Probabilistic Ontological Data Exchange with Bayesian Networks

Thomas Lukasiewicz[1], Maria Vanina Martinez[3],
Livia Predoiu[1,2], and Gerardo I. Simari[3]

[1] Department of Computer Science, University of Oxford, UK
[2] Department of Computer Science, Otto-von-Guericke Universität Magdeburg, Germany
[3] Dept. of Comp. Sci. and Eng., Univ. Nacional del Sur and CONICET, Argentina

Abstract. We study the problem of exchanging probabilistic data between ontology-based probabilistic databases. The probabilities of the probabilistic source databases are compactly encoded via Boolean formulas with the variables adhering to the dependencies imposed by a Bayesian network, which are closely related to the management of provenance. For the ontologies and the ontology mappings, we consider different kinds of existential rules from the Datalog+/– family. We provide a complete picture of the computational complexity of the problem of deciding whether there exists a probabilistic (universal) solution for a given probabilistic source database relative to a (probabilistic) ontological data exchange problem. We also analyze the complexity of answering UCQs (unions of conjunctive queries) in this framework.

1 Introduction

Large volumes of uncertain data are best modeled, stored, and processed in probabilistic databases [22]. Enriching databases with terminological knowledge encoded in ontologies has recently gained increasing importance in the form of ontology-based data access (OBDA) [21]. A crucial problem in OBDA is to integrate and exchange knowledge. Not only in the context of OBDA, but also in the area of the Semantic Web, there are distributed ontologies that we may have to map and integrate to enable query answering over them. Here, apart from the uncertainty attached to source databases, there may also be uncertainty regarding the ontology mappings establishing the proper correspondence between items in the source ontology and items in the target ontology. This especially happens when the mappings are created automatically.

Data exchange [11] is an important theoretical framework used for studying data-interoperability tasks that require data to be transferred from existing databases to a target database that comes with its own (independently created) schema and schema constraints. The expressivity of the data exchange framework goes beyond the classical data integration framework [17]. For the translation, schema mappings are used, which are declarative specifications that describe the relationship between two database schemas. In classical data exchange, we have a source database, a target database, a deterministic mapping, and deterministic target dependencies. Recently, a framework for probabilistic data exchange [10] has been proposed where the classical data exchange

framework based on weakly acyclic existential rules has been extended to consider a probabilistic source database and a probabilistic source-to-target mapping.

In this paper, we study an expressive extension of the probabilistic data exchange framework in [10], where the source and the target are ontological knowledge bases, each consisting of a probabilistic database and a deterministic ontology describing terminological knowledge about the data stored in the database. The two ontologies and the mapping between them are expressed via existential rules. Our extension of the data exchange framework is strongly related to exchanging data between incomplete databases, as proposed in [3], which considers an incomplete deterministic source database in the data exchange problem. However, in that work, the databases are deterministic, and the mappings and the target database constraints are full existential rules only. In our complexity analysis in this paper, we consider a host of different classes of existential rules, including some subclasses of full existential rules. In addition, our source is a probabilistic database relative to an underlying ontology.

Our work in this paper is also related to the recently proposed knowledge base exchange framework [2, 1], which allows knowledge to be exchanged between deterministic *DL-Lite*$_{RDFS}$ and *DL-Lite*$_{\mathcal{R}}$ ontologies. In this paper, besides considering probabilistic source databases, we are also using more expressive ontology languages, since already linear existential rules from the Datalog+/– family are strictly more expressive than the description logics (DLs) *DL-Lite*$_X$ of the *DL-Lite* family [9] as well as their extensions with n-ary relations *DLR-Lite*$_X$. Guarded existential rules are sufficiently expressive to model the tractable DL $\mathcal{EL}$ [4, 5] (and $\mathcal{ELI}^f$ [16]). Note that existential rules are also known as tuple-generating dependencies (TGDs) and Datalog+/– rules [7].

The main contributions of this paper are summarized as follows.

– We introduce deterministic and probabilistic ontological data exchange problems, where probabilistic knowledge is exchanged between two Bayesian network-based probabilistic databases relative to their underlying deterministic ontologies, and the deterministic and probabilistic mapping between the two ontologies is defined via deterministic and probabilistic existential mapping rules, respectively.

