complexity in probabilistic OBDA. We say that a CQ q *is in* PTime *relative to a TBox* $\mathcal{T}$ if there is a polytime algorithm that, given an ABox $\mathcal{A}$ and a candidate answer $\boldsymbol{a} \in \mathsf{Ind}(\mathcal{A})^n$ to q, computes $p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q)$. We say that q is #P*-hard relative to* $\mathcal{T}$ if the afore mentioned problem is hard for the counting complexity class #P, please see [32] for more information. We pursue a non-uniform approach to the complexity of query answering in probabilistic OBDA, as recently initiated in [26]: ideally, we would like to understand the precise complexity of every CQ q relative to every TBox $\mathcal{T}$, against the background of some preferably expressive 'master logic' used for $\mathcal{T}$. Note, though, that our framework yields one counting problem for each CQ and

TBox, while [26] has one decision problem for each TBox, quantifying over all CQs.

Unsurprisingly, pABoxes are too strong a formalism to admit *any* tractable queries worth mentioning. An n-ary CQ q is *trivial* for a TBox $\mathcal{T}$ iff for every ABox $\mathcal{A}$, we have $\mathsf{cert}_\mathcal{T}(\mathcal{A}, q) = \mathsf{Ind}(\mathcal{A})^n$.

Theorem 2. *Over pABoxes, every CQ q is #P-hard relative to every first-order TBox $\mathcal{T}$ for which it is nontrivial.*

Proof. The proof is by reduction of counting the number of satisfying assignments of a propositional formula.[1] Assume that q has answer variables $x_1, \ldots, x_n$ and let φ be a propositional formula over variables $z_1, \ldots, z_m$. Convert φ into a pABox $\mathcal{A}$ as follows: take q viewed as an ABox, replacing every variable x with an individual name a_x; then associate every ABox assertion with φ viewed as an event expression over events $z_1, \ldots, z_m$ and set $p(z_i) = 0.5$ for all i. We are interested in the answer $\boldsymbol{a} = a_{x_1} \cdots a_{x_n}$. For all $E \subseteq E_\mathcal{A}$ with $E \not\models \varphi$, we have $\mathcal{A}_E = \emptyset$ and thus $\boldsymbol{a} \notin \mathsf{cert}_\mathcal{T}(q, \mathcal{A}_E)$ since q is non-trivial for $\mathcal{T}$. For all $E \subseteq E_\mathcal{A}$ with $E \models \varphi$, the ABox $\mathcal{A}_E$ is the ABox-representation of q and thus $\boldsymbol{a} \in \mathsf{cert}_\mathcal{T}(q, \mathcal{A}_E)$. Consequently, the number of assignments that satisfy φ is $p_{\mathcal{A},\mathcal{T}}(\overline{a} \in q) * 2^m$. Thus, there is a PTime algorithm for counting the number of satisfying assignments given an oracle for computing answer probabilities for q and $\mathcal{T}$. ❏

Theorem 2 motivates the study of more lightweight probabilistic ABox formalisms. While pABoxes (roughly) correspond to c-tables, which are among the most expressive probabilistic data models, we now move to the other end of the spectrum and introduce ipABoxes as a counterpart of *tuple independent databases* [9,12]. Argueably, the latter are the most inexpressive probabilistic data model that is still useful.

Definition 4 (ipABox). *An* assertion-independent probabilistic ABox (short: ipABox) *is a probabilistic ABox in which all event expressions are atomic and where each atomic event expression is associated with at most one ABox assertion.*

To save notation, we write ipABoxes in the form $(\mathcal{A}, p)$ where $\mathcal{A}$ is an ABox and p is a map $\mathcal{A} \to [0, 1]$ that assigns a probability to each ABox assertion. In this representation, the events are only implicit (one atomic event per ABox assertion). For $\mathcal{A}' \subseteq \mathcal{A}$, we write $p(\mathcal{A}')$ as a shorthand for $p(\{e \in E \mid \exists \alpha \in \mathcal{A}' : e(\alpha) = e\})$. Note that $p(\mathcal{A}') = \prod_{\alpha \in \mathcal{A}'} p(\alpha) \cdot \prod_{\alpha \in \mathcal{A} \setminus \mathcal{A}'} (1 - p(\alpha))$ and thus all assertions in an ipABox can be viewed as independent events; also note that for any CQ q, we have $p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) = \sum_{\mathcal{A}' \subseteq \mathcal{A} : \boldsymbol{a} \in \mathsf{cert}_\mathcal{T}(q, \mathcal{A}')} p(\mathcal{A}')$. Cases (1) and (4) of our web data extraction example yield ipABoxes, whereas cases (2), (3), and (5) do not. We refer to [31] for a discussion of the usefulness of ipABoxes/tuple independent databases. For the remainder of the paper, we assume that only ipABoxes are admitted unless explicitly noted otherwise.

[1] Throughout the paper, we use the standard oracle-based notion of reduction originally introduced by Valiant in the context of counting complexity [32].

4.2 Lifting the PTime vs. #P Dichotomy

We now use Theorem 1 to lift a PTime vs. #P dichotomy recently obtained in the area of probabilistic databases to probabilistic OBDA in DL-Lite. Note that, for any CQ and DL-Lite TBox, an FO-rewriting is guaranteed to exist [6]. The central observation is that, by Theorem 1, computing the probability of answers to a CQ q relative to a TBox $\mathcal{T}$ over ipABoxes is *exactly the same problem* as computing the probability of answers to $q_{\mathcal{T}}$ over (ipABoxes viewed as) tuple independent databases. We can thus analyze the complexity of CQs/TBoxes over ipABoxes by analyzing the complexity of their rewritings. In particular, standard rewriting techniques produce for each CQ and DL-Lite TBox an FO-rewriting that is a union of conjunctive queries (a UCQ) and thus, together with Theorem 1, Dalvi, Suciu and Schnaitter's PTime vs. #P dichotomy for UCQs over tuple independent databases [8] immediately yields the following.

Theorem 3 (Abstract Dichotomy). *Let q be a CQ and $\mathcal{T}$ a DL-Lite TBox. Then q is in* PTime *relative to $\mathcal{T}$ or q is #P-hard relative to $\mathcal{T}$.*

Note that Theorem 3 actually holds for *every* DL that enjoys FO-rewritability, including full OWL2 QL. Although interesting from a theoretical perspective, Theorem 3 is not fully satisfactory as it does not tell us *which* CQs are in PTime relative to which TBoxes. In the remainder of this chapter, we carry out a careful inspection of the FO-rewritings obtained in our framework and of the dichotomy result obtained by Dalvi, Suciu and Schnaitter, which results in a more concrete formulation of the dichotomy stated in Theorem 3 and provides a transparent characterization of the PTime cases. For simplicity and without further notice, we concentrate on CQs that are connected, Boolean, and do not contain individual names.

For two CQs q, q' and a TBox $\mathcal{T}$, we say that q *$\mathcal{T}$-implies* q' and write $q \sqsubseteq_{\mathcal{T}} q'$ when $\mathsf{cert}_{\mathcal{T}}(q,\mathcal{A}) \subseteq \mathsf{cert}_{\mathcal{T}}(q',\mathcal{A})$ for all ABoxes $\mathcal{A}$. We say that q and q' are *$\mathcal{T}$-equivalent* and write $q \equiv_{\mathcal{T}} q'$ if $q \sqsubseteq_{\mathcal{T}} q'$ and $q' \sqsubseteq_{\mathcal{T}} q$. We say that q is *$\mathcal{T}$-minimal* if there is no $q' \subsetneq q$ such that $q \equiv_{\mathcal{T}} q'$. When $\mathcal{T}$ is empty, we simply drop it from the introduced notation, writing for example $q \sqsubseteq q'$ and speaking of *minimality.* To have more control over the effect of the TBox, we will generally work with CQs q and TBoxes $\mathcal{T}$ such that q is *$\mathcal{T}$-minimal.* This is without loss of generality because for every CQ q and TBox $\mathcal{T}$, we can find a CQ q' that is $\mathcal{T}$-minimal and such that $q \equiv_{\mathcal{T}} q'$ [4]; note that the answer probabilities relative to $\mathcal{T}$ are identical for q and q'.

We now introduce a class of queries that will play a crucial role in our analysis.

Definition 5 (Simple Tree Queries). *A CQ q is a* simple tree *if there is a variable $x_r \in \mathsf{var}(q)$ that occurs in every atom in q, i.e., all atoms in q are of the form $A(x_r)$, $r(x_r, y)$, or $r(y, x_r)$ ($y = x_r$ is possible). Such a variable x_r is called a* root variable.

As examples, consider the CQs in Figure 1, which are all simple tree queries. The following result shows why simple tree queries are important. A UCQ $\widehat{q}$ is *reduced* if for all disjuncts q, q' of $\widehat{q}$, $q \sqsubseteq q'$ implies $q = q'$.

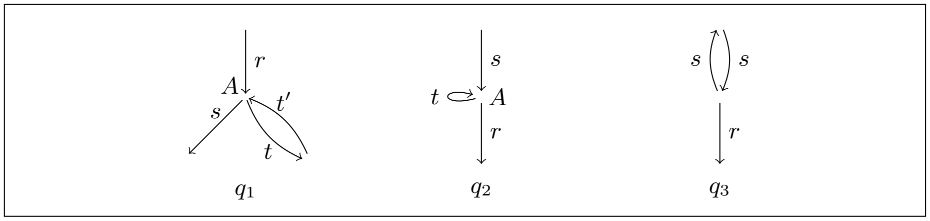

Fig. 1. Example queries

Theorem 4. *Let q be a CQ and $\mathcal{T}$ a DL-Lite TBox such that q is $\mathcal{T}$-minimal and not a simple tree query. Then q is #P-hard relative to $\mathcal{T}$*

Proof. (sketch) Let $q_\mathcal{T}$ be a UCQ that is an FO-rewriting of q relative to $\mathcal{T}$. By definition of FO-rewritings, we can w.l.o.g. assume that q occurs as a disjunct of $q_\mathcal{T}$. The following is shown in [8]:

1. if a minimal CQ does not contain a variable that occurs in all atoms, then it is #P-hard over tuple independent databases;
2. if a reduced UCQ $\widehat{q}$ contains a CQ that is #P-hard over tuple independent databases, then $\widehat{q}$ is also hard over tuple independent databases.

Note that since q is $\mathcal{T}$-minimal, it is also minimal. By Points 1 and 2 above, it thus suffices to show that $q_\mathcal{T}$ can be converted into an equivalent *reduced* UCQ such that q is still a disjunct, which amounts to proving that there is no disjunct q' in $q_\mathcal{T}$ such that $q \sqsubseteq q'$ and $q' \not\sqsubseteq q$. The details of the proof, which is surprisingly subtle, are given in the appendix. ❏

To obtain a dichotomy, it thus remains to analyze simple tree queries. We say that a role R *can be generated in a CQ q* if one of the following holds: (i) there is an atom $R(x_r, y) \in q$ and $y \neq x_r$; (ii) there is an atom $A(x_r) \in q$ and $\mathcal{T} \models \exists R \sqsubseteq A$; (iii) there is an atom $S(x, y) \in q$ with x a root variable and such that $y \neq x$ occurs only in this atom, and $\mathcal{T} \models \exists R \sqsubseteq \exists S$. The concrete version of our dichotomy result is as follows. Its proof is based on a careful analysis of FO-rewritings and the results in (the submitted journal version of) [8].

Theorem 5 (Concrete Dichotomy). *Let $\mathcal{T}$ be a DL-Lite TBox. A $\mathcal{T}$-minimal CQ q is in* PTime *relative to $\mathcal{T}$ iff*

1. *q is a simple tree query, and*
2. *if r and r^- are $\mathcal{T}$-generated in q, then $\{r(x, y)\} \sqsubseteq_\mathcal{T} q$ or q is of the form $\{S_1(x, y), \ldots, S_k(x, y)\}$.*

Otherwise, q is #P-hard relative to $\mathcal{T}$.

As examples, consider again the queries q_1, q_2, and q_3 in Figure 1 and let $\mathcal{T}_\emptyset$ be the empty TBox. All CQs are $\mathcal{T}_\emptyset$-minimal, q_1 and q_2 are in PTime, and q_3 is #P-hard (all relative to $\mathcal{T}_\emptyset$). Now consider the TBox $\mathcal{T} = \{\exists s \sqsubseteq \exists r\}$. Then q_1 is $\mathcal{T}$-minimal and still in PTime; q_2 is $\mathcal{T}$-minimal, and is now #P-hard because both s and s^- is $\mathcal{T}$-generated. The CQ q_3 can be made $\mathcal{T}$-minimal by dropping the r-atom, and is in PTime relative to $\mathcal{T}$.

Theorems 4 and 5 show that only very simple CQs can be answered in PTime. This issue is taken up again in Section 6. We refrain from analyzing in more detail the case where also answer variables and individual names can occur in CQs, and where CQs need not to be connected. It can however be shown that, whenever a connected Boolean CQ q is in PTime relative to a DL-Lite TBox $\mathcal{T}$, then any CQ obtained from q by replacing quantified variables with answer variables and individual names is still in PTime relative to $\mathcal{T}$.

5 Beyond Query Rewriting

We have established FO-rewritability as a tool for proving PTime results for CQ answering in the context of probabilistic OBDA. The aim of this section is to establish that, in a sense, the tool is *complete*: we prove that whenever a CQ q is not FO-rewritable relative to a TBox $\mathcal{T}$, then q is #P-hard relative to $\mathcal{T}$; thus, when a query is in PTime relative to a TBox $\mathcal{T}$, then this can *always* be shown via FO-rewritability. To achieve this goal, we select a DL as the TBox language that, unlike DL-Lite, also embraces non FO-rewritable CQs/TBoxes. Here we choose $\mathcal{ELI}$, which is closely related to the OWL2 EL profile and properly generalizes DL-Lite (as in the previous sections, we do not explicitly consider the $\bot$ constructor). Note that, in traditional OBDA, there is a drastic difference in data complexity of CQ-answering between DL-Lite and $\mathcal{ELI}$: the former is in AC_0 while the latter is PTime-complete.

We focus on Boolean CQs q that are *rooted*, i.e., q involves at least one individual name and is connected. This is a natural case since, for any non-Boolean connected CQ $q(\boldsymbol{x})$ and potential answer $\boldsymbol{a}$, the probability $p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q(\boldsymbol{x}))$ that $\boldsymbol{a}$ is a certain answer to q w.r.t. $\mathcal{A}$ and $\mathcal{T}$ is identical to the probability $p(\mathcal{A},\mathcal{T} \models q[\boldsymbol{a}])$ that $\mathcal{A}$ and $\mathcal{T}$ entail the rooted Boolean CQ $q[\boldsymbol{a}]$. Our main theorem is as follows.

