Ontology-Based Access to Probabilistic Data

Jean Christoph Jung and Carsten Lutz

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

Abstract. We propose a framework for querying probabilistic data in the presence of an ontology, arguing that the interplay of probabilities and ontologies is fruitful in applications such as managing data that was extracted from the web. The prime inference problem is computing answer probabilities, and we show that it can be implemented using standard probabilistic database systems, similar to traditional ontology-based data access. We demonstrate that query rewriting into first-order logic is an important tool for our framework. First, it is used to establish a PTime vs. #P dichotomy for the data complexity of this problem by lifting a corresponding result from probabilistic databases. Then, we use it to characterize which pairs of query and TBox are in PTime. Finally, it is shown that non-existence of such a rewriting implies #P-hardness.

1 Introduction

In recent years, *ontology based data access* (OBDA) has become an active area of description logic research. In OBDA, an ontology provides a semantics for *incomplete* data with the aim of facilitating the computation of more complete answers to queries. There are applications, though, in which it is necessary to query data that is not only incomplete, but also *uncertain*. For instance, data extracted from web sources [24] such as an estate agents' web page is typically incomplete. It is also uncertain because web sources tend to be unreliable and extraction tools are based on heuristic decisions and thus significantly error prone. In this paper, we propose an extension of OBDA that captures uncertain data through a probabilistic data model and replaces the computation of certain answers with computing the probabilities of certain answers. In brief, our approach relates to probabilistic database systems (PDBMSs) in the same way that traditional OBDA relates to RDBMSs.

Framework. We consider probabilistic ABox formalisms that are inspired by data models from the currently very active area of probabilistic databases [7,32]. Specifically, *pABoxes* enrich classical ABoxes with probabilities that are attached to *probabilistic events*, and with *event expressions* that are attached to ABox assertions. For example, a pABox assertion $\mathsf{SoccerPlayer}(\mathsf{messi})$ 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'. Event e_1 can then be associated with probability .7, and e_2 with .9. Events are assumed to be probabilistically independent, which results

in a straightforward semantics that is similar to well-known probabilistic versions of datalog [28,12]. Ontologies are represented by description logic TBoxes. In this setting, which we call *ontology based access to probabilistic data (pOBDA)*, we are interested in *computing the probabilities of certain answers* to *conjunctive queries (CQs)*. Note that uncertainty occurs only in the data, but neither in the ontology nor in the query. We believe that pOBDA 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, the ontology can help to reduce the uncertainty of the data.

Contributions. The main aim of this paper is to study the *data complexity* of pOBDA. More precisely, we pursue a *non-uniform* approach as recently initiated in [27], which aims at fully classifying the data complexity of every pair $(q, \mathcal{T})$ that consists of a CQ q and a TBox formulated in a fixed 'master DL'. As a central tool of our analysis, we use *query rewriting* into first-order (FO) queries, which is an important technique for traditional OBDA [6]. We start with showing that FO-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 a pABox $\mathcal{A}$ relative to $\mathcal{T}$ is identical to the probability that $\boldsymbol{a}$ is an answer to the FO-rewriting $q_{\mathcal{T}}$ of q and $\mathcal{T}$ over $\mathcal{A}$ viewed as a probabilistic database. This *lifting* of FO-rewritings to the probabilistic case immediately implies that one can implement pOBDA based on existing PDBMSs such as MayBMS, Trio, and MystiQ [1,35,5].

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 TBoxes formulated in (the core version of) DL-Lite and to ipABoxes, which are a special case of pABoxes in which all ABox assertions are probabilistically independent. Based on a careful syntactic analysis, we provide a concrete 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 $\mathcal{ELI}$ 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 some $(q, \mathcal{T})$ 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. This negative result is relativized by the fact that answer probabilities can often be efficiently approximated. In particular, all pairs $(q, \mathcal{T})$ admit approximation in terms of a *fully polynomial randomized approximation scheme (FPRAS)* whenever q is FO-rewritable relative to $\mathcal{T}$. We provide a brief

discussion of such approximations and refer to the full version [19] of this paper for more information.

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 [28,12] and references therein. There have recently been other approaches to combining ontologies and uncertainty 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 in the presence of ontologies. Note that the analogous property of being implementable using state-of-the-art RDBMSs is a favorable feature of traditional OBDA [6]. There has also been a large number of proposals for enriching description logic TBoxes (instead of ABoxes) with probabilities, see [25,26] 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.

This paper is a condensed version of [19]. For the missing proofs and some additional material we refer the reader to the long version of [19].