– We provide an in-depth analysis of the data and combined complexity of deciding the existence of probabilistic (universal) solutions and obtain a (fairly) complete picture of the data complexity, general combined complexity, bounded-arity (*ba*) combined, and fixed-program combined (*fp*) complexity for the main sublanguages of the Datalog+/– family. We also delineate some tractable special cases, and provide complexity results for exact UCQ (union of conjunctive queries) answering.

– For the complexity analysis, we consider a compact encoding of probabilistic source databases and mappings, which is used in the area of both incomplete and probabilistic databases, and also known as data provenance or data lineage [14, 12, 13, 22]. Here, we consider data provenance for probabilistic data that is structured according to an underlying Bayesian network.

2 Preliminaries

We assume infinite sets of *constants* $\mathbf{C}$, *(labeled) nulls* $\mathbf{N}$, and regular *variables* $\mathbf{V}$. A *term* t is a constant, null, or variable. An *atom* has the form $p(t_1, \ldots, t_n)$, where p is

an n-ary predicate, and $t_1, \ldots, t_n$ are terms. Conjunctions of atoms are often identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms $p(\mathbf{t})$, where $\mathbf{t}$ is a tuple of constants and nulls. A *database* D is a finite instance that contains only constants. A *homomorphism* is a substitution $h : \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \rightarrow \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ that is the identity on $\mathbf{C}$. We assume familiarity with *conjunctive queries (CQs)*. The answer to a CQ q over an instance I is denoted $q(I)$. A Boolean CQ (BCQ) q evaluates to *true* over I, denoted $I \models q$, if $q(I) \neq \varnothing$.

A *tuple-generating dependency (TGD)* σ is a first-order formula $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \exists \mathbf{Y}\, p(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} \cup \mathbf{Y} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Y})$ is an atom. We call $\varphi(\mathbf{X})$ the *body* of σ, denoted $body(\sigma)$, and $p(\mathbf{X}, \mathbf{Y})$ the *head* of σ, denoted $head(\sigma)$. We consider only TGDs with a single atom in the head, but our results can be extended to TGDs with a conjunction of atoms in the head. An instance I *satisfies* σ, written $I \models \sigma$, if the following holds: whenever there exists a homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$, then there exists $h' \supseteq h|_{\mathbf{X}}$, where $h|_{\mathbf{X}}$ is the restriction of h to $\mathbf{X}$, such that $h'(p(\mathbf{X}, \mathbf{Y})) \in I$. A *negative constraint (NC)* ν is a first-order formula $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bot$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms, called the *body* of ν, denoted $body(\nu)$, and $\bot$ denotes the truth constant $false$. An instance I *satisfies* ν, denoted $I \models \nu$, if there is no homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. Given a set Σ of TGDs and NCs, I *satisfies* Σ, denoted $I \models \Sigma$, if I satisfies each TGD and NC of Σ. For brevity, we omit the universal quantifiers in front of TGDs and NCs.

Given a database D and a set Σ of TGDs and NCs, the answers that we consider are those that are true in *all* models of D and Σ. Formally, the *models* of D and Σ, denoted $mods(D, \Sigma)$, is the set of instances $\{I \mid I \supseteq D \text{ and } I \models \Sigma\}$. The *answer* to a CQ q relative to D and Σ is defined as the set of tuples $ans(q, D, \Sigma) = \bigcap_{I \in mods(D,\Sigma)} \{\mathbf{t} \mid \mathbf{t} \in q(I)\}$. The answer to a BCQ q is *true*, denoted $D \cup \Sigma \models q$, if $ans(q, D, \Sigma) \neq \varnothing$. The problem of *CQ answering* is defined as follows: given a database D, a set Σ of TGDs and NCs, a CQ q, and a tuple of constants $\mathbf{t}$, decide whether $\mathbf{t} \in ans(q, D, \Sigma)$. Following Vardi's taxonomy [23], the *combined complexity* of BCQ answering is calculated by considering all the components, i.e., the database, the set of dependencies, and the query, as part of the input. The *bounded-arity combined complexity* (or simply *ba-combined complexity*) is calculated by assuming that the arity of the underlying schema is bounded by an integer constant. Notice that in the context of description logics (DLs), whenever we refer to the combined complexity in fact we refer to the ba-combined complexity since, by definition, the arity of the underlying schema is at most two. The *fixed-program combined complexity* (or simply *fp-combined complexity*) is calculated by considering the set of TGDs and NCs as fixed.