Theorem 6. *If a Boolean rooted CQ q is not FO-rewritable relative to an $\mathcal{ELI}$-TBox $\mathcal{T}$, then q is #P-hard relative to $\mathcal{T}$.*

Since the proof of Theorem 6 involves some parts that are rather technical, we defer full details to the appendix and present only a sketch of the ideas. A central step is the following observation, whose somewhat laborious proof consists of a sequence of ABox transformations. It uses a notion of boundedness similar to the one introduced in [26], but adapted from instance queries to CQs.

Lemma 1. *If a Boolean rooted CQ q is not FO-rewritable relative to an $\mathcal{ELI}$-TBox $\mathcal{T}$, then there exists an ABox $\mathcal{A}$ and assertions $R_3(a_3,a_2)$, $R_2(a_2,a_1)$, $R_1(a_1,a_0)$ such that $\mathcal{A},\mathcal{T} \models q$, but $\mathcal{A}',\mathcal{T} \not\models q$ when $\mathcal{A}'$ is $\mathcal{A}$ with any of the assertions $R_3(a_3,a_2), R_2(a_2,a_1), R_1(a_1,a_0)$ dropped.*

We now prove Theorem 6 by a reduction of the problem of counting the number of satisfying assignments for a monotone bipartite DNF formula, which is known to

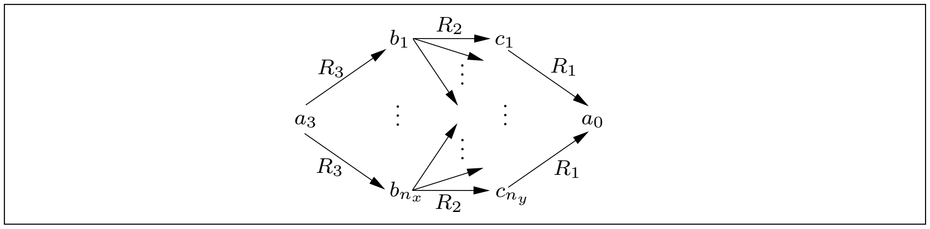

Fig. 2. Gadget for the #P-hardness proof.

be #P-hard. The reduction is similar to what was done in [9]. More specifically, input formulas are of the form $\psi = (x_{i_1} \wedge y_{j_1}) \vee \cdots \vee (x_{i_k} \wedge y_{j_k})$ where the set X of variables that occur on the left-hand side of a conjunction in ψ is disjoint from the set Y of variables that occur on the right-hand side of a conjunction in ψ.

For the reduction, let ψ be a formula as above, $X = \{x_1, \ldots, x_{n_x}\}$, and $Y = \{y_1, \ldots, y_{n_y}\}$. We define an ipABox $(\mathcal{A}_\psi, p_\psi)$ by starting with the ABox $\mathcal{A}$ from Lemma 1 and duplicating the assertions $R_3(a_3, a_2)$, $R_2(a_2, a_1)$, $R_1(a_1, a_0)$ using fresh individual names $b_1, \ldots, b_{n_x}$ and $c_1, \ldots, c_{n_y}$. This is indicated in Figure 2 where, in the middle part, there is an R_2-edge from every b_i to every c_j. Apart from what is shown in the figure, each b_i receives exactly the same role assertions and outgoing edges that a_2 has in $\mathcal{A}$, and each c_i is, in the same sense, a duplicate of a_1 in $\mathcal{A}$.

In the resulting ipABox $\mathcal{A}_\psi$, every assertion except those of the form $R_3(a_3, b_i)$ and $R_1(c_i, a_0)$ has probability 1; specifically, these are all assertions in $\mathcal{A}_\psi$ that are not displayed in the snapshot shown in Figure 2 and all R_2-edges in that figure. The edges of the form $R_3(a_3, b_i)$ and $R_1(c_i, a_0)$ have probability 0.5. For computing the answer probability $p(\mathcal{A}_\psi, \mathcal{T} \models q)$, one has to consider the ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\psi$ with $p(\mathcal{A}') > 0$. Each such ABox has probability $\frac{1}{2^{|X|+|Y|}}$ and corresponds to a truth assignment $\delta_{\mathcal{A}'}$ to the variables in $X \cup Y$: for $x_i \in X$, $\delta_{\mathcal{A}'}(x_i) = 1$ iff $R_3(a_3, b_i) \in \mathcal{A}'$ and for $y_i \in Y$, $\delta_{\mathcal{A}'}(y_i) = 1$ iff $R_1(c_i, a_0) \in \mathcal{A}'$. Let $\#\psi$ the number of truth assignments to the variables $X \cup Y$ that satisfy ψ. To complete the reduction, we show that $p(\mathcal{A}_\psi, \mathcal{T} \models q) = \frac{\#\psi}{2^{|X|+|Y|}}$. By what was said above, this is an immediate consequence of the following lemma, proved in the appendix.

Lemma 2. *For all ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\psi$ with $p_\psi(\mathcal{A}') > 0$, $\delta_{\mathcal{A}'} \models \psi$ iff $\mathcal{A}', \mathcal{T} \models q$.*

This finishes the proof of Theorem 6. As a by-product, we obtain the following; the proof can be found in the long version.

Theorem 7 ($\mathcal{ELI}$ dichotomy). *Let q be a connected Boolean CQ and $\mathcal{T}$ an $\mathcal{ELI}$-TBox. Then q is in* PTime *relative to $\mathcal{T}$ or* #P*-hard relative to $\mathcal{T}$.*

6 Monte Carlo Approximation

The results in Sections 4 and 5 show that PTime complexity is an elusive property even for ipABoxes and relatively inexpressive TBox languages such as DL-

Lite and $\mathcal{ELI}$. Of course, the same is true for probabilistic databases, even for very simple data models such as tuple independent databases. To address this fundamental problem, researchers are often trading accuracy for efficiency, replacing exact answers with approximate ones. In particular, it is popular to use Monte Carlo approximation in the incarnation of a *fully polynomial randomized approximation scheme (FPRAS)*. In this section, we discuss FPRASes in the context of probabilistic OBDA.

An *FPRAS for a Boolean CQ q and TBox $\mathcal{T}$* is a randomized polytime algorithm that, given an ipABox $\mathcal{A}$ and an error bound $\epsilon > 0$, computes a real number x such that

$$\mathsf{Pr}\Big(\frac{|p(\mathcal{A},\mathcal{T} \models q) - x|}{p(\mathcal{A},\mathcal{T} \models q)} \leq \frac{1}{\epsilon}\Big) \geq \frac{3}{4}.$$

In words: with a high probability (the value of $\frac{3}{4}$ can be amplified by standard methods), the algorithm computes a result that deviates from the actual result by at most the factor $\frac{1}{\epsilon}$.

It follows from the proof of Theorem 2 and the fact that there is no FPRAS for the number of satisfying assignments of a propositional formula (unless the complexity classes RP and NP coincide, which is commonly assumed not to be the case) that, over pABoxes, there is no FPRAS for any CQ q and TBox $\mathcal{T}$. Thus, we again have to restrict ourselves to ipABoxes. As observed in [9], it is an easy consequence of a result of Karp and Luby [21] that there is an FPRAS for every CQ over tuple independent databases. By Theorem 1, there is thus also an FPRAS for every CQ q and DL-Lite TBox $\mathcal{T}$ over ipABoxes. The same is true for every FO-rewritable TBox formulated in $\mathcal{ELI}$ or any other TBox language. This observation clearly gives hope for the practical feasibility of probabilistic OBDA.

It is a natural question whether FPRASes also exist for (CQs and) TBoxes formulated in richer ontology languages. No general positive result can be expected for expressive DLs that involve all Boolean operators; the basic such DL is $\mathcal{ALC}$ with concept constructors $\neg C$, $C \sqcap D$, and $\exists r.C$, a typically well-behaved fragment of OWL DL. As analyzed in detail in [26], there is a large class of Boolean CQs q and $\mathcal{ALC}$-TBoxes $\mathcal{T}$ such that, given a non-probabilistic ABox $\mathcal{A}$, it is coNP-hard to check the entailment $\mathcal{A},\mathcal{T} \models q$. A computation problem whose decision version is coNP-hard cannot have an FPRAS [19], and thus we obtain the following.

Theorem 8. *There are CQs q and $\mathcal{ALC}$-TBoxes $\mathcal{T}$ such that there is no FPRAS for q and $\mathcal{T}$.*

In $\mathcal{ELI}$, entailment by non-probabilistic ABoxes can be checked in PTime for all CQs q and TBoxes $\mathcal{T}$. By what was said above, the interesting cases are those that involve a TBox which is not FO-rewritable. For example, answering the query $A(a)$ and TBox $\{\exists r.A \sqsubseteq A\}$ over ipABoxes roughly corresponds to a directed, two-terminal version of network reliability problems, for which FPRASes can be rather hard to find, see for example [20,34]. We leave a detailed analysis of FPRASes for (CQs q and) $\mathcal{ELI}$-TBoxes $\mathcal{T}$ as interesting future work. Ideally, one would like to have a full classification of all pairs $(q, \mathcal{T})$ according to whether or not an FPRAS exists.

7 Conclusion

We have introduced a framework for ontology-based access to probabilistic data that can be implemented using existing probabilistic database system, and we have analyzed the data complexity of computing answer probabilities in this framework. There are various opportunities for future work. For example, it would be interesting to extend the *concrete* dichotomy from the basic DL-Lite dialect studied in this paper to more expressive versions of DL-Lite that, for example, allow role hierarchy statements in the TBox. It would also be worthwhile to add probabilities to the TBox instead of admitting them only in the ABox; this is done for example in [27,12], but it remains to be seen whether the semantics used there is appropriate for our purposes. Finally, it would be interesting to study the existence of FPRASes for approximating answer probabilities when TBoxes are formulated in $\mathcal{ELI}$.

Acknowledgement This work was supported by the DFG project Prob-DL (LU1417/1-1).

References

1. Antova, L., Jansen, T., Koch, C., Olteanu, D.: Fast and simple relational processing of uncertain data. In: Proc. of ICDE. 983–992 (2008)
2. Antova, L., Koch, C., Olteanu, D.: 10^{10^6} worlds and beyond: efficient representation and processing of incomplete information. VLDB J. 18(5), 1021–1040 (2009)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press (2003)
4. Bienvenu, M., Lutz, C., Wolter, F.: Query containment in description logics reconsidered. In: Proc. of KR (2012)
5. Boulos, J., Dalvi, N.N., Mandhani, B., Mathur, S., Ré, C., Suciu, D.: MYSTIQ: a system for finding more answers by using probabilities. In: Proc. of SIGMOD. 891–893 (2005)
6. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reasoning 39(3), 385–429 (2007)
7. Dalvi, N.N., Ré, C., Suciu, D.: Probabilistic databases: diamonds in the dirt. Commun. ACM 52(7), 86–94 (2009)
8. Dalvi, N.N., Schnaitter, K., Suciu, D.: Computing query probability with incidence algebras. In: Proc. of PODS. 203–214. ACM (2010)
9. Dalvi, N.N., Suciu, D.: Efficient query evaluation on probabilistic databases. VLDB J. 16(4), 523–544 (2007)
10. Dalvi, N.N, Suciu, D.: The Dichotomy of Probabilistic Inference for Unions of Conjunctive Queries. submitted to Journal of the ACM.

11. Finger, M., Wassermann, R., Cozman, F.G.: Satisfiability in $\mathcal{EL}$ with sets of probabilistic ABoxes. Proc. of DL. CEUR-WS, Vol. 745 (2011)
12. Fuhr, N., Rölleke, T.: A probabilistic relational algebra for the integration of information retrieval and database systems. ACM Trans. Inf. Syst. 15(1), 32–66 (1997)
13. Furche, T., Gottlob, G., Grasso, G., Gunes, O., Guo, X., Kravchenko, A., Orsi, G., Schallhart, C., Sellers, A.J., Wang, C.: Diadem: domain-centric, intelligent, automated data extraction methodology. In Proc. of WWW. 267–270. ACM (2012)
14. Gottlob, G., Lukasiewicz, T., Simari, G.I.: CQ answering in probabilistic datalog+/− ontologies. In Proc. of RR. Vol. 6902 of LNCS, 77–92. Springer (2011)
15. Green, T.J., Tannen, V.: Models for incomplete and probabilistic information. IEEE Data Engineering Bulletin 29(1), 17–24 (2006)
16. Gupta, R., Sarawagi, S.: Creating probabilistic databases from information extraction models. In Proc. of VLDB. 965–976. ACM (2006)
17. Halpern, J.Y.: An analysis of first-order logics of probability. Artif. Intell. 46(3), 311–350 (1990)
18. Imielinski, T., Jr., W.L.: Incomplete information in relational databases. J. of the ACM 31(4), 761–791 (1984)
19. Jerrum, M., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. 43, 169–188 (1986)
20. Karger, D.R.: A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. SIAM J. Comput. 29(2), 492–514 (1999)
21. Karp, R.M., Luby, M.: Monte-carlo algorithms for enumeration and reliability problems. In Proc. of FoCS. 56–64. IEEE Computer Society (1983)
22. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to query answering in DL-Lite. In Proc. of KR. AAAI Press (2010)
23. Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S., Teixeira, J.S.: A brief survey of web data extraction tools. SIGMOD Record 31(2), 84–93 (2002)
24. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Sem. 6(4), 291–308 (2008)
25. Lutz, C., Schröder, L.: Probabilistic description logics for subjective uncertainty. In Proc. of KR. AAAI Press (2010)
26. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: Proc. of KR. AAAI Press (2012)
27. Raedt, L.D., Kimmig, A., Toivonen, H.: Problog: a probabilistic prolog and its application in link discovery. In Proc. of IJCAI. 2468–2473. AAAI Press (2007)
28. Rossmann, B.: Homomorphism preservation theorems. J. ACM. 55(3). 1–54 (2008).
29. Sarma, A.D., Benjelloun, O., Halevy, A.Y., Widom, J.: Working models for uncertain data. In: Proc. of ICDE. IEEE Computer Society (2006)
30. Straccia, U.: Top-k retrieval for ontology mediated access to relational databases. In: Information Sciences. 108, 1–23 (2012).
31. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
32. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comput. 8(3), 410–421 (1979)
33. Widom, J.: Trio: A system for integrated management of data, accuracy, and lineage. In Proc. of CIDR. 262–276 (2005)
34. Zenklusen, R., Laumanns, M.: High-confidence estimation of small s-t reliabilities in directed acyclic networks. Networks 57(4), 376–388 (2011)

A Proofs for Section 4

A.1 Theorem 4

Throughout the whole section, we assume CQs to be Boolean and connected, and to not contain individual names. In this section and the subsequent ones, we will sometimes make use of canonical models as defined in [22], introduced here as a preliminary. To construct the *canonical model* $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ of an ABox $\mathcal{A}$ and a DL-Lite TBox $\mathcal{T}$, we start with $\mathcal{A}$ viewed as an interpretation and then exhaustively apply the CIs from $\mathcal{T}$ as rules, introducing fresh elements for existential quantifiers. Formally, the domain of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ consists of paths of the form $aR_1 \cdots R_n$, $n \geq 0$, such that the following conditions hold:

(agen) $\mathcal{A}, \mathcal{T} \models \exists R_1(a)$ but $R_1(a,b) \notin \mathcal{A}$ for all $b \in \mathsf{Ind}(\mathcal{A})$
(rgen) for $i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$ (written $c_{R_i} \rightsquigarrow c_{R_{i+1}}$).