2 Preliminaries

We use standard notation for the syntax and semantics of description logics (DLs) and refer to [3] for full details. As usual, 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 R^- denotes r. We consider the ontology languages *DL-Lite* and $\mathcal{ELI}$. Regarding the former, we concentrate on the dialect DL-Lite$_{\mathsf{core}}$, where *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. In $\mathcal{ELI}$, a *TBox* is a finite set of CIs $C \sqsubseteq D$ where C and D are (potentially) compound concepts of the form $\top$, A, $C' \sqcap D'$, $\exists r.C'$, and $\exists r^-.C'$.

As usual, an *ABox* 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 write $r^-(a, b) \in \mathcal{A}$ for $r(b, a) \in \mathcal{A}$. The semantics of DLs is based on interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$; we adopt the unique name assumption (UNA), i.e., we require $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all individuals $a \neq b$.

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 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 individual names $\boldsymbol{a} = a_1 \cdots a_k$, 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 individual names $a \in \mathsf{term}(q)$, $\pi(t) \in A^{\mathcal{I}}$ 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 $\mathcal{I}$ and let $\mathsf{ans}(q,\mathcal{I})$ denote the set of all $\boldsymbol{a}$ with $\mathcal{I} \models q[\boldsymbol{a}]$. 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 $\mathcal{A}$. The set $\mathsf{cert}_{\mathcal{T}}(q,\mathcal{A})$ of all *certain answers* consists of all tuples $\boldsymbol{a}$ over $\mathsf{Ind}(\mathcal{A})$ with $\mathcal{T},\mathcal{A} \models q[\boldsymbol{a}]$.

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 $E \subseteq \mathcal{E}$* 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 an 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 $p(e_5) = 0.5$.

We now definte the semantics of OBDA over pABoxes. 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:

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

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

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

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. Nowadays, there is an abundance of probabilistic data models that provide compact representation of distributions over potentially large sets of *possible worlds*, see [15,30,2] and the references therein. 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]. 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 [17,26].

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 aim of this section is to show that *FO-rewriting*, a prominent approach to traditional OBDA, are fruitful also in the case of computing probabilities of certain answers 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

A first-order query $q_{\mathcal{T}}$ is an *FO-rewriting* of a CQ q and an FO-TBox $\mathcal{T}$ (i.e., a first-order theory) if $\mathsf{cert}_{\mathcal{T}}(q,\mathcal{A}) = \mathsf{ans}(q_{\mathcal{T}},\mathcal{I}_{\mathcal{A}})$ for every ABox $\mathcal{A}$, where $\mathcal{I}_{\mathcal{A}}$ to denotes the ABox $\mathcal{A}$ viewed as an interpretation. The query rewriting approach to traditional OBDA consists of computing, given a CQ q and a TBox $\mathcal{T}$, an FO-rewriting $q_{\mathcal{T}}$ and then handing it over for execution to a relational database system that stores the ABox $\mathcal{A}$.

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 [1,35,5]. 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 [33]. We pursue a non-uniform approach to the complexity of query answering in probabilistic OBDA, as recently initiated in [27]: 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 $\mathcal{T}$.

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.*

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 3 (ipABox). *An* assertion-independent pABox *(or* 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 implicit (one atomic event per ABox assertion) and we write $p(\alpha)$ to denote the probability of the event associated with the assertion α. Analogously to pABoxes, all events (and thus all assertions) are independent. Note that the answer probability of $\mathbf{a} \in \mathsf{Ind}(\mathcal{A})^n$ to an n-ary conjunctive query q over an ipABox $(\mathcal{A}, p)$ relative to a TBox $\mathcal{T}$ simplifies to

$$p_{\mathcal{A},\mathcal{T}}(\boldsymbol{a} \in q) = \sum_{\mathcal{A}' \subseteq \mathcal{A} \,:\, \boldsymbol{a} \in \mathsf{cert}_{\mathcal{T}}(q, \mathcal{A}')} p(\mathcal{A}')$$

where $p(\mathcal{A}') = \prod_{\alpha \in \mathcal{A}'} p(\alpha) \cdot \prod_{\alpha \in \mathcal{A} \setminus \mathcal{A}'} (1 - p(\alpha))$.

Reconsidering our web data extraction example, it turns out that cases (1) and (4) yield ipABoxes, whereas cases (2), (3), and (5) do not. We refer to [32] 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.