3 Ontological Data Exchange

In this section, we define the notions of *deterministic* and *probabilistic ontological data exchange*. The source (resp., target) of the deterministic/probabilistic ontological data exchange problems that we consider in this paper is a probabilistic database (resp., probabilistic instance), each relative to a deterministic ontology. Here, a *probabilistic database* (resp., *probabilistic instance*) over a schema $\mathbf{S}$ is a probability space $Pr = (\mathcal{I}, \mu)$ such that $\mathcal{I}$ is the set of all (possibly infinitely many) databases (resp., instances) over $\mathbf{S}$, and $\mu \colon \mathcal{I} \rightarrow [0, 1]$ is a function that satisfies $\sum_{I \in \mathcal{I}} \mu(I) = 1$.

3.1 Deterministic Ontological Data Exchange

Ontological data exchange formalizes data exchange from a probabilistic database relative to a source ontology Σ_s (consisting of TGDs and NCs) over a schema $\mathbf{S}$ to a probabilistic target instance Pr_t relative to a target ontology Σ_t (consisting of a set of TGDs and NCs) over a schema $\mathbf{T}$ via a (source-to-target) mapping (also consisting of a set of TGDs and NCs). More specifically, an *ontological data exchange (ODE) problem* $\mathcal{M}$= ($\mathbf{S}$, $\mathbf{T}$, Σ_s, Σ_t, Σ_{st}) consists of (i) a source schema $\mathbf{S}$, (ii) a target schema $\mathbf{T}$ disjoint from $\mathbf{S}$, (iii) a finite set Σ_s of TGDs and NCs over $\mathbf{S}$ (called *source ontology*), (iv) a finite set Σ_t of TGDs and NCs over $\mathbf{T}$ (called *target ontology*), and (v) a finite set Σ_{st} of TGDs and NCs σ over $\mathbf{S} \cup \mathbf{T}$ (called *(source-to-target) mapping*) such that *body*(σ) and *head*(σ) are defined over $\mathbf{S} \cup \mathbf{T}$ and $\mathbf{T}$, respectively.

Ontological data exchange with deterministic databases is based on defining a target instance J over $\mathbf{T}$ as being a *solution* for a deterministic source database I over $\mathbf{S}$ relative to an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$, if $(I \cup J) \models \Sigma_s \cup \Sigma_t \cup \Sigma_{st}$. We denote by $Sol_{\mathcal{M}}$ the set of all such pairs (I, J). Among the possible deterministic solutions J to a deterministic source database I relative to $\mathcal{M}$ in $Sol_{\mathcal{M}}$, we prefer *universal* solutions, which are the most general ones carrying only the necessary information for data exchange, i.e., those that transfer only the source database along with the relevant implicit derivations via Σ_s to the target ontology. A universal solution can be homomorphically mapped to all other solutions leaving the constants unchanged. Hence, a deterministic target instance J over $\mathbf{S}$ is a *universal solution* for a deterministic source database I over $\mathbf{T}$ relative to a schema mapping $\mathcal{M}$, if (i) J is a solution, and (ii) for each solution J' for I relative to $\mathcal{M}$, there is a homomorphism $h\colon J \to J'$. We denote by $USol_{\mathcal{M}}$ ($\subseteq Sol_{\mathcal{M}}$) the set of all pairs (I, J) of deterministic source databases I and target instances J such that J is a universal solution for I relative to $\mathcal{M}$.

When considering probabilistic databases and instances, a joint probability space Pr over the solution relation $Sol_{\mathcal{M}}$ and the universal solution relation $USol_{\mathcal{M}}$ must exist. More specifically, a probabilistic target instance $Pr_t = (\mathcal{J}, \mu_t)$ is a *probabilistic solution* (resp., *probabilistic universal solution*) for a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$ relative to an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$, if there exists a probability space $Pr = (\mathcal{I} \times \mathcal{J}, \mu)$ such that (i) the left and right marginals of Pr are Pr_s and Pr_t, respectively, i.e., (i.a) $\mu_s(I) = \sum_{J \in \mathcal{J}} \mu(I, J)$ for all $I \in \mathcal{I}$, (i.b) $\mu_t(J) = \sum_{I \in \mathcal{I}} \mu(I, J)$ for all $J \in \mathcal{J}$; and (ii) $\mu(I, J) = 0$ for all $(I, J) \notin Sol_{\mathcal{M}}$ (resp., $(I, J) \notin USol_{\mathcal{M}}$). Note that this intuitively says that all non-solutions (I, J) have probability zero and the existence of a solution does not exclude that some source databases with probability zero have no corresponding target instance.