We denote by $\mathsf{tail}(\sigma)$ the last element in a path σ. Now, $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ is defined as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \{a \cdot c_{R_1} \cdots c_{R_n} \mid a \in \mathsf{Ind}(\mathcal{A}), n \geq 0, a \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n}\},\\
a^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= a, \text{ for all } a \in \mathsf{Ind}(\mathcal{A}),\\
A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \{\sigma \cdot R \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \mid \mathcal{T} \models \exists R^- \sqsubseteq A\},\\
P^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup\\
&\quad\ \{(\sigma, \sigma \cdot c_P) \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \times \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \mid \mathsf{tail}(\sigma) \rightsquigarrow c_P\} \cup\\
&\quad\ \{(\sigma \cdot c_{P^-}, \sigma) \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \times \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \mid \mathsf{tail}(\sigma) \rightsquigarrow c_{P^-}\},
\end{aligned}$$

where '$\cdot$' denotes concatenation. Domain elements in $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ that are true path (i.e., do not only consists of an individual name) are called *anonymous*. The following is the central property of canonical models.

Theorem 9 ([22]). *For every consistent DL-Lite KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and every CQ q, we have $\mathbf{a} \in \mathsf{cert}_{\mathcal{T}}(q, \mathcal{A})$ iff $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q[\mathbf{a}]$.*

We will sometimes also use canonical models $\mathcal{I}_{q,\mathcal{T}}$ for a CQ q and a TBox $\mathcal{T}$, defined as $\mathcal{I}_{\mathcal{A}_q,\mathcal{T}}$ where $\mathcal{A}_q$ is q viewed as an ABox, i.e., the variables in q are viewed as the ABox individuals of $\mathcal{A}_q$. In particular, we thus have

$$\Delta^{\mathcal{I}_{q,\mathcal{T}}} = \{x \cdot c_{R_1} \cdots c_{R_n} \mid x \in \mathsf{var}(q), n \geq 0, x \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n} \text{ in } \mathcal{A}_q\}.$$

The following is easy to prove.

Lemma 3 ([4]). *$q \sqsubseteq_{\mathcal{T}} q'$ iff there is a homomorphism from q' to $\mathcal{I}_{q,\mathcal{T}}$, i.e., a map $h : \mathsf{var}(q') \to \Delta^{\mathcal{I}_{q,\mathcal{T}}}$ such that the following conditions are satisfied:*

1. *if $A(t) \in q'$, then $h(t) \in A^{\mathcal{I}_{q,\mathcal{T}}}$;*
2. *if $r(t,t') \in q'$, then $(h(t), h(t')) \in r^{\mathcal{I}_{q,\mathcal{T}}}$.*

As explained in the proof sketch in the main paper, Theorem 4 is a consequence of the following lemma. We call an FO-rewriting of a CQ q and a TBox $\mathcal{T}$ that happens to be a UCQ a *UCQ-rewriting*.

Lemma 4. *Let q be a CQ and $\mathcal{T}$ a DL-Lite TBox such that q is $\mathcal{T}$-minimal. Then there is a UCQ-rewriting $q_\mathcal{T}$ of q relative to $\mathcal{T}$ that is reduced and such that q occurs as a disjunct in $q_\mathcal{T}$. Moreover, if q is a simple tree query, then so is every disjunct of $q_\mathcal{T}$.*

Proof. Let $q_\mathcal{T}$ be a UCQ-rewriting of q relative to $\mathcal{T}$. It is easily verified that standard approaches such as the one from [6] will generate a UCQ in which every disjunct is a simple tree query if q is a simple tree query. By definition of FO-rewritings, $q_\mathcal{T} \vee q$ is also an FO-rewriting and thus we can w.l.o.g. assume that $q_\mathcal{T}$ contains q as a disjunct. Since q is $\mathcal{T}$-minimal and thus minimal, we can further assume that every disjunct of $q_\mathcal{T}$ is minimal. We show that q is not eliminated when $q_\mathcal{T}$ is converted into reduced form, i.e., that for every disjunct q' of $q_\mathcal{T}$ with $q \sqsubseteq q'$, we have $q \equiv q'$. Choose such a disjunct q'. Using the definition of FO-rewritings, it is not hard to show that we must have $q' \sqsubseteq_\mathcal{T} q$.

We first consider the case where q has two root variables. Then q must be of the form $\{s_1(x,y),\dots,s_k(x,y)\}$, where the s_i are role names or inverse roles. If $k > 1$, then $q' \sqsubseteq_\mathcal{T} q$ implies that $q' = q$ or $q' = \{s_1(x,x),\dots,s_k(x,x)\}$. In both cases, we obtain $q' \sqsubseteq q$, which yields $q \equiv q'$. If $k = 1$, then q has the form $s(x,y)$ with $x \neq y$ and since $q \sqsubseteq q'$ and the CQ q' (like any CQ) cannot be empty, we have $q' = q$ and are done.

Now assume that q has only a single root variable. As a preliminary, we make the following observation.

Claim. If x' is a root variable of q', then there is a homomorphism from q' to q that maps x' to the unique root variable of q.

Proof of claim. Since $q \sqsubseteq q'$, there is a homomorphism g from q' to q. Let x' be a root variable in q' and x the root of q. We want to show that $g(x') = x$. Assume to the contrary that this is not the case. Then $g(x') = y \neq x$. Since q has only one root variable, y is not a root variable and q does neither contain concept atoms nor reflexive role atoms on y. Since g is a homomorphism, q' thus does not contain concept atoms or reflexive role atoms on x'. Since q and q' are simple tree queries, all variables from $\mathsf{var}(q') \setminus \{x'\}$ are connected via some role atom to x' in q' and the only variable in q connected via some role atom to y is the root x. Therefore, we have $g(y') = x$ for all $y' \in \mathsf{var}(q') \setminus \{x'\}$.

- There is a role R such that for all atoms $S(y',x')$ in q', we have $R = S$. Since q' is minimal, it then has the form $R(y',x')$. Due to the existence of g, there is an atom $R(x,y)$ in q. Since $q' \subseteq_\mathcal{T} q$, we have $R(x,y) \sqsubseteq_\mathcal{T} q$. Since q is $\mathcal{T}$-minimal, this yields $q = R(x,y)$. This is a contradiction to q having only one root variable.
- There are atoms $R_1(y_1',x')$ and $R_2(y_2',x')$ in q' such that $R_1 \neq R_2$. For all atoms $R(y',x')$ in q', we have $R(x,y) \in q$ since $g(x') = y$ and $g(y') = x$. Since there is more than one role name R that occurs in such an atom, the restriction of q to the query q_y containing those atoms from q that involve y is not tree-shaped. Since $q' \sqsubseteq_\mathcal{T} q$, there is a homomorphism h from q to $\mathcal{I}_{q',\mathcal{T}}$. By construction of the latter and since q_y is not tree-shaped, the restriction of h to q_y is a homomorphism from q_y to q'. Thus, there is a

y'_0 in q' such that for all atoms $R(y', x') \in q'$, we have $R(y'_0, x') \in q'$. Since q' is minimal, it thus cannot contain any other variables than y'_0 and x'. Since y is not a root variable of q, there must be at least one atom in q that involves x, but not y. Let q^- be obtained from q by dropping all such atoms. Then g is also a homomorphism from q' to q^-. By construction of $\mathcal{I}_{q^-,\mathcal{T}}$, we find a homomorphism g^+ from $\mathcal{I}_{q',\mathcal{T}}$ to $\mathcal{I}_{q^-,\mathcal{T}}$ with $g^+(x') = y$ and $g^+(y'_0) = x$. Composing the homomorphism h, restricted to q^-, with g^+ yields a homomorphism from q to $\mathcal{I}_{q^-,\mathcal{T}}$. Consequently, $q' \sqsubseteq_{\mathcal{T}} q$ in contradiction to the minimality of q.

This finishes the proof of the claim.

Since $q' \sqsubseteq_{\mathcal{T}} q$, there is a homomorphism h from q to $\mathcal{I}_{q',\mathcal{T}}$. Define the subset $Z \subseteq q'$ of atoms that are 'hit by h' as

$$\begin{aligned} Z = \{&A(x') \in q' \mid \exists A(x) \in q : h(x) = x'\} \cup \\ &\{r(x', y') \in q' \mid \exists r(x, y) \in q : h(x) = x' \wedge h(y) = y'\}. \end{aligned}$$

We distinguish the following cases.

- $A(x) \in q$ implies $A(h(x)) \in Z$ and $r(x, y) \in q$ implies $r(h(x), h(y)) \in Z$. Then $q' \sqsubseteq q$ and thus $q \equiv q'$. Consequently, we are done.
- There is an $A(x) \in q$ with $A(h(x)) \notin Z$. Then x is the root variable in the simple tree query q. Since $h(x) \in A^{\mathcal{I}_{q,\mathcal{T}}}$, one of the following cases applies:
  1. $h(x)$ is anonymous in $\mathcal{I}_{q,\mathcal{T}}$. Let $h(x) = x' c_{R_1} \cdots c_{R_k}$ with $k > 0$. Since q is a simple tree query, it follows that h maps all variables in q to the subtree of $\mathcal{I}_{q',\mathcal{T}}$ rooted at x', i.e., all elements in the range of h have x' as a (not necessarily proper) prefix. Moreover, since the root variable x is not mapped to x' and all concept atoms in q involve x, there is even a match of q in the subtree rooted in $\mathcal{I}_{q',\mathcal{T}}$ at x' when all concept atoms at the root of that subtree are removed. By construction of $\mathcal{I}_{q',\mathcal{T}}$, there is thus a single atom at in q' such that q has a match in $\mathcal{I}_{\mathsf{at},\mathcal{T}}$. Since $q \sqsubseteq q'$, we must find at also in q, modulo renaming of variables. Since q is $\mathcal{T}$-minimal, this implies $q = \{\mathsf{at}\}$. Since $q \sqsubseteq q'$ and q' is non-empty and minimal, we must have $q' = \mathsf{at}$, thus $q = q'$ and we are done.
  2. There is a $B(h(x)) \in q'$ such that $\mathcal{T} \models B \sqsubseteq A$. Then $h(x)$ is a root variable of q'. By the above claim, there is a homomorphism g from q' to q with $g(h(x)) = x$ and thus $B(x) \in q$. Since $\mathcal{T} \models B \sqsubseteq A$, this is a contradiction against $\mathcal{T}$-minimality of q and the fact that $A(x) \in q$.
  3. There is an $R(h(x), y') \in q'$ such that $\mathcal{T} \models \exists R \sqsubseteq A$. If $h(x)$ is a root variable of q', we can argue as in the previous case. Otherwise, h must map all variables in q to $h(x)$, a root variable y' of q' with $S(y', h(x)) \in q'$ for at least one S, and to the anonymous subtree rooted at $h(x)$. Let $X' = \{R_1(y', h(x)), \ldots R_m(y', h(x))\}$ be all assertions

in q' of this form. By construction of $\mathcal{I}_{q',\mathcal{T}}$, there is a match of q in $\mathcal{I}_{X',\mathcal{T}}$ and thus $X', \mathcal{T} \models q$. Since $q \sqsubseteq q'$, we find X' as a subquery X in q, probably after identifying the two variables in X. Since $X, \mathcal{T} \models q$, this is a contradiction against $\mathcal{T}$-minimality of q and the fact that $A(x) \in q \setminus X$.

- There is an $R(x,y) \in q$ with $R(h(x), h(y)) \notin Z$.
  Then either x or y is a root variable. We assume w.l.o.g. that x is the root variable, the other case is symmetric. Make a case distinction as follows:
  1. $h(x)$ is anonymous in $\mathcal{I}_{q,\mathcal{T}}$.
     Then we can argue as in Case 1 above.
  2. $h(x)$ is a root variable of q'.
     Since $(h(x), h(y)) \in R^{\mathcal{I}_{q,\mathcal{T}}} \setminus Z$ and $h(x)$ is not anonymous, $h(y)$ must be anonymous by construction of $\mathcal{I}_{q',\mathcal{T}}$. Thus, $(h(x), h(y)) \in S^{\mathcal{I}_{q',\mathcal{T}}}$ implies $S = R$, which yields that $S(x,y) \in q$ implies $S = R$. Moreover, q does not contain any concept atoms on y since y is not a root variable and thus, in summary, y is not involved in any other atoms than $R(x,y)$ in q. By the claim, there is a homomorphism g from q' to q with $g(h(x)) = x$. By construction of $\mathcal{I}_{q',\mathcal{T}}$, $(h(x), h(y)) \in R^{\mathcal{I}_{q',\mathcal{T}}}$ with $h(y)$ anonymous implies that there is no atom $R(h(x), z)$ in q'. Thus, g is still a homomorphism from q' to $q^- := q \setminus \{R(x,y)\}$. Consequently, there is an outgoing R-edge from x in $\mathcal{I}_{q^-,\mathcal{T}}$. Thus and since y is not involved in any other atoms than $R(x,y)$ in q, we have $q^- \sqsubseteq_{\mathcal{T}} q$, in contradiction to the $\mathcal{T}$-minimality of q.
  3. $h(x)$ is a non-root variable of q'.
     Then, h must map all variables in q to $h(x)$, a root variable y' of q' with $S(y', h(x)) \in q'$ for at least one S, and to the anonymous subtree rooted at $h(x)$. Let $X' = \{R_1(y', h(x)), \ldots R_m(y', h(x))\}$ be all assertions in q' of this form. By Since $(h(x), h(y)) \in R^{\mathcal{I}_{q,\mathcal{T}}} \setminus Z$ and $h(x)$ is not anonymous, $h(y)$ must be anonymous. By construction of $\mathcal{I}_{q',\mathcal{T}}$, R^- is thus not among the $R_1, \ldots, R_m$. Moreover, there is a match of q in $\mathcal{I}_{X',\mathcal{T}}$ and thus $X', \mathcal{T} \models q$. We can argue as in the previous case that y is not involved in any other atoms than $R(x,y)$ in q. Since $q \sqsubseteq q'$, we find X' as a subquery X of q, probably after identifying the two variables in X'. Clearly $X, \mathcal{T} \models q$. If y does not occur in X, then $X \subsetneq q$, in contradiction to the $\mathcal{T}$-minimality of q. Otherwise, since R^- is not among the $R_1, \ldots, R_m$ and y occurs in no other atoms than $R(x,y)$ in q, we must have $X' = R(y', h(x))$. From $X', \mathcal{T} \models q$, we thus obtain $\{R(x,y)\}, \mathcal{T} \models q$. Since q is $\mathcal{T}$-minimal and $R(x,y) \in q$, we must have $q = R(x,y)$. This is a contradiction to q having only one root variable.