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

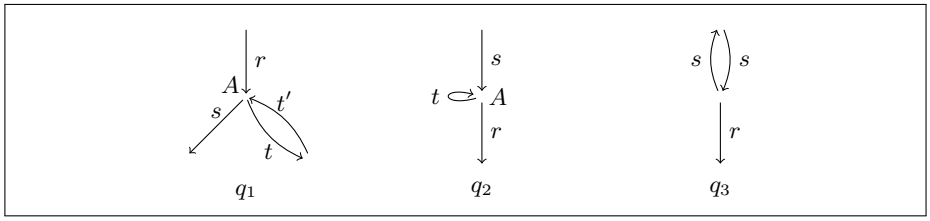

Fig. 1. Example queries

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. 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 section, 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, 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 4 (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 $q = q'$.

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 finish anlyzing the dichotomy, it thus remains to analyze simple tree queries. We say that a role R *is $\mathcal{T}$-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 [10].

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 $\{S_1(x, y), \ldots, S_k(x, y)\}$ for roles $S_1, \ldots, S_k$.*

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 $\mathcal{T}$.

Approximations. Theorems 4 and 5 show that only very simple pairs $(q, \mathcal{T})$ can be answered in PTime. Hence, it is reasonable to consider the approximation of answer probabilities. Of particular relevance are *fully-polynomial randomized approximation schemes (FPRASes)*, see [22,34] for a definition and more details. In [9] it is observed that there is an FPRAS for every UCQ over a probabilistic database. As a consequence of Theorem 1, there is thus also an FPRAS for every pair $(q, \mathcal{T})$ such that q is FO-rewritable relative to $\mathcal{T}$; in particular, there are FPRASes for every CQ and DL-Lite ontology even if additional features such

as role inclusions are admitted. This observation clearly gives hope for practical feasibility of probabilistic OBDA. In the full paper, we also consider the existence of FPRASes for more expressive ontology languages [19].

5 Beyond Query Rewriting

A CQ q is *FO-rewritable* relative to a TBox $\mathcal{T}$ if there is an FO-rewriting of q relative to $\mathcal{T}$. The aim of this section is to establish that, in a sense, FO-rewriting is a *complete* tool for proving PTIME results for CQ answering in probabilistic OBDA: we show 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 $\mathcal{ELI}$ as the TBox language because it properly generalizes DL-Lite (as in the previous sections we disregard $\bot$) and, unlike DL-Lite, also embraces non FO-rewritable CQs/TBoxes. 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 $q[\boldsymbol{a}]$. The following theorem says that there is no hope for PTIME algorithms in the case a query q is not FO-rewritable relative to $\mathcal{T}$.

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 $\mathcal{T}$.*

As a by-product of Theorem 6 we obtain the following dichotomy.

Theorem 7 ($\mathcal{ELI}$ dichotomy). *Let q be a rooted 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 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: on the one hand, it can be extended to CQs that involve constants and are not necessarily connected; on the other hand, one could study more expressive versions of DL-Lite that, for example, allow role hierarchy statements in the TBox. It would also be worthwhile to add means to express uncertainty to the TBox formalism instead of admitting it only in the ABox; this is done for example in [28,12], but it remains to be seen whether the semantics used there is appropriate for our purposes.

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. J. ACM 59(6), 30 (2012)
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. Jung, J.C., Lutz, C.: Ontology-based access to probabilistic data with OWL QL. In: Proc. of ISWC. pp. 182–197. Springer (2012)
20. Jerrum, M., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. 43, 169–188 (1986)
21. 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)
22. Karp, R.M., Luby, M.: Monte-carlo algorithms for enumeration and reliability problems. In Proc. of FoCS. 56–64. IEEE Computer Society (1983)

23. 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)
24. 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)
25. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Sem. 6(4), 291–308 (2008)
26. Lutz, C., Schröder, L.: Probabilistic description logics for subjective uncertainty. In Proc. of KR. AAAI Press (2010)
27. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: Proc. of KR. AAAI Press (2012)
28. 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)
29. Rossman, B.: Homomorphism preservation theorems. J. ACM. 55(3). 1–54 (2008).
30. Sarma, A.D., Benjelloun, O., Halevy, A.Y., Widom, J.: Working models for uncertain data. In: Proc. of ICDE. IEEE Computer Society (2006)
31. Straccia, U.: Top-k retrieval for ontology mediated access to relational databases. In: Information Sciences. 108, 1–23 (2012).
32. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
33. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comput. 8(3), 410–421 (1979)
34. Vazirani, V.V.: Approximation algorithms. Springer (2001)
35. Widom, J.: Trio: A system for integrated management of data, accuracy, and lineage. In Proc. of CIDR. 262–276 (2005)
36. Zenklusen, R., Laumanns, M.: High-confidence estimation of small s-t reliabilities in directed acyclic networks. Networks 57(4), 376–388 (2011)