Example 1. An ontological data exchange (ODE) problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ is given by the source schema $\mathbf{S} = \{Researcher/2, ResearchArea/2, Publication/3\}$ (the number after each predicate denotes its arity), the target schema $\mathbf{T} = \{URearch\text{-}Area/3, Lecturer/2\}$, the source ontology $\Sigma_s = \{\sigma_s, \nu_s\}$, the target ontology $\Sigma_t = \{\sigma_t, \nu_t\}$, and the mapping $\Sigma_{st} = \{\sigma_{st}, \nu_m\}$, where:

$$\sigma_s : Publication(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) \to ResearchArea(\mathrm{X}, \mathrm{Y}),$$
$$\nu_s : Researcher(\mathrm{X}, \mathrm{Y}) \wedge ResearchArea(\mathrm{X}, \mathrm{Y}) \to \bot,$$
$$\sigma_t : URearchArea(\mathrm{U}, \mathrm{D}, \mathrm{T}) \to \exists \mathrm{Z}\, Lecturer(\mathrm{T}, \mathrm{Z}),$$
$$\nu_t : Lecturer(\mathrm{X}, \mathrm{Y}) \wedge Lecturer(\mathrm{Y}, \mathrm{X}) \to \bot,$$

Possible source database facts

r_a	*Researcher*(Alice, UnivOx)
r_p	*Researcher*(Paul, UnivOx)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Derived source database facts

a_{aml}	*ResearchArea*(Alice, ML)
a_{adb}	*ResearchArea*(Alice, DB)
a_{pdb}	*ResearchArea*(Paul, DB)
a_{pai}	*ResearchArea*(Paul, AI)

Probabilistic source database $Pr_s = (I, \mu_s)$

$I_1 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{aml}, \mathrm{p}_{pdb}, \mathrm{a}_{aml}, \mathrm{a}_{pdb}\}$	0.5
$I_2 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{aml}, \mathrm{p}_{pai}, \mathrm{a}_{aml}, \mathrm{a}_{pai}\}$	0.2
$I_3 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{adb}, \mathrm{p}_{pai}, \mathrm{a}_{adb}, \mathrm{a}_{pai}\}$	0.15
$I_4 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{adb}, \mathrm{p}_{pdb}, \mathrm{a}_{adb}, \mathrm{a}_{pdb}\}$	0.075
$I_5 = \{\mathrm{r}_a, \mathrm{p}_{adb}, \mathrm{a}_{adb}\}$	0.075

Possible target instance facts

u_{ml}	*UResearchArea*(UnivOx, N_1, ML)
u_{ai}	*UResearchArea*(UnivOx, N_2, AI)
u_{db}	*UResearchArea*(UnivOx, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic target instance $Pr_{t_1} = (J_1, \mu_{t_1})$

$J_1 = \{\mathrm{u}_{ml}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{db}\}$	0.5
$J_2 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{l}_{ml}, \mathrm{l}_{ai}\}$	0.2
$J_3 = \{\mathrm{u}_{ai}, \mathrm{u}_{db}, \mathrm{l}_{ai}, \mathrm{l}_{db}\}$	0.15
$J_4 = \{\mathrm{u}_{db}, \mathrm{l}_{db}\}$	0.15

Probabilistic target instance $Pr_{t_2} = (J_2, \mu_{t_2})$

$J_5 = \{\mathrm{u}_{ml}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{db}\}$	0.55
$J_6 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{l}_{ml}, \mathrm{l}_{ai}\}$	0.1
$J_7 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{ai}, \mathrm{l}_{db}\}$	0.35

Table 1. Probabilistic source database and two probabilistic target instances for Example 1 ($N_1, \ldots, N_6$ are nulls); both are probabilistic solutions, but only Pr_{t_1} is universal.