❑

A.2 Theorem 5

To prove Theorem 5, we analyze the results in [8,10]. As a preparation for that proof, we first introduce some notions and results, all taken from [10], to which we also refer for further explanations and examples.

A *disjunctive sentence* is a disjunction of *connected* CQs. A disjunctive sentence $q = q_1 \vee \cdots \vee q_k$ is *symbol-connected* if the graph (V, E) with

- $V = \{q_1, \ldots, q_k\}$
- $(q_i, q_j) \in E$ iff there is a relational symbol that occurs both in q_i and q_j

has precisely one connected component. A *root variable* of a UCQ q is a variable that occurs in all atoms of q. A root variable x of q is a *separator variable* of q if additionally for every relation name r that occurs in the query, there is a number i_r such that every atom in q with symbol r contains exactly one occurrence of x and that is on position i_r.

Let $x_1, \ldots, x_n$ be a set of variables. A set Θ of predicates of the form $x_i < x_j$ is called *consistent* if there is a total order $<^*$ such that $x_i < x_j \in \Theta$ implies $x_i <^* x_j$. A CQ q is *ranked* if the set

$$\{x < y \mid r(x, y) \in q\}$$

is consistent.

In what follows, we assume the disjunctive sentence q to be reduced and all disjuncts p of q to be minimal, i.e., there is no $p' \subsetneq p$ with $p' \sqsubseteq p$. We call a disjunctive sentence q *immediately unsafe* if it is symbol-connected, contains at least one variable, and does not have a separator.

Theorem 10 (Theorem 4.4 of ??). *If a disjunctive query q is ranked and immediately unsafe, then computing $P(q)$ is* #P*-hard.*

The following also follows from this theorem and the algorithm in **??**.

Corollary 1. *If a disjunctive query q is ranked and does not have a separator, then computing $P(q)$ is* #P*-hard.*

Proof. We can write q as $q_1 \vee \ldots \vee q_k$ where $q_1, \ldots, q_k$ are symbol-connected disjunctive sentences. The algorithm in **??** computes $P(q)$ as $1 - (1 - P(q_1)) \cdot \ldots \cdot (1 - P(q_k))$. Since q is not ranked, there exists i such that q_i does not have a separator. Hence, the algorithm fails in the computation of $P(q_i)$ because of Theorem 10.

❏

In **??**, the following is proven (note that we use slightly different notation, e.g., D denotes a tuple-independent probabilistic database).

Proposition 1. *Every UCQ q is computationally equivalent to a ranked UCQ $\overline{q}$ over an extended vocabulary, i.e., the problem "given a probabilistic database D, compute $P_D(q)$" can be reduced in polynomial time to "given $\overline{D}$ compute $P_{\overline{D}}(\overline{q})$", and vice versa.*

Since the construction of $\overline{q}$ in the proof of this Proposition is needed later, we present it here for the binary case. We define three functions $\tau_=$, $\tau_<$, $\tau_>$ as follows:

$$\begin{aligned}
\tau_=(x_1, x_2) &= (x_1) \\
\tau_<(x_1, x_2) &= (x_1, x_2) \\
\tau_>(x_1, x_2) &= (x_2, x_1)
\end{aligned}$$

Additionally, introduce for every binary relation symbol R occurring in q three fresh relation symbols $R^{=}$ (which is unary), $R^{<}$, and $R^{>}$ (both binary). Finally define $\Omega = \{=, <, >\}$.

If $p = p_0, R_1(y_1, z_1), \ldots, R_m(y_m, z_m)$ is a CQ in q such that p_0 contains precisely the unary atoms of p and $\sim_1, \ldots, \sim_m \in \Omega$ we call the CQ

$$p_0, \bigwedge_{i=1}^{m} R_i^{\sim_i}(\tau_{\sim_i}(y_i, z_i)), y_i \sim_i z_i \tag{1}$$

the $\sim_1, \ldots, \sim_m$*-reduct of* p. We call the reduct *strict* if $\sim_i \in \{<, >\}$ for all $i \in \{1, \ldots, m\}$.

The ranking $\overline{q}$ of q is now the disjunction of all $\sim_1, \ldots, \sim_m$-reducts of some disjunct p in q and $\sim_1, \ldots, \sim_m \in \Omega$. Obviously, the set of appended order predicates $y_i \sim_i z_i$ might be inconsistent and therefore the corresponding reduct can be dropped from the ranking. However, it is easy to see that there is a consistent reduct of every disjunct p of a UCQ q. Moreover, there is even a consistent strict reduct of every p in q when one $\sim_i$ is fixed (to $<$ or $>$). We prove that strict reducts are not removed during the reduction of the obtained UCQ.

Lemma 5. *If q is a reduced UCQ and p' a strict $\sim_1, \ldots, \sim_m$-reduct of some CQ p in q. Then p' is contained in the ranking $\overline{q}$ of q, i.e., p' is not removed during the reduction $\overline{q}$.*

Proof. Suppose to the contrary that there is some CQ p_1 in q and $\sim'_1, \ldots, \sim'_k \in \Omega$ such that the $\sim'_1, \ldots, \sim'_k$-reduct p'_1 of p_1 satisfies $p'_1 \sqsubseteq p'$ and $p'_1 \neq p'$. Thus, there is a homomorphism h from p' to p'_1. We claim that h is also a homomorphism from p to p_1. Let $p = p_0, R_1(y_1, z_1), \ldots, R_m(y_m, z_m)$ such that p_0 contains precisely the unary atoms of p. Then we have:

- If $A(x) \in p$, formula (1) implies that $A(x) \in p'$ and since h is homomorphism, we have also $A(h(x)) \in p'_1$. Using again formula (1) yields $A(h(x)) \in p_1$.
- Let $R_i(x, y) \in p$. Since p' is a strict reduct, we have to distinguish two cases:
  - Let $\sim_i = <$. Then $R_i^{<}(x, y) \in p'$ and since h is a homomorphism, we have that $R_i^{<}(h(x), h(y)) \in p'_1$. By formula (1), $R_i(h(x), h(y)) \in p_1$.
  - Let $\sim_i = >$. Then $R_i^{>}(y, x) \in p'$ and since h is a homomorphism, we have that $R_i^{>}(h(y), h(x)) \in p'_1$. By formula (1), $R_i(h(x), h(y)) \in p_1$.

Because of the homomorphism from p to p_1, we obtain $p_1 \sqsubseteq p$. Since q is reduced this yields $p_1 = p$ and minimality of p implies that the homomorphism h is surjective (from p to p_1). Hence, h is also a surjective homomorphism from p' to p'_1 ❑

With the next lemma we provide an intuition why the notion of "being $\mathcal{T}$-generated" is central in the presence of rewritings.

Lemma 6. *Let q be a simple tree query and $\mathcal{T}$ a DL-Lite TBox such that q is $\mathcal{T}$-minimal. Then there exists a reduced UCQ-rewriting $q_{\mathcal{T}}$ of q relative to $\mathcal{T}$ such that the following conditions are satisfied:*

1. *every disjunct of $q_\mathcal{T}$ is a simple tree query;*
2. *q is a disjunct in $q_\mathcal{T}$;*
3. *for every role R the following are equivalent*
   - *R is $\mathcal{T}$-generated in q or $\{R(x,y)\} \sqsubseteq_\mathcal{T} q$*
   - *there exists a disjunct q' in $q_\mathcal{T}$ such that q' contains an atom $R(x,y)$ where x is a root variable of q' and $x \neq y$*

Proof. Let q be a simple tree query and $\mathcal{T}$ a DL-Lite TBox. The reduced UCQ-rewriting $q_\mathcal{T}$ whose existence is guaranteed by Lemma 4 satisfies Conditions 1 and 2 of Lemma 6. It thus remains to show that it also satisfies Condition 3. We deal with the two directions separately.
"$\Rightarrow$": If $\{R(x,y)\} \sqsubseteq_\mathcal{T} q$, we can set $q'_\mathcal{T} \equiv q_\mathcal{T} \vee \{R(x,y)\}$ and it is clear that the disjunct $\{R(x,y)\}$ is not removed during reduction of $q'_\mathcal{T}$. If R is a role that is $\mathcal{T}$-generated in q, then one of the following cases applies:

- $R(x,y) \in q$, x is a root variable, and $x \neq y$.
  By choice of $q_\mathcal{T}$, q is a disjunct of $q_\mathcal{T}$ and we are done.
- $A(x) \in q$ and $\mathcal{T} \models \exists R \sqsubseteq A$.
  Let $q_R = (q \setminus \{A(x)\}) \cup \{R(x,y)\}$ with y a fresh variable. Clearly, we have $\mathcal{A}_{q_R}, \mathcal{T} \models q$, thus $\mathcal{I}_{\mathcal{A}_{q_R}} \models q_\mathcal{T}$. Let q' be a disjunct of $q_\mathcal{T}$ with $\mathcal{I}_{\mathcal{A}_{q_R}} \models q'$. Then $q_R \sqsubseteq q'$ and we find a homomorphism h from q' to q_R.
  We show that there must be an atom $R(x',y') \in q'$ with $h(x') = x$, $h(y') = y$. Assume to the contrary that this is not the case. Let $q^- = q_R \setminus \{R(x,y)\}$ and observe that, since $A(x) \notin q^-$, we have $q^- \subsetneq q$. Then h is also a homomorphism from q' to q^- and thus $q^- \sqsubseteq q'$ and also $q \sqsubseteq q'$. Since $q_\mathcal{T}$ is reduced and both q and q' are disjuncts of $q_\mathcal{T}$, it follows from the latter that $q = q'$. Thus $q^- \sqsubseteq q'$ means in fact $q^- \sqsubseteq q$, which is a contradiction to the minimality of q.
  Consider the atom $R(x',y') \in q'$ with $h(x') = x$, $h(y') = y$ whose existence we have just proved. Clearly, $x' \neq y'$. Thus, it remains to show that x' is a root variable in q'. Assume to the contrary that x' is not a root variable. Then y' must be a root variable and there must be an atom in q' that involves y', but not x'. This implies that q contains an atom that involves y, but not x, in contradiction to x being a root variable of q.
- $S(x,y) \in q$ such that y occurs only once and $\mathcal{T} \models \exists S \sqsubseteq \exists R$.
  Identical to previous case.

"$\Leftarrow$": Assume that there is a disjunct q' in $q_\mathcal{T}$ that contains $R(x,y)$ and x is a root variable and $x \neq y$. We can assume the following:

(A) there is no $q'' \subsetneq q'$ with $q'' \sqsubseteq_\mathcal{T} q$.
(B) $(q' \setminus \{S(x,z)\}) \cup \{S(x,z')\} \not\sqsubseteq_\mathcal{T} q$ where
  - x is root variable in q',
  - z occurs more than once, and
  - z' is a fresh variable.

If Point (A) is not satisfied, set $q'_{\mathcal{T}} = q_{\mathcal{T}} \vee q'$. It is clear that $q'_{\mathcal{T}} \equiv q_{\mathcal{T}}$ and that during reduction of $q'_{\mathcal{T}}$ the disjunct q_i will be removed since $q_i \sqsubseteq q'_i$. However, q'_i will not be removed since q_i was also not removed from $q_{\mathcal{T}}$. If Point (B) is not satisfied, the same arguments apply.

