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

P. Cudré-Mauroux et al. (Eds.): ISWC 2012, Part I, LNCS 7649, pp. 182–197, 2012.

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 $x_1, \ldots, x_k$. 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

$$\mathcal{T} = \{ \quad \exists \mathsf{playsfor} \sqsubseteq \mathsf{Player} \qquad \mathsf{Player} \sqsubseteq \exists \mathsf{playsfor}$$
$$\exists \mathsf{playsfor}^- \sqsubseteq \mathsf{SoccerClub} \qquad \mathsf{Striker} \sqsubseteq \mathsf{Player} \quad \}$$

Consider the following subcases considered above.

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

$$\mathsf{Player}(\mathsf{messi}) \rightsquigarrow e_1 \quad \mathsf{playsfor}(\mathsf{messi}, \mathsf{FCbarca}) \rightsquigarrow e_3$$
$$\mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca}) \rightsquigarrow e_3 \quad \mathsf{playsfor}(\mathsf{pedro}, \mathsf{FCbarca}) \rightsquigarrow e_3$$

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 $\mathsf{Player}(\mathsf{villa})$ 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

$$\mathsf{playsfor}(\mathsf{messi}, \mathsf{FCbarca}) \rightsquigarrow e_3 \quad \mathsf{Striker}(\mathsf{messi}) \rightsquigarrow e_4$$
$$\mathsf{playsfor}(\mathsf{villa}, \mathsf{FCbarca}) \rightsquigarrow e_3 \quad \mathsf{Striker}(\mathsf{villa}) \rightsquigarrow e'_4$$
$$\mathsf{playsfor}(\mathsf{pedro}, \mathsf{FCbarca}) \rightsquigarrow e_3 \quad \mathsf{Striker}(\mathsf{pedro}) \rightsquigarrow e''_4$$

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 $\approx$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 $q = q'$.

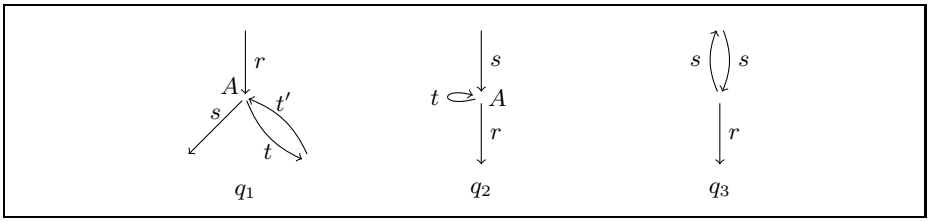

Fig. 1. Example queries

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

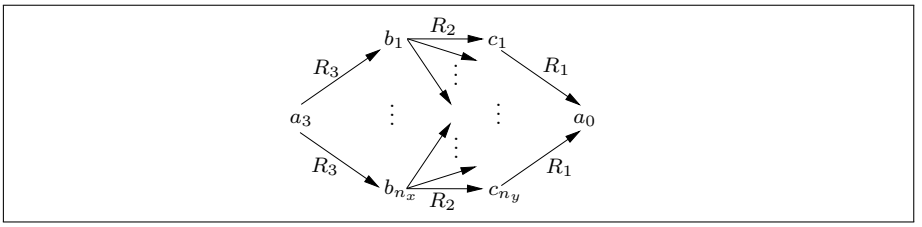

Fig. 2. Gadget for the #P-hardness proof

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, pp. 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, pp. 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, pp. 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. In: 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, pp. 267–270. ACM (2012)
14. Gottlob, G., Lukasiewicz, T., Simari, G.I.: Conjunctive Query Answering in Probabilistic Datalog+/– Ontologies. In: Rudolph, S., Gutierrez, C. (eds.) RR 2011. LNCS, vol. 6902, pp. 77–92. Springer, Heidelberg (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, pp. 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., Lipski Jr., W.: 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, pp. 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, pp. 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. 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, pp. 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)