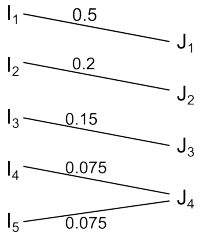

Fig. 1. Probabilistic universal solution Pr_{t_1}.

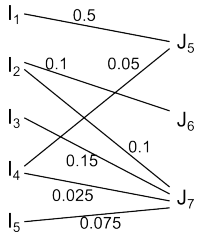

Fig. 2. Probabilistic solution Pr_{t_2}.

σ_{st} : *ResearchArea*(N, T) $\wedge$ *Researcher*(N, U) $\rightarrow \exists$D *UResearchArea*(U, D, T),
ν_{st} : *ResearchArea*(N, T) $\wedge$ *UResearchArea*(U, T, N) $\rightarrow \bot$.

Given the probabilistic source database in Table 1, two probabilistic instances $Pr_{t_1} = (\mathcal{J}_1, \mu_{t_1})$ and $Pr_{t_2} = (\mathcal{J}_2, \mu_{t_2})$ that are probabilistic solutions are shown in Table 1. Note that only Pr_{t_1} is also a probabilistic universal solution. Note also that Figures 1 and 2 show the probability spaces over Pr_{t_1} and Pr_{t_2}, respectively. ■

Query answering in ontological data exchange is performed over the target ontology and is generalized from deterministic data exchange. A *union of conjunctive queries* (or *UCQ*) has the form $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i \ \Phi_i(\mathbf{X}, \mathbf{Y}_i, \mathbf{C}_i)$, where each $\exists \mathbf{Y}_i \, \Phi_i(\mathbf{X}, \mathbf{Y}_i, \mathbf{C}_i)$ with $i \in \{1, \ldots, k\}$ is a CQ with exactly the variables $\mathbf{X}$ and $\mathbf{Y}_i$, and the constants $\mathbf{C}_i$. Given an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$, probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$, UCQ $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i \, \Phi_i(\mathbf{X}, \mathbf{Y}_i, \mathbf{C}_i)$, and tuple $\mathbf{t}$ (a ground instance of $\mathbf{X}$ in q) over $\mathbf{C}$, the *confidence* of $\mathbf{t}$ relative to q, denoted $conf_q(\mathbf{t})$, in Pr_s relative to $\mathcal{M}$ is the infimum of $Pr_t(q(\mathbf{t}))$ subject to all probabilistic solutions Pr_t for Pr_s relative to $\mathcal{M}$. Here, $Pr_t(q(\mathbf{t}))$ for $Pr_t = (\mathcal{J}, \mu_t)$ is the sum of all $\mu_t(J)$ such that $q(\mathbf{t})$ evaluates

to true in the instance $J \in \mathcal{J}$ (i.e., some BCQ $\exists \mathbf{Y}_i \, \Phi_i(\mathbf{t}, \mathbf{Y}_i, \mathbf{C}_i)$ with $i \in \{1, \ldots, k\}$ evaluates to true in J).

Example 2. Consider again the setting of Example 1, and let q be a UCQ of a student who wants to know whether she can study either machine learning or artificial intelligence at the University of Oxford: $q() = \exists \mathrm{X}, \mathrm{Z}(\mathit{Lecturer}(\mathit{AI}, \mathrm{X}) \wedge \mathit{UResearchArea}(\mathit{UnivOx}, \mathrm{Z}, \mathrm{AI})) \vee \exists \mathrm{X}, \mathrm{Z}(\mathit{Lecturer}(\mathit{ML}, \mathrm{X}) \wedge \mathit{UResearchArea}(\mathit{UnivOx}, \mathrm{Z}, \mathit{ML}))$. Then, q yields the probabilities 0.85 and 1 on Pr_{t_1} and Pr_{t_2}, respectively. ∎