We show that R is $\mathcal{T}$-generated in q. Since $q' \sqsubseteq_{\mathcal{T}} q$, Lemma 3 implies the existence of a homomorphism h from q to $\mathcal{I}_{q',\mathcal{T}}$. We distinguish two cases:

- Assume that q' is of the form $\{S_1(x,y), \ldots, S_k(x,y)\}$, where the S_i are role names or inverse roles and R is among them. Hence, both x and y are root variables.
    - If q has two root variables and $k > 1$, it is clear that $q \equiv q'$, because of Points (A) and (B). Hence, R is $\mathcal{T}$-generated in q. If $k = 1$, then $\{R(x,y)\} = q' \sqsubseteq_{\mathcal{T}} q$.
    - If q has one root variable x_r, assume first that $h(x_r) \notin \{x,y\}$, i.e., x_r is mapped to some anonymous element of $\mathcal{I}_{q_1,\mathcal{T}}$. This clearly gives a contradiction to Point (B) since q is a simple tree query and, thus, not both x and y are in the image of h. Hence, we have $h(x_r) \in \{x,y\}$. Assume w.l.o.g. that $h(x_r) = x$. By Point (A), none of the atoms $S_i(x,y)$ can be dropped from q'. By Point (B), there has to be *one* variable $z \in \mathsf{var}(q)$ such that $h(z) = y$ and $S_i(x_r,z) \in q$ for all $i \in \{1,\ldots,k\}$. Hence we obtain $q \sqsubseteq q'$. By Lemma 4, q is a disjunct in $q_{\mathcal{T}}$. Since $q_{\mathcal{T}}$ is reduced, this implies $q \equiv q'$. Thus, q has two root variables, contradiction.
- Assume that q' contains an atom at^* that involves x but not y. Thus, x is a root variable in q' and y is not. We distinguish cases regarding the root variable(s) of q. In the first two cases we identify atoms at that can be dropped from q' such that still $q' \setminus \{at\} \sqsubseteq_{\mathcal{T}} q$. However, this contradicts Point (A).
    - q has two root variables, i.e., it is of the form $\{S_1(x_r,y_r), \ldots, S_k(x_r,y_r)\}$. Let first be $k > 1$. If $(h(x_r), h(y_r)) \in \{(x,y),(y,x)\}$, then at^* can be dropped. Otherwise, we can drop $R(x,y)$.
      If $k = 1$ and $(h(x_r), h(y_r) \in \{(x,y),(y,x)\}$, then at^* can be dropped. Otherwise, if either $h(x_r)$ or $h(y_r)$ are in the anonymous part of $\mathcal{I}_{q',\mathcal{T}}$, then one can drop all but one atoms from q', i.e., either at^* or $R(x,y)$. If $S_1(h(x_r), h(y_r)) \in q'$, one can drop all atoms except for $S_1(h(x_r), h(y_r))$.
    - q has precisely one root variable x_r and h maps x_r to $y' \neq x$. If $y' = y \cdot c_{R_1} \cdots c_{R_n}$ (possibly $n = 0$), then at^* can be dropped from q_1, since q is a simple tree query. Analogously, if $y' \neq y \cdot c_{R_1} \cdots c_{R_n}$, one can drop all but one atoms from q', i.e., one of at^* or $R(x,y)$ can be omitted.
    - The homomorphism h maps the root variable x_r of q to x. If there is $R(x_r,z) \in q$ such that $h(z) = y$, then $z \neq x_r$, and thus R is $\mathcal{T}$-generated in q. If not, there has to be either an atom $B(x_r) \in q$ and $\mathcal{T} \models \exists R \sqsubseteq B$ or an atom $S(x_r,z) \in q$ such that $z \neq x_r$ occurs only once and $\mathcal{T} \models \exists R \sqsubseteq \exists S$, otherwise Point (A) from above is not satisfied, as $R(x,y)$ can be dropped from q'. In both cases R is $\mathcal{T}$-generated in q.

❏

We are finally ready to prove Theorem 5 which we state again for completeness.

Theorem 5 Let $\mathcal{T}$ be a DL-Lite-TBox. A $\mathcal{T}$-minimal CQ q is in PTIME relative to $\mathcal{T}$ iff

1. q is a simple tree query, and
2. if r and r^- are $\mathcal{T}$-generated in q, then $\{r(x,y)\} \sqsubseteq_\mathcal{T} q$ or q is of the form $\{S_1(x,y),\ldots,S_n(x,y)\}$.

Otherwise, q is #P-hard relative to $\mathcal{T}$.

Proof. "$\Rightarrow$": We show the contrapositive. Assume first that q is not a simple tree query. Then Theorem 4 implies that answering q relative to $\mathcal{T}$ is #P-hard. Assume now that q is a simple tree query but condition 2 from Theorem 5 is not satisfied, i.e., there is a role r such that both r and r^- can be generated in q w.r.t. $\mathcal{T}$ but (i) $\{r(x,y)\} \not\sqsubseteq_\mathcal{T} q$ and (ii) $q \neq \{S_1(x,y),\ldots,S_n(x,y)\}$. By Point 1 of Lemma 6, we can view $q_\mathcal{T}$ as a query where one variable x_r occurs in all atoms. We make a case distinction on how r and r^- are generated.

- $\{r(x_r,y), r(y,x_r)\} \subseteq q$. W.l.o.g. we can assume that q has to contain an atom at that involves x_r but not y because of (ii). We assume here that $\mathsf{at} = A(x_r)$, the case where $\mathsf{at} = s(x_r,y')$ is treated analogously. By Lemma 4, we can assume that q is contained in $q_\mathcal{T}$. Since x_r is root variable of q but no separator, the same holds for $q_\mathcal{T}$. Because $q_\mathcal{T}$ is not ranked, Corollary 1 is not applicable. However, due to the construction in (the proof of) Proposition 1, the ranking $\overline{q_\mathcal{T}}$ of $q_\mathcal{T}$ contains strict reducts q_1, q_2 of q such that

  $$\{r^<(x_r,y_1), r^>(x_r,y_1), x_r < y_1, A(x_r)\} \subseteq q_1$$
  $$\{r^>(y_2,x_r), r^<(y_2,x_r), x_r > y_2, A(x_r)\} \subseteq q_2$$

  According to Lemma 5, q_1 and q_2 are not removed during the reduction of $\overline{q_\mathcal{T}}$. Moreover, it is clear that $q_1 \not\equiv q_2$. Observe now that x_r is still root variable but no separator, e.g., in $r^<$ it occurs both in position 1 and 2. Now, the application of Corollary 1 to (the reduction of) $\overline{q_\mathcal{T}}$ yields #P-hardness of q.
- Assume $\{r(x_r,y), r(y,x_r)\} \not\subseteq q$. Since both r and r^- are $\mathcal{T}$-generated, Point 3 of Lemma 6 implies the existence of disjuncts q_1, q_2 in $q_\mathcal{T}$ such that
  - q_1 contains $r(x_r,y_1)$ and $y_1 \neq x$, and
  - q_2 contains $r(y_2,x_r)$ and $y_2 \neq x$.

  By definition of $q_\mathcal{T}$ we have that $q_i \sqsubseteq_\mathcal{T} q$, hence $q_1 \neq \{r(x_r,y_1)\}$ and $q_2 \neq \{r(y_2,x_r)\}$ by (i). We argue first that y_1 is not a root variable of q_1: If it is, q_1 is of the form $\{R_1(x_r,y_1),\ldots,R_m(x_r,y_1)\}$ with $m > 1$ because of the previous argument. Since $q_1 \sqsubseteq_\mathcal{T} q$, there is a homomorphism h from q to $\mathcal{I}_{q_1,\mathcal{T}}$. W.l.o.g. we can assume that q_1 satisfies Points (A) and (B) from the proof of Lemma 6. If $m > 1$, then h has to 'hit' all atoms $R_1(x_r,y_1),\ldots,R_m(x_r,y_1)$. This yields $q_1 \sqsubseteq q$ and thus $q_1 \equiv q$, since $q_\mathcal{T}$ is reduced. But this is in contradiction with (ii). The same arguments apply to variable y_2 in q_2.

It is clear that x_r is not a separator because it occurs in position 1 and 2 for r. If $q_\mathcal{T}$ is ranked then it is #P-hard, by Corollary 1. If $q_\mathcal{T}$ is not ranked, consider its ranking $\overline{q_\mathcal{T}}$. The construction in (the proof of) Proposition 1 together with Lemma 5 imply the existence of disjuncts q_1', q_2' such that

- $\{r^<(x_r, y_1), x < y_1\} \subseteq q_1'$ and
- $\{r^<(y_2, x_r), y_2 < x\} \subseteq q_2'$

where in both queries x_r is a root variable but y_1 and y_2 are not. Thus, $\overline{q_\mathcal{T}}$ does not have a separator and Corollary 1 implies #P-hardness of $\overline{q_\mathcal{T}}$ and thus of q.

"$\Leftarrow$". Assume that q and $\mathcal{T}$ satisfy Conditions 1 and 2 from the Theorem. By Theorem 1, it suffices to show that, for the FO-rewriting $q_\mathcal{T}$ of q relative to $\mathcal{T}$, answer probabilities over tuple-independent databases can be computed in PTime. Hence, let D be the ipABox $\mathcal{A}$ viewed as a tuple-independent database. For a Boolean CQ p, we write $p_D(p)$ instead of $p^d_D(() \in p)$ and omit D whenever it is clear from the context.

By Point 2 of Lemma 6, we may assume that $q_\mathcal{T}$ is a UCQ in which all disjuncts are simple tree queries. Hence, we can assume w.l.o.g. that there is a variable x_r occurring in all atoms of $q_\mathcal{T}$. Lemma 6 helps us also to prove the following useful claim.

Claim. Let r be a role name such that either r or r^- is not $\mathcal{T}$-generated in q. There are no two disjuncts q_1 and q_2 of $q_\mathcal{T}$ such that $r(x_r, y_1) \in q_1$ with $x_r \neq y_1$ and $r(y_2, x_r) \in q_2$ with $x_r \neq y_2$.

Proof of Claim. Assume to the contrary that there are such q_1, q_2. By Point 3 of Lemma 6, we have that both r and r^- are $\mathcal{T}$-generated in q or $\{r(x,y)\} \sqsubseteq_\mathcal{T} q$. The former is excluded by our assumption, hence the latter is the case. Set $q'_\mathcal{T} = q_\mathcal{T} \vee \{r(x,y)\}$. It is easy to see that $q'_\mathcal{T} \equiv q_\mathcal{T}$ and that during reduction of $q'_\mathcal{T}$ the added disjunct is not removed but q_1 and q_2 are (unless they are equivalent to $\{r(x,y)\}$). This finishes the proof of the claim.

Now, we distinguish three cases.

Case (i). For all role names r, either r or r^- is not $\mathcal{T}$-generated in q. The above claim provides a necessary condition for x_r being a separator. However, it is not yet sufficient, since there might be 'reflexive' atoms of the form $r(x_r, x_r)$ occurring in $q_\mathcal{T}$. We show that these atoms can be removed in the following way. If $\sim$ is an equivalence relation over $\mathsf{var}(q')$ and at an atom, we define $\mathsf{at}_\sim$ by taking

$$A(x)_\sim = A(x)$$
$$R(x,y)_\sim = \begin{cases} R^{\neq}(x,y) & \text{if } x \nsim y \\ R^{=}(x) & \text{if } x \sim y \end{cases}$$

Denote with $\mathsf{EQ}(q')$ the set of all equivalence relations over $\mathsf{var}(q')$. Then, define $q'_\mathcal{T}$ as

$$\bigvee_{q' \in q_\mathcal{T}} \bigvee_{\mathsf{EQ}(q')} \{\mathsf{at}_\sim \mid \mathsf{at} \in q'\}$$

It is not hard to prove that D can be transformed in polynomial time into some D' such that $P_D(q_\mathcal{T}) = P_{D'}(q'_\mathcal{T})$. Moreover, it should be clear that the statement of the claim remains true after this transformation. Finally observe that x_r is a separator variable in $q'_\mathcal{T}$.

We now perform the following steps in order to compute $p_{D'}(q'_\mathcal{T})$:

- If $q'_\mathcal{T}$ is not symbol-connected and its symbol-connected components are $q_1, \ldots, q_k$, compute:
$$P(q'_\mathcal{T}) = 1 - (1 - P(q_1)) \cdot \ldots \cdot (1 - P(q_k))$$
- Continue with q_i, which is clearly symbol-connected and, by construction, still has the separator x_r. Compute
$$P(q_i) = 1 - \prod_{a \in \mathsf{Ind}(\mathcal{A})} (1 - P(q_i[a/x_r])$$
- The query $q_i[a/x_r]$ comprises disjuncts q'_i with $\mathsf{var}(q'_i) = \{z_1, \ldots, z_m\}$ of the form
$$A_1(a) \wedge \ldots \wedge A_n(a) \wedge q'_{i1} \wedge \ldots q'_{im} \qquad (\star)$$
where each q'_{ij} is obtained from q'_i by restricting to all atoms that involve z_j, i.e., q'_{ij} is of the form $S_1(a, z_j), \ldots, S_\ell(a, z_j)$.
In order to compute $p(q_i[a/x_r])$, we apply the distributivity law and distribute the '$\wedge$'s in $(\star)$ over the (outer) disjunctions. We obtain a conjunction of N disjunctive sentences $q_\wedge = p_1 \wedge \ldots \wedge p_N$. The restricted form of the conjuncts in $(\star)$ implies that each p_i is a disjunction of CQs of the form $A(a)$ or $S_1(a, y), \ldots, S_\ell(a, y)$.
- The probability $p(q_\wedge)$ can be computed using the (dual) inclusion/exclusion principle as follows:
$$p(q_\wedge) = \sum_{\emptyset \subset Y \subseteq [N]} (-1)^{|Y|+1} P(q_Y)$$
where $[N] = \{1, \ldots, N\}$ and $q_Y = \bigvee_{y \in Y} p_y$. Note that q_Y is a disjunctive sentence for all $Y \subseteq [N]$.
- Let us analyze the structure of q_Y for some (non-empty) $Y \subseteq [N]$. Since its disjuncts are obtained from some combination of the p_i, q_Y can be partitioned into $q_{A_1}, \ldots, q_{A_k}$, and $q_{Y'}$ such that
  - $q_{A_i} = A_i(a)$ and
  - $q_{Y'}$ contains only CQs of the form $S_1(a, y), \ldots, S_\ell(a, y)$

  In tuple-independent databases, the disjuncts of the former form are pairwise independent of each other and independent of disjuncts of the latter form, so compute
$$p(q_Y) = 1 - (1 - p(q_{Y'})) \cdot \prod_{i=1}^{k} (1 - p(q_{A_i}))$$
- Observe that $p(q_{A_i})$ can be read off from D for all i.

- For computing $p(q_{Y'})$ observe that $q_{Y'}$ can be viewed as a query with a separator z, since every atom has precisely one free variable. Hence, we can compute
$$p(q_{Y'}) = 1 - \prod_{b \in \mathsf{Ind}(\mathcal{A})} (1 - p(q_{Y'}[b/z]))$$
- The query $q_{Y'}[b/z]$ consists of disjuncts of the form $S_1(a,b) \wedge \ldots \wedge S_\ell(a,b)$. Again, distribute the $\wedge$'s over the outer disjunctions to obtain a conjunction of disjunctive sentences where each disjunctive sentence comprises only *one* atom $S(a,b)$ and apply the (dual) inclusion/exclusion principle. Each disjunctive sentence q_d obtained in this way is of the form $S_1(a,b) \vee \ldots \vee S_n(a,b)$. Hence, its probability can be computed as
$$p(q_d) = 1 - (1 - p(S_1(a,b)) \cdots (1 - p(S_n(a,b))))$$
which can be read off from D.

Case (ii). There is a role name r such that both r and r^- is $\mathcal{T}$-generated in q, but q is not of the form $\{S_1(x,y), \ldots, S_k(x,y)\}$. Statement 2 from the Theorem tells us that for those r we have $\{r(x,y)\} \sqsubseteq_{\mathcal{T}} q$. In the same way as in the proof of the claim we obtain that there are no disjuncts q_1 and q_2 in $q_{\mathcal{T}}$ with the properties described in the statement of the claim. Hence, we can continue with computing $p_D(q_{\mathcal{T}})$ in the same way as in Case (i).