3.2 Probabilistic Ontological Data Exchange

Probabilistic ontological data exchange extends deterministic ontological data exchange by turning the deterministic source-to-target mapping into a probabilistic source-to-target mapping, i.e., we have a probability distribution over the set of all subsets of Σ_{st}. More specifically, a *probabilistic ontological data exchange (PODE) problem* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$ consists of (i) a source schema $\mathbf{S}$, (ii) a target schema $\mathbf{T}$ disjoint from $\mathbf{S}$, (iii) a finite set Σ_s of TGDs and NCs over $\mathbf{S}$ (called *source ontology*), (iv) a finite set Σ_t of TGDs and NCs over $\mathbf{T}$ (called *target ontology*), (v) a finite set Σ_{st} of TGDs and NCs σ over $\mathbf{S} \cup \mathbf{T}$, and (vi) a function $\mu_{st} \colon 2^{\Sigma_{st}} \to [0, 1]$ such that $\sum_{\Sigma' \subseteq \Sigma_{st}} \mu_{st}(\Sigma') = 1$ (called *probabilistic (source-to-target) mapping*).

A probabilistic target instance $\mathit{Pr}_t = (\mathcal{J}, \mu_t)$ is a *probabilistic solution* (resp., *probabilistic universal solution*) for a probabilistic source database $\mathit{Pr}_s = (\mathcal{I}, \mu_s)$ relative to a PODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$, if there exists a probability space $\mathit{Pr} = = (\mathcal{I} \times \mathcal{J} \times 2^{\Sigma_{st}}, \mu)$ such that: (i) the three marginals of μ are μ_s, μ_t, and μ_{st}, such that: (i.a) $\mu_s(I) = \sum_{J \in \mathcal{J},\, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma')$ for all $I \in \mathcal{I}$, (i.b) $\mu_t(J) = \sum_{I \in \mathcal{I},\, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma')$ for all $J \in \mathcal{J}$, and (i.c) $\mu_{st}(\Sigma') = \sum_{I \in \mathcal{I},\, J \in \mathcal{J}} \mu(I, J, \Sigma')$ for all $\Sigma' \subseteq \Sigma_{st}$; and (ii) $\mu(I, J, \Sigma') = 0$ for all $(I, J) \notin \mathit{Sol}_{(\mathbf{S}, \mathbf{T}, \Sigma')}$ (resp., $(I, J) \notin \mathit{USol}_{(\mathbf{S}, \mathbf{T}, \Sigma')}$).

Using probabilistic (universal) solutions for probabilistic source databases relative to PODE problems, the semantics of UCQs is lifted to PODE problems as follows. Given a PODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$, a probabilistic source database $\mathit{Pr}_s = (\mathcal{I}, \mu_s)$, a UCQ $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i \, \Phi_i(\mathbf{X}, \mathbf{Y}_i, \mathbf{C}_i)$, and a tuple $\mathbf{t}$ (a ground instance of $\mathbf{X}$ in q) over $\mathbf{C}$, the *confidence* of $\mathbf{t}$ relative to q, denoted $\mathit{conf}_q(\mathbf{t})$, in Pr_s relative to $\mathcal{M}$ is the infimum of $\mathit{Pr}_t(q(\mathbf{t}))$ subject to all probabilistic solutions Pr_t for Pr_s relative to $\mathcal{M}$. Here, $\mathit{Pr}_t(q(\mathbf{t}))$ for $\mathit{Pr}_t = (\mathcal{J}, \mu_t)$ is the sum of all $\mu_t(J)$ such that $q(\mathbf{t})$ evaluates to true in the instance $J \in \mathcal{J}$.

3.3 Compact Encoding

We use a compact encoding of both probabilistic databases and probabilistic mappings, which is based on annotating facts, TGDs, and NCs by probabilistic events in a Bayesian network, rather than explicitly specifying the whole probability space.

We first define annotations and annotated atoms. Let $e_1, \ldots, e_n$ be $n \geq 1$ *elementary events*. A *world* w is a conjunction $\ell_1 \wedge \cdots \wedge \ell_n$, where each ℓ_i, $i \in \{1, \ldots, n\}$, is either the elementary event e_i or its negation $\neg e_i$. An *annotation* λ is any Boolean combination of elementary events (i.e., all elementary events are annotations, and if λ_1 and λ_2

	Possible source database facts	Annotation
r_a	*Researcher*(Alice, UnivOx)	true
r_p	*Researcher*(Paul, UnivOx)	$e_1 \vee e_2 \vee e_3 \vee e_4$
p_{aml}	*Publication*(Alice, ML, JMLR)	$e_1 \vee e_2$
p_{adb}	*Publication*(Alice, DB, TODS)	$\neg e_1 \wedge \neg e_2$
p_{pdb}	*Publication*(Paul, DB, TODS)	$e_1 \vee (\neg e_2 \wedge \neg e_3 \wedge e_4)$
p_{pai}	*Publication*(Paul, AI, AIJ)	$(\neg e_1 \wedge e_2) \vee (\neg e_1 \wedge e_3)$

Table 2. Annotation encoding of the probabilistic source database in Table 1.