Case (iii). There is a role name r such that both r and r^- is $\mathcal{T}$-generated in q and q is of the form $\{S_1(x,y), \ldots, S_k(x,y)\}$. It is not hard to prove that $q' \sqsubseteq_{\mathcal{T}} q$ implies actually $q' \sqsubseteq q$. Hence, $q_{\mathcal{T}} \equiv q$ and it suffices to compute $p_D(q)$. Again, x_r is root variable but no separator, since q contains atoms $r(x,y)$ and $r(y,x)$ (recall that r and r^- are $\mathcal{T}$-generated). Thus, q is not ranked and we need to consider the ranking $\overline{q}$ of q in order to compute $p(q)$. Following (the proof of) Proposition 1, $\overline{q}$ is obtained as

$$\begin{array}{l} \quad S_1^=(x), \ldots, S_k^=(x) \\ \vee\; S_1^<(x,y), \ldots, S_k^<(x,y), x < y \\ \vee\; S_1^>(x,y), \ldots, S_k^>(x,y), x > y \end{array}$$

where $S_i^= = s_i^=$ (independent from S_i being s_i or s_i^-), $S_i^<$ is $s_i^<$ ($s_i^>$, respectively) when $S_i = s_i$ ($S_i = s_i^-$, respectively), and analogously for $S_i^>$. It is easy to see that $\overline{q}$ has the separator x. Observe that the ranking also removes reflexive atoms, hence we can continue with computing $p(\overline{q})$ using the algorithm in Case (i).

❏

B Proofs for Section 5

Let $\mathcal{T}$ be an $\mathcal{ELI}$-TBox and q a Boolean connected CQ that involves at least one individual name. We first show that we can assume w.l.o.g. that $\mathcal{T}$ contains only CIs of the forms

$$\begin{array}{cc} A \sqsubseteq B & A \sqsubseteq \exists R.B \\ B_1 \sqcap B_2 \sqsubseteq A & \exists R.B \sqsubseteq A \end{array}$$

where R ranges over role names and their inverse and A, B, B_1, B_2 over concept names and $\top$. Let $\mathsf{sub}(\mathcal{T})$ denote the set of all subconcepts of (concepts that occur in) $\mathcal{T}$ and reserve a concept name X_C for every $C \in \mathsf{sub}(\mathcal{T}) \setminus (\mathsf{N_C} \cup \{\top\})$ such that X_C occurs neither in $\mathcal{T}$ nor in q. Set

$$\sigma(C) = \begin{cases} C & \text{if } C \in \mathsf{N_C} \cup \{\top\} \\ X_{D_1} \sqcap X_{D_2} & \text{if } C = D_1 \sqcap D_2 \\ \exists r.X_D & \text{if } C = \exists r.D \end{cases}$$

Then put

$$\mathcal{T}' = \bigcup_{C \sqsubseteq D \in \mathcal{T}} X_C \sqsubseteq X_D \quad \cup \quad \bigcup_{C \in \mathsf{sub}(\mathcal{T}) \setminus (\mathsf{N_C} \cup \{\top\})} X_C \equiv \sigma(C)$$

where $C \equiv D$ abbreviates $C \sqsubseteq D, D \sqsubseteq C$. After further replacing each CI of the form $A \sqsubseteq B_1 \sqcap B_2$ with $A \sqsubseteq B_1$ and $A \sqsubseteq B_2$, $\mathcal{T}'$ is of the required syntactic form. Clearly, the conversion can be done in polynomial time.

We want to replace $\mathcal{T}$ with the TBox $\mathcal{T}'$ in normal form. To implement this, we consider ABoxes in a restricted signature. A *predicate* is a set of concept names and role names and a *signature* is a set of predicates. We use $\mathsf{sig}(\mathcal{T})$ to denote the set of predicates that occur in $\mathcal{T}$ and likewise for $\mathsf{sig}(q)$. A *Σ-ABox* is an ABox that contains only symbols from Σ. Due to the following result, we shall indeed be able to replace $\mathcal{T}$ with $\mathcal{T}'$ when we are careful about ABox signatures.

Theorem 11. *Let $\Sigma = \mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$.*

1. *q is FO-rewritable relative to $\mathcal{T}$ (over all ABoxes) iff q is FO-rewritable relative to $\mathcal{T}'$ over Σ-ABoxes;*
2. *q is #P-hard relative to $\mathcal{T}$ (over all ipABoxes) iff q is #P-hard relative to $\mathcal{T}'$ over Σ-ipABoxes.*

Proof. For Point 1, first assume that q is FO-rewritable relative to $\mathcal{T}$ and let $q_\mathcal{T}$ be an FO-rewriting. We show that $q_\mathcal{T}$ is also an FO-rewriting of q relative to $\mathcal{T}'$ over Σ-ABoxes. To this end, let $\mathcal{A}$ be a Σ-ABox. Since the fresh concept names X_C occur neither in q nor in $\mathcal{A}$, it is easy to show that $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{A}, \mathcal{T}' \models q$. In summary, $\mathcal{A}, \mathcal{T}' \models q$ iff $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{I}_\mathcal{A} \models q_\mathcal{T}$, thus we are done. Conversely, assume that q is FO-rewritable relative to $\mathcal{T}'$ over Σ-ABoxes and let $q_{\mathcal{T}'}$ be an FO-rewriting. Since each non-Σ-symbol is interpreted as the empty set in $\mathcal{I}_\mathcal{A}$ for any Σ-ABox $\mathcal{A}$, we can w.l.o.g. assume that no such symbol occurs in $q_{\mathcal{T}'}$ (if it does, replace it with false). We show that $q_{\mathcal{T}'}$ is also an FO-rewriting of q relative to $\mathcal{T}$. Let $\mathcal{A}$ be an ABox and $\mathcal{A}|_\Sigma$ the result of dropping all non-Σ-assertions from $\mathcal{A}$. We have $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{A}|_\Sigma, \mathcal{T} \models q$ (since q and $\mathcal{T}$ contain only Σ-symbols) iff $\mathcal{A}|_\Sigma, \mathcal{T}' \models q$ (since the X_C occur neither in q nor in $\mathcal{A}|_\Sigma$) iff $\mathcal{I}_{\mathcal{A}_\Sigma} \models q_{\mathcal{T}'}$ (since $q_{\mathcal{T}'}$ is an FO-rewriting) iff $\mathcal{I}_\mathcal{A} \models q_{\mathcal{T}'}$ (since there are no non-Σ-symbols in $q_{\mathcal{T}'}$).

For Point 2, first assume that q is #P-hard relative to $\mathcal{T}$. Since $\mathcal{T}$ and q contain only Σ-symbols, we have $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{A}|_\Sigma, \mathcal{T} \models q$ for all ABoxes $\mathcal{A}$. It follows that q is #P-hard relative to $\mathcal{T}$ over Σ-ABoxes. Since we have $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{A}, \mathcal{T}' \models q$ for all Σ-ABoxes $\mathcal{A}$ (see proof of Point 1 above) and thus also $p(\mathcal{A}, \mathcal{T} \models q) = p(\mathcal{A}, \mathcal{T}' \models q)$, answering q relative to $\mathcal{T}$ over Σ-ABoxes is simply the same problem as answering q relative to $\mathcal{T}'$ over Σ-ABoxes and we are done. For the converse direction, $p(\mathcal{A}, \mathcal{T} \models q) = p(\mathcal{A}, \mathcal{T}' \models q)$ for all Σ-ABoxes $\mathcal{A}$ means that answering q relative to $\mathcal{T}'$ over Σ-ABoxes is simply a subproblem of answering q relative to $\mathcal{T}$, thus #P-hardness of the former implies #P-hardness of the latter. ❑

The proof of Theorem 6 is based on a connection between FO-rewritability and boundedness of an appropriate fixpoint operator, similar to what was observed in [26]. Given an ABox $\mathcal{A}$ and an $a \in \mathsf{Ind}(\mathcal{A})$, we denote with $\mathcal{A}|_a$ the *neighbourhood* of a, i.e., the restriction of $\mathcal{A}$ to the individual name a and all members of $\{b \mid R(a,b) \in \mathcal{A}\} \cup \{b \mid R(b,a) \in \mathcal{A}\}$ (where R is a role name or its inverse). For a TBox $\mathcal{T}$, define

$$f_\mathcal{T}(\mathcal{A}) = \mathcal{A} \cup \{A(a) \mid a \in \mathsf{Ind}(\mathcal{A}) \wedge \mathcal{A}|_a, \mathcal{T} \models A(a)\}$$

and set $f^\infty_\mathcal{T}(\mathcal{A}) := \bigcup_{i \geq 0} f^i_\mathcal{T}(\mathcal{A})$, where $f^i_\mathcal{T}(\cdot)$ denotes application of $f_\mathcal{T}$, iterated i times. Note that the application of $f_\mathcal{T}(\cdot)$ yields ABoxes, though not necessarily Σ-ABoxes, $\Sigma = \mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$. It is not hard to prove that for all $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$, we have $\mathcal{A}, \mathcal{T} \models A(a)$ iff $A(a) \in f^\infty_\mathcal{T}(\mathcal{A})$ [26]. We want to establish an analogous claim for CQs, which requires first introducing several technical notions. In particular, we construct a query $q_\mathcal{T}$ from q and $\mathcal{T}$ such that answering q over a Σ-ABox $\mathcal{A}$ is equivalent to answering $q_\mathcal{T}$ over $f^\infty_\mathcal{T}(\mathcal{A})$. Note that $q_\mathcal{T}$ is in general *not* an FO-rewriting of q relative to $\mathcal{T}$, which are not guaranteed to exist in $\mathcal{ELI}$; intuitively, $q_\mathcal{T}$ constitutes the part of the FO-rewriting that deals with objects generated by existential quantifiers.

We use $\mathsf{EQ}(q)$ to denote the set of all equivalence relations on $\mathsf{term}(q)$ such that $a \not\sim b$ for all distinct $a, b \in \mathsf{N_I}$. Every $\sim \in \mathsf{EQ}(q)$ defines the following *collapsing of* q:

$$q_\sim = \{r(s_{[t]}, s_{[t']}) \mid r(t,t') \in q\} \cup \{A(s_{[t]}) \mid A(t) \in q\}.$$

where $s_{[t]} = a$ if the individual name a is in $[t]$ and $s_{[t]}$ is the fresh variable $z_{[t]}$ otherwise.

We call a CQ *tree-shaped* if the undirected graph (V_q, E_q) with $V_q = \mathsf{term}(q)$ and $E_q = \{\{t, t'\} \mid r(t,t') \in q\}$ is a tree in which an individual name occurs, if at all, only at the root. A *splitting* S of a CQ q is a partition $q_0, \ldots, q_k$ of q (with q_0 possibly empty and $q_1, \ldots, q_k$ non-empty) such that for $1 \leq i \leq k$, we have

1. $q_1, \ldots, q_k$ are tree-shaped queries with roots $t_1, \ldots, t_k$

2. $\mathsf{term}(q_i) \cap \mathsf{term}(q_j) = \emptyset$ for $i < j \leq k$
3. $\mathsf{term}(q_0) \cap \mathsf{term}(q_i) = \{t_i\}$.

Let $\mathsf{split}(q)$ denote the set of all splittings of q and $\mathsf{CN}(\mathcal{T})$ the concept names that occur in $\mathcal{T}$. For a subset $\rho \subseteq \mathsf{CN}(\mathcal{T})$, we use $\mathcal{A}_\rho$ to denote the ABox $\{A(\widehat{a}) \mid A \in \rho\}$ where $\widehat{a}$ is a fixed, 'special' individual name. Given a splitting $S = q_0, \ldots, q_k$ of a CQ q, a map $\rho : \{1, \ldots, k\} \rightarrow 2^{\mathsf{CN}(\mathcal{T})}$ is a *justification for S relative to $\mathcal{T}$* if for all $i \in \{1, \ldots, k\}$, we have $\mathcal{A}_{\rho(i)}, \mathcal{T} \models^{\widehat{a}} q_i$ meaning that in every model $\mathcal{I}$ of $\mathcal{A}_{\rho(i)}$ and $\mathcal{T}$, there is a match of the tree-shaped Boolean query q_i that maps its root t_i to $\widehat{a}^{\mathcal{I}}$. We use $q_{\rho(i)}$ to denote the query $\bigwedge_{A \in \rho(i)} A(t_i)$ where, again t_i denotes the root of q_i. Let $\mathsf{just}(S, \mathcal{T})$ denote all possible justifications of S relative to $\mathcal{T}$. For each $\sim \in \mathsf{EQ}(q)$, set

$$\widehat{q_\sim} = \exists z_{[\hat{t}_1]} \cdots \exists z_{[\hat{t}_n]} \bigvee_{S = q_0, \ldots, q_k \in \mathsf{split}(q_\sim)} \; \bigvee_{\rho \in \mathsf{just}(S, \mathcal{T})} \big(q_0 \wedge \bigwedge_{i=1}^{k} q_{\rho(i)}\big)$$

where $[\hat{t}_1], \ldots, [\hat{t}_n]$ are the equivalence classes of $\sim$ that do not contain an individual name. Finally, we define the CQ $q_{\mathcal{T}}$ as $\bigvee_{\sim \in \mathsf{EQ}(q)} \widehat{q_\sim}$. Clearly, $q_{\mathcal{T}}$ is a UCQ.

Theorem 12. *For any CQ q, $\mathcal{ELI}$-TBox $\mathcal{T}$ in normal form, and Σ-ABox $\mathcal{A}$, where $\Sigma = \mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$, we have $\mathcal{A}, \mathcal{T} \models q$ iff $\mathcal{I}_{f^\infty_{\mathcal{T}}(\mathcal{A})} \models q_{\mathcal{T}}$.*

To prove Theorem 12, we introduce canonical models for ABoxes and $\mathcal{ELI}$-TBoxes. Let $\mathcal{A}$ be an ABox and $\mathcal{T}$ an $\mathcal{ELI}$-TBox in normal form. A *type for $\mathcal{T}$* is a set $T \subseteq \mathsf{CN}(\mathcal{T})$. When $a \in \mathsf{Ind}(\mathcal{A})$, T, T' are types for $\mathcal{T}$, and R is a role, we write

- $a \leadsto_R T$ if $\mathcal{A}, \mathcal{T} \models \exists R. \bigsqcap T(a)$ and for each type S for $\mathcal{T}$ with $T \subsetneq S$, we have $\mathcal{A}, \mathcal{T} \not\models \exists R. \bigsqcap T(a)$;
- $T \leadsto_R T'$ if $\mathcal{T} \models \bigsqcap T \sqsubseteq \exists R. \bigsqcap T'$ and for each type S for $\mathcal{T}$ with $T' \subsetneq S$, we have $\mathcal{T} \not\models \bigsqcap T \sqsubseteq \exists R. \bigsqcap S$.

A *path for $\mathcal{A}$ and $\mathcal{T}$* is a sequence $p = aR_1T_1 \cdots T_{n-1}R_nT_n$, $n \geq 0$, with $a \in \mathsf{Ind}(\mathcal{A})$, $R_1, \ldots, R_n$ roles, and $T_1, \ldots, T_n$ types for $\mathcal{T}$ such that $a \leadsto_{R_1} T_1$ and $T_i \leadsto_{R_i} T_{i+1}$ for $1 \leq i < n$. When $n > 0$, we use $\mathsf{tail}(p)$ to denote T_n. Define the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \text{the set of all paths for } \mathcal{A} \text{ and } \mathcal{T} \\
A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{A}, \mathcal{T} \models A(a)\} \cup \\
&\quad\; \{p \in \Delta^{\mathcal{I}} \setminus \mathsf{Ind}(\mathcal{A}) \mid \mathcal{T} \models \textstyle\bigsqcap \mathsf{tail}(p) \sqsubseteq A\} \\
r^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid r(a,b) \in \mathcal{A}\} \cup \\
&\quad\; \{(p, prT) \mid prT \in \mathsf{Paths}\} \cup \\
&\quad\; \{(pr^-T, p) \mid pr^-T \in \mathsf{Paths}\} \\
a^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} &= a
\end{aligned}$$

It is standard to prove that $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ is canonical in the following sense.

Lemma 7. *For any CQ q, $\mathcal{ELI}$-TBox $\mathcal{T}$ in normal form, and ABox $\mathcal{A}$, we have $\mathcal{A},\mathcal{T} \models q$ iff $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q$.*

We are now ready to prove Theorem 12.

Proof.(sketch of Theorem 12) "if". Assume that $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$ and let π be a match of $q_\mathcal{T}$ in $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})}$. Let q' be the disjunct of $q_\mathcal{T}$ with $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q'$. By definition of $q_\mathcal{T}$, there is an $S = q_0,\dots,q_k \in \mathsf{split}(q_\sim)$ and a $\rho \in \mathsf{just}(S,\mathcal{T})$ such that $q' = q_0 \wedge \bigwedge_{i=1}^k q_{\rho(i)}$. For $1 \leq i \leq k$, we have $\mathcal{A}_{\rho(i)},\mathcal{T} \models^{\widehat{a}} q_i$ and, by Lemma 7, there is thus a match π_i of q_i in $\mathcal{I}_{\mathcal{A}_{\rho(i)},\mathcal{T}}$ such that the root t_i of q_i is mapped to $\widehat{a}$. Since π is a match of $q_{\rho(i)}$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, it can be shown that there is a homomorphism from $\mathcal{I}_{\mathcal{A}_{\rho(i)},\mathcal{T}}$ to $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ with $h(\widehat{a}) = \pi(t_i)$. Thus, we find a match π'_i of q_i in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ with $\pi'_i(t_i) = \pi(t_i)$. It can be verified that $\pi \cup \pi_1 \cup \dots \cup \pi_k$ is a match of q in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. By Lemma 7, we have $\mathcal{A},\mathcal{T} \models q$ as required.

"only if". Assume that $\mathcal{A},\mathcal{T} \models q$. By Lemma 7, this means $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q$. Let π be a match of q in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ and let $a_1,\dots,a_k$ be the elements of $\mathsf{Ind}(\mathcal{A})$ that are in the range of π. Define a splitting $S = q_0,\dots,q_k$ of q, where q_0 is the restriction of q to the terms t with $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ and each q_i is the restriction of q to the terms t with $\pi(t) = a_i p$ for some non-empty p. Define a justification ρ for S relative to $\mathcal{T}$ by setting $\rho(i) = \{A \in \mathsf{CN}(\mathcal{T}) \mid a_i \in A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}\}$ for $1 \leq i \leq k$. Then $q' = q_0 \wedge \bigwedge_{i=1}^k q_{\rho(i)}$ is a disjunct in $q_\mathcal{T}$ and it can be verified that $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q'$. ❏

We say that a CQ q is *k-bounded relative to a TBox $\mathcal{T}$ over Σ-ABoxes* if for every Σ-ABox $\mathcal{A}$, we have that $\mathcal{I}_{f^k_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$ iff $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$. We say that q is *bounded relative to a TBox $\mathcal{T}$ over Σ-ABoxes* if it is k-bounded for some k.

Theorem 13. *If a CQ q is bounded relative to $\mathcal{T}$ over Σ-ABoxes, $\Sigma = \mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(q)$, then it is FO-rewritable relative to $\mathcal{T}$ over Σ-ABoxes.*

Proof. Assume that q is bounded relative to $\mathcal{T}$ over Σ-ABoxes and let $k > 0$ be such that $\mathcal{I}_{f^k_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$ iff $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$ for every Σ-ABox $\mathcal{A}$.

Observe that we work with finite Σ, and thus there are only finitely many neighborhoods $\mathcal{A}_a$ that can occur in a Σ-ABox $\mathcal{A}$, up to isomorphism. Every such neighborhood $\mathcal{N}$ with individual name a in the center can be converted in a straightforward way into an existential, conjunctive, positive FO-formula:

$$\varphi_\mathcal{N} = \bigwedge_{A(a)\in\mathcal{N}} A(x) \wedge \bigwedge_{b\in\mathsf{Ind}(\mathcal{N})} \exists y : \Big(\bigwedge_{R(a,b)\in\mathcal{N}} R(x,y) \wedge \bigwedge_{B(b)\in\mathcal{N}} B(y)\Big)$$

where $R(x,y)$ denotes $r(y,x)$ when $R = r^-$. For each concept name A, we use Γ_A to denote the set of neighborhoods $\mathcal{N}$ with center a such that $\mathcal{N} \models A(a)$. For every concept name A and $i \geq 0$, set

- $q^0_A(x) := A(x)$
- $q^{i+1}_A(x) := q^i_A \vee \bigvee_{\mathcal{N}\in\Gamma_A} \varphi'_\mathcal{N}$ where $\varphi'_\mathcal{N}$ is obtained from $\varphi_\mathcal{N}$ by replacing, for each concept name B, every atom $B(z)$ with $q^i_B[z/x]$.

The following can be proved by induction on i:

Claim. For every Σ-ABox $\mathcal{A}$ and $i \geq 0$, we have $\mathcal{I}_\mathcal{A} \models q^i_A[a]$ iff $A(a) \in f^i_\mathcal{T}(\mathcal{A})$.

Let $\widehat{q}_\mathcal{T}$ be $q_\mathcal{T}$ with every atom $A(t)$ replaced with $q^k_A[t/x]$. We show that $\widehat{q}_\mathcal{T}$ is an FO-rewriting of q relative to $\mathcal{T}$, which finishes the proof. By the above claim and due to the fact that $f^k_\mathcal{T}(\mathcal{A})$ differs from $\mathcal{A}$ only by additional concept assertions, we have $\mathcal{I}_\mathcal{A} \models \widehat{q}_\mathcal{T}$ iff $\mathcal{I}_{f^k_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$. This is the case iff $\mathcal{I}_{f^\infty_\mathcal{T}(\mathcal{A})} \models q_\mathcal{T}$ (by choice of k) iff $\mathcal{A}, \mathcal{T} \models q$ (by Theorem 12). ❏

The next and central step is to prove the following result.

Theorem 14. *If a CQ q is unbounded relative to an $\mathcal{ELI}$-TBox $\mathcal{T}$ over Σ-ABoxes, then q is* #P*-hard relative to $\mathcal{T}$ over Σ-ABoxes.*

Let q be unbounded relative to $\mathcal{T}$ over Σ-ABoxes and set

$$m = |\mathcal{T}| \cdot |q_\mathcal{T}| \cdot (|\mathcal{T}| + |q_\mathcal{T}|)^{|q_\mathcal{T}|+2} + 1.$$

Since q is not bounded relative to $\mathcal{T}$, there is a Σ-ABox $\mathcal{A}_0$ such that $\mathcal{I}_{f^m_\mathcal{T}(\mathcal{A}_0)} \models q_\mathcal{T}$, but $\mathcal{I}_{f^{m-1}_\mathcal{T}(\mathcal{A}_0)} \not\models q_\mathcal{T}$. Choose a concrete match π of $q_\mathcal{T}$ in $\mathcal{I}_{f^m_\mathcal{T}(\mathcal{A}_0)}$. In the following, when writing $R(a,b) \in \mathcal{A}$ with $R = r^-$, we mean $r(b,a) \in \mathcal{A}$. Construct a Σ-ABox $\mathcal{A}_1$ as follows:

- a *π-path in $\mathcal{A}_0$ of length n* is a sequence $p = a_0 R_1 a_1 \cdots R_n a_n$ such that $a_0 \in \mathsf{ran}(\pi)$ and $R_i(a_{i-1}, a_i) \in \mathcal{A}_0$ for $1 \leq i \leq n$; we use $\mathsf{tail}(p)$ to denote a_n;
- $\mathsf{Ind}(\mathcal{A}_1)$ is the set of all π-paths in $\mathcal{A}_0$ of length at most $m + |q_\mathcal{T}|$;
- if $A(a) \in \mathcal{A}_0$, $p \in \mathsf{Ind}(\mathcal{A}_1)$, and $\mathsf{tail}(p) = a$, then $A(p) \in \mathcal{A}$;
- if $pRa \in \mathsf{Ind}(\mathcal{A}_1)$, then $R(p, pRa) \in \mathcal{A}_1$;
- if $r(a,b) \in \mathcal{A}_0$ and $a, b \in \mathsf{ran}(\pi)$, then $r(a,b) \in \mathcal{A}_1$.
- these are all assertions in $\mathcal{A}_1$.

Note that $\mathcal{A}_1$ has, in a loose sense, a forest shape: the roots, which are the elements of $\mathsf{Ind}(\mathcal{A}_1) \cap \mathsf{Ind}(\mathcal{A}_0)$, form a 'core' whose relational structure is not restricted in any way; moreover, each root a gives rise to a tree-shaped sub-ABox of $\mathcal{A}_1$, namely the restriction to the individuals p that have a as a prefix.

Lemma 8. *$\mathcal{I}_{f^m_\mathcal{T}(\mathcal{A}_1)} \models q_\mathcal{T}$, but $\mathcal{I}_{f^{m-1}_\mathcal{T}(\mathcal{A}_1)} \not\models q_\mathcal{T}$.*

Proof. We first prove the following.

Claim. For all $i \leq m$, $p \in \mathsf{Ind}(\mathcal{A}_1)$ of length at most $m + |q_\mathcal{T}| - i$ with $\mathsf{tail}(p) = a$, and $A \in \mathsf{N_C}$, we have $A(p) \in f^i_\mathcal{T}(\mathcal{A}_1)$ iff $A(a) \in f^i_\mathcal{T}(\mathcal{A}_0)$.

The proof is by induction on i. The induction start is trivial by construction of $\mathcal{A}_1$. For the induction step, we have $A(p) \in f^i_\mathcal{T}(\mathcal{A}_1)$ iff $f^{i-1}_\mathcal{T}(\mathcal{A}_1)|_a, \mathcal{T} \models A(p)$ (by definition of the function $f_\mathcal{T}$) iff $f^{i-1}_\mathcal{T}(\mathcal{A}_0)|_a, \mathcal{T} \models A(a)$ (by induction hypothesis) iff $A(a) \in f^i_\mathcal{T}(\mathcal{A}_0)$.

To see that $\mathcal{I}_{f^m_\mathcal{T}(\mathcal{A}_1)} \models q_\mathcal{T}$, consider the match π used in the construction of $\mathcal{A}_1$. By the claim, for all $a \in \mathsf{ran}(\pi)$ and $A \in \mathsf{N_C}$, we have $A(a) \in f^m_\mathcal{T}(\mathcal{A}_0)$ iff $A(a) \in f^m_\mathcal{T}(\mathcal{A}_1)$. It follows that π is also a match of $q_\mathcal{T}$ in $\mathcal{I}_{f^m_\mathcal{T}(\mathcal{A}_1)}$.

For $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)} \not\models q_{\mathcal{T}}$, assume that there is a match τ of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)}$ and select a concrete disjunct q' of $q_{\mathcal{T}}$ that is matched by τ. Since q is connected and contains at least one individual name, by construction the same holds for q'. It follows that all paths $p \in \mathsf{ran}(\tau)$ are of length at most $|q_{\mathcal{T}}|$. By the claim, it follows that for all $p \in \mathsf{ran}(\tau)$ with $\mathsf{tail}(p) = a$ and $A \in \mathsf{N_C}$, we have $A(a) \in f_{\mathcal{T}}^{m-1}(\mathcal{A}_0)$ iff $A(p) \in f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)$. This, in turn, implies that the function τ' defined by setting $\tau'(t) := \mathsf{tail}(\tau(t))$ for all $t \in \mathsf{term}(q')$ is a match of q' in $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_0)}$, in contradiction to $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_0)} \not\models q_{\mathcal{T}}$. ❏

We further modify $\mathcal{A}_1$ by exhaustively applying the following operation: if $p \in \mathsf{Ind}(\mathcal{A}_1)$ is a non-root (i.e., p is a path with length > 1) and $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A}_p^-)} \models q_{\mathcal{T}}$,[2] where $\mathcal{A}_p^-$ is obtained from $\mathcal{A}_1$ by removing the subtree rooted at p, then replace $\mathcal{A}_1$ with $\mathcal{A}_p^-$. Let $\mathcal{A}$ be the Σ-ABox finally obtained. For $i \geq 0$, we say that $\mathcal{A}$ *has outdegree at most* i if for every $p \in \mathsf{Ind}(\mathcal{A}_1)$, the cardinality of the set

$$\{p' \in \mathsf{Ind}(\mathcal{A}_1) \setminus \mathsf{Ind}(\mathcal{A}_0) \mid \exists R : R(p, p') \in \mathcal{A}_1\}$$

is bounded by i. In words, every individual in $\mathcal{A}_1$ has at most i successors that are non-root nodes.