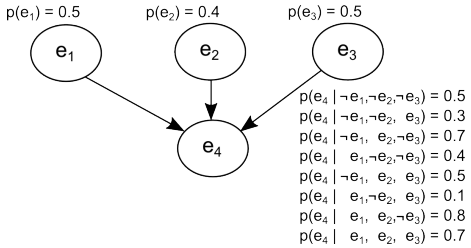

Table 3. Bayesian network over the elementary events.

are annotations, then also $\neg\lambda_1$ and $\lambda_1 \wedge \lambda_2$). An *annotated atom* has the form $a\colon \lambda$, where a is an atom, and λ is an annotation.

The compact encoding of probabilistic databases can then be defined as follows. Note that this encoding is also underlying our complexity analysis in Section 4. A set $\mathbf{A}$ of annotated atoms along with a probability $\mu(w) \in [0,1]$ for every world w *compactly encodes a probabilistic database* $Pr = (\mathcal{I}, \mu)$ whenever: (i) the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and (ii) the probability μ of every subset-maximal database $\{a_1, \ldots, a_m\} \in \mathcal{I}$ [4] such that $\{a_1\colon \lambda_1, \ldots, a_m\colon \lambda_m\} \subseteq \mathbf{A}$ for some annotations $\lambda_1, \ldots, \lambda_m$ is the probability μ of $\lambda_1 \wedge \cdots \wedge \lambda_m$ (and the probability μ of every other database in $\mathcal{I}$ is 0).

We assume that the probability distributions for the underlying events are given by a Bayesian network, which is usually used for compactly specifying a joint probability space, encoding also a certain causal structure between the variables. The following example in Tables 2 and 3 illustrates the compact encoding of probabilistic source databases via Boolean annotations relative to an underlying Bayesian network.

If the mapping is probabilistic as well, then we use two disjoint sets of elementary events, one for encoding the probabilistic source database and the other one for the mapping. In this way, the probabilistic source database is independent from the probabilistic mapping. We now define the compact encoding of probabilistic mappings. An *annotated* TGD (resp., NC) has the form $\sigma\colon \lambda$, where σ is a TGD (resp., NC), and λ is an annotation. A set Σ of annotated TGDs and NCs $\sigma\colon \lambda$ with $\sigma \in \Sigma_{st}$ along with a probability $\mu(w) \in [0,1]$ for every world w *compactly encodes a probabilistic mappings* $\mu_{st}\colon 2^{\Sigma_{st}} \to [0,1]$ whenever (i) the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and (ii) the probability μ_{st} of every

[4] That is, we do not consider subsets of the databases here.

subset-maximal $\{\sigma_1, \ldots, \sigma_k\} \subseteq \Sigma_{st}$ such that $\{\sigma_1 : \lambda_1, \ldots, \sigma_k : \lambda_k\} \subseteq \Sigma$ for some annotations $\lambda_1, \ldots, \lambda_k$ is the probability μ of $\lambda_1 \wedge \cdots \wedge \lambda_k$ (and the probability μ_{st} of every other subset of Σ_{st} is 0).

3.4 Computational Problems

We consider the following computational problems:

Existence of a solution (resp., universal solution): Given an ODE or a PODE problem $\mathcal{M}$ and a probabilistic source database Pr_s, decide whether there exists a probabilistic (resp., probabilistic universal) solution for Pr_s relative to $\mathcal{M}$.

Answering UCQs: Given an ODE or a PODE problem $\mathcal{M}$, a probabilistic source database Pr_s, a UCQ $q(\mathbf{X})$, and a tuple $\mathbf{t}$ over $\mathbf{C}$, compute $conf_Q(\mathbf