Lemma 9.

1. $\mathcal{I}_{f_{\mathcal{T}}^{m'}(\mathcal{A})} \models q_{\mathcal{T}}$*, but* $\mathcal{I}_{f_{\mathcal{T}}^{m'-1}(\mathcal{A})} \not\models q_{\mathcal{T}}$ *for some* $m' \geq m$*;*
2. $\mathcal{A}$ *has outdegree at most* $|\mathcal{T}| + |q_{\mathcal{T}}|$*.*

Proof. For Point 1, it suffices to show that $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A})} \not\models q_{\mathcal{T}}$ and $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})} \models q_{\mathcal{T}}$. The former is a consequence of the facts that $\mathcal{I}_{f_{\mathcal{T}}^{m-1}(\mathcal{A}_1)} \not\models q_{\mathcal{T}}$, $\mathcal{A} \subseteq \mathcal{A}_1$, and $q_{\mathcal{T}}$ is a UCQ; the latter is is immediate by construction of $\mathcal{A}$.

For Point 2, assume to the contrary of what is to be shown that there is a $p \in \mathsf{Ind}(\mathcal{A})$ such that the cardinality of

$$\Gamma = \{p' \in \mathsf{Ind}(\mathcal{A}) \setminus \mathsf{Ind}(\mathcal{A}_0) \mid \exists R : R(p, p') \in \mathcal{A}\}$$

exceeds $|\mathcal{T}| + |q_{\mathcal{T}}|$. Fix a match π of $q_{\mathcal{T}}$ in $\mathcal{I}_{f_{\mathcal{T}}^{\infty}(\mathcal{A})}$ and select all individual names in Γ that are in the range of π. For each $\exists r.A \sqsubseteq B \in \mathcal{T}$ such that $B(p) \in f_{\mathcal{T}}^{\infty}(\mathcal{A})$, select an individual name $p' \in \Gamma$ such that

1. $r(p, p') \in \mathcal{A}$ and $A(p') \in f_{\mathcal{T}}^{j}(\mathcal{A})$ for some $j < m$;
2. there is no p' that satisfies Point 1 for some smaller j.

Note that such a p' need not exist, in which case no node is selected for $\exists r.A \sqsubseteq B$. Since $|\Gamma| \geq |\mathcal{T}| + |q_{\mathcal{T}}|$, there is at least one element $p_0 \in \Gamma$ that is not selected. Consider the ABox $\mathcal{A}_{p_0}^-$ obtained from $\mathcal{A}$ by dropping the subtree rooted at p_0. Exploiting that $\mathcal{T}$ is in normal form and using the definition of the function $f_{\mathcal{T}}(\cdot)$, it is not difficult to show that no deductions are lost at the individuals

[2] For the remainder of the proof, is it essential to consider all matches here instead of only the match π used to define $\mathcal{A}_0$.

that remain in $\mathcal{A}^-_{p_0}$, i.e.,

Claim. For all $i \geq 0$, $p \in \mathcal{A}^-_{p_0}$, and concept names A, we have $A(p) \in f^i_{\mathcal{T}}(\mathcal{A}^-_{p_0})$ iff $A(p) \in f^i_{\mathcal{T}}(\mathcal{A})$.

Consequently, π is still a match of $q_{\mathcal{T}}$ in $\mathcal{I}_{f^m_{\mathcal{T}}(\mathcal{A}^-_{p_0})}$. This is a contradiction to the fact that the subtree rooted at p_0 has not been dropped during the construction of $\mathcal{A}$. ❑

Since iterated applications of $f_{\mathcal{T}}$ can assign, in the worst case, every concept name from $\mathcal{T}$ to every individual name, the following is immediate.

Lemma 10. *On every ABox $\mathcal{B}$ with $|\mathsf{Ind}(\mathcal{B})| = i$, we have $f^\ell_{\mathcal{T}}(\mathcal{B}) = f^{\ell+1}_{\mathcal{T}}(\mathcal{B})$ where $\ell = |\mathcal{T}| \cdot i$.*

We say that a path $p' \in \mathsf{Ind}(\mathcal{A})$ is an *extension* of a path $p \in \mathsf{Ind}(\mathcal{A})$ if $p' = pRa$ for some R and a. Set $d = |q_{\mathcal{T}}| + 3$. We claim that the ABox $\mathcal{A}$ must contain a sequence

$$R_d(p_d, p_{d-1}), \ldots, R_1(p_1, p_0)$$

such that p_d is of length one and p_i is an extension of p_{i+1} for all $i < d$. Assume this is not the case. Then all paths in $\mathsf{Ind}(\mathcal{A})$ are of length at most $d-1$. Since $\mathcal{A}$ is forest-shaped with at most $|q_{\mathcal{T}}|$ roots, Point 2 of Lemma 9 yields $|\mathsf{Ind}(\mathcal{A})| \leq |q_{\mathcal{T}}| \cdot (|\mathcal{T}| + |q_{\mathcal{T}}|)^{d-1}$ and by Lemma 10 we have $f^\ell_{\mathcal{T}}(\mathcal{B}) = f^{\ell+1}_{\mathcal{T}}(\mathcal{B})$ where $\ell = |\mathcal{T}| \cdot |q_{\mathcal{T}}| \cdot (|\mathcal{T}| + |q_{\mathcal{T}}|)^{d-1}$. This is a contradiction to Point 1 of Lemma 9 and the fact that $m' > m > \ell$.

Lemma 11. *If at least one of $R_3(p_3, p_2), R_2(p_2, p_1), R_1(p_1, p_0)$ is dropped from $\mathcal{A}$ resulting in an ABox $\mathcal{A}'$, then $\mathcal{A}', \mathcal{T} \not\models q$.*

Proof. Let $R_i(p_i, p_{i-1})$ be the assertion that was dropped from $\mathcal{A}$ to obtain $\mathcal{A}'$ and assume to the contrary of what is to be shown that $\mathcal{A}', \mathcal{T} \models q$. Thus $\mathcal{I}_{f^\infty_{\mathcal{T}}(\mathcal{A}')} \models q_{\mathcal{T}}$ by Theorem 12 and we can choose a concrete match π of $q_{\mathcal{T}}$ in $\mathcal{I}_{f^\infty_{\mathcal{T}}(\mathcal{A}')}$. Since the length of p_{i-1} exceeds $|q_{\mathcal{T}}|$ and $q_{\mathcal{T}}$ is rooted (contains at least one individual name and is connected), none of $p_{i-1}, \ldots, p_0$ can appear in the range of π and thus π is also a map from $\mathsf{term}(q_{\mathcal{T}})$ to $\mathsf{Ind}(\mathcal{A}^-_{p_{i-1}})$. Trivially, we have $A(p) \in f^\infty_{\mathcal{T}}(\mathcal{A}')$ iff $A(p) \in f^\infty_{\mathcal{T}}(\mathcal{A}^-_{p_{i-1}})$ for all $p \in \mathsf{Ind}(\mathcal{A}^-_{p_{i-1}})$ and concept names A. It follows that π is actually a match of $q_{\mathcal{T}}$ in $\mathcal{I}_{\mathcal{A}^-_{p_{i-1}}}$. This is a contradiction to the fact that the subtree rooted at p_{i-1} was not removed during the construction of $\mathcal{A}$. ❑

We now prove Theorem 14 by a reduction of the problem of counting the number of satisfying assignments for a monotone bipartite DNF formula, which is known to be #sc P-hard. More specifically, input formulas are of the form

$$\psi = (x_{i_1} \wedge y_{j_1}) \vee \cdots \vee (x_{i_k} \wedge y_{j_k})$$

where the set X of variables that occur on the left-hand side of a conjunction in ψ is disjoint from the set Y of variables that occur on the right-hand side of a conjunction in ψ.

For the reduction, let ψ be a formula as above, $X = \{x_1, \dots, x_{n_x}\}$, and $Y = \{y_1, \dots, y_{n_y}\}$. Define an ipABox $(\mathcal{A}_\psi, p_\psi)$ by starting with the ABox $\mathcal{A}$ constructed above and 'multiplying' the assertions $R_3(a_3, a_2)$, $R_2(a_2, a_1)$, $R_1(a_1, a_0)$ from Lemma 11 using fresh individual names $b_1, \dots, b_{n_x}$ and $c_1, \dots, c_{n_y}$. In detail:

- start with the ABox $\mathcal{A}$ constructed above without the assertions $R_3(a_3, a_2)$, $R_2(a_2, a_1)$, $R_1(a_1, a_0)$ from Lemma 11, assign probability 1 to all assertions;
- add the following assertions with probability 1:

$$\begin{array}{rl}
A(b_i) & \text{for all } A(a_2) \in \mathcal{A} \text{ and } 1 \leq i \leq n_x \\
R(b_i, d) & \text{for all } R(a_2, d) \in \mathcal{A} \text{ and } 1 \leq i \leq n_x \\
A(c_i) & \text{for all } A(a_1) \in \mathcal{A} \text{ and } 1 \leq i \leq n_y \\
R(c_i, d) & \text{for all } R(a_1, d) \in \mathcal{A} \text{ and } 1 \leq i \leq n_y \\
R_2(b_{i_\ell}, c_{j_\ell}) & \text{for } 1 \leq \ell \leq k;
\end{array}$$

- add the following assertions with probability 0.5,

$$\begin{array}{rl}
R_3(a_3, b_i) & \text{for } 1 \leq i \leq n_x \\
R_1(c_i, a_0) & \text{for } 1 \leq i \leq n_y.
\end{array}$$

We are interested in ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\psi$ with $p_\psi(\mathcal{A}') > 0$. Each such ABox has probability $\frac{1}{2^{|X|+|Y|}}$ and corresponds to a truth assignment $\delta_{\mathcal{A}'}$ to the variables in $X \cup Y$: for $x_i \in X$, $\delta_{\mathcal{A}'}(x_i) = 1$ iff $R_3(a_3, b_i) \in \mathcal{A}'$ and for $y_i \in Y$, $\delta_{\mathcal{A}'}(y_i) = 1$ iff $R_1(c_i, a_0) \in \mathcal{A}'$. Let $\#\psi$ the number of truth assignments to the variables $X \cup Y$ that satisfy ψ. To complete the reduction, we show that $p(\mathcal{A}_\psi, \mathcal{T} \models q) = \frac{\#\psi}{2^{|X|+|Y|}}$. By what was said above, this is an immediate consequence of the following observation.

Lemma 12. *For all ABoxes $\mathcal{A}' \subseteq \mathcal{A}_\psi$ with $p_\psi(\mathcal{A}') > 0$, $\delta_{\mathcal{A}'} \models \psi$ iff $\mathcal{A}', \mathcal{T} \models q$.*

Proof. "if". Let $\delta_{\mathcal{A}'} \not\models \psi$ and assume to the contrary of what is to be shown that $\mathcal{A}', \mathcal{T} \models q$. Since $\delta_{\mathcal{A}'} \not\models \psi$ and by construction of $\mathcal{A}_\psi$, there are no i, j such that $R_3(a_3, b_i), R_2(b_i, c_j), R_1(c_j, a_0) \in \mathcal{A}'$. Let $\mathcal{A}''$ be the restriction of $\mathcal{A}'$ to those individuals names that are reachable along role assertions from individual names that are roots, i.e., that occur already in $\mathcal{A}$ and are paths of length one. Since q contains at least one individual name (which must be a root) and is connected, we have $\mathcal{A}'', \mathcal{T} \models q$. Let $\widehat{\mathcal{A}}$ be $\mathcal{A}$ without $R_1(a_1, a_0)$ and let $h : \mathsf{Ind}(\mathcal{A}'') \to \mathsf{Ind}(\mathcal{A})$ be the identity on $\mathsf{Ind}(\mathcal{A}'') \cap \mathsf{Ind}(\mathcal{A})$ and such that $h(b_i) = a_2$ for all $b_i \in \mathsf{Ind}(\mathcal{A}'')$ and $h(c_i) = a_1$ for all $c_i \in \mathsf{Ind}(\mathcal{A}'')$. One can prove the following by induction on i.

Claim. For all $i \geq 0$, $p \in \mathsf{Ind}(\mathcal{A}'')$, and $A \in \mathsf{N_C}$, $A(h(p)) \in f^i_{\mathcal{T}}(\mathcal{A}'')$ implies $A(p) \in f^i_{\mathcal{T}}(\widehat{\mathcal{A}})$.

Since $\mathcal{A}'', \mathcal{T} \models q$, by Lemma 7 there is a match π of $q_{\mathcal{T}}$ in $\mathcal{I}_{f^\infty_{\mathcal{T}}(\mathcal{A}'')}$. By the above claim, π is also a match of $q_{\mathcal{T}}$ in $\mathcal{I}_{f^\infty_{\mathcal{T}}(\widehat{\mathcal{A}})}$, thus $\widehat{\mathcal{A}}, \mathcal{T} \models q$ by Lemma 7, in contradiction to Lemma 11.

"only if". By construction of $\mathcal{A}_\psi$, $\delta_{\mathcal{A}'} \models \psi$ implies that, up to renaming of individual names that do not occur in q, we have $\mathcal{A} \subseteq \mathcal{A}'$. Since $\mathcal{A}, \mathcal{T} \models q$ by choice of $\mathcal{A}$, we must thus also have $\mathcal{A}', \mathcal{T} \models q$. ❑

Theorem 7 ($\mathcal{ELI}$ dichotomy). Let q be a connected Boolean CQ and $\mathcal{T}$ an $\mathcal{ELI}$-TBox. Then q is in PTime relative to $\mathcal{T}$ or #P-hard relative to $\mathcal{T}$.

Proof. If q is not FO-rewritable relative to $\mathcal{T}$, then it is #P-hard by Theorem 6. Using the semantics of CQs and of $\mathcal{ELI}$, it is not hard to show that FO-rewritings of CQs relative to $\mathcal{ELI}$-TBoxes are FO-formulas that are preserved under homomorphisms. By Rossman's homomorphism preservation theorem on finite structures [**?**], we thus obtain that whenever there is an FO-rewriting for a CQ q and $\mathcal{ELI}$-TBox $\mathcal{T}$, then there is an FO-rewriting for q and $\mathcal{T}$ that is a UCQ. [3] Thus Theorem 1 and the PTime vs. #P dichotomy for UCQs over tuple independent databases [8,31] immediately yields the dichotomy. ❑

[3] We are grateful to Meghyn Bienvenu for suggesting this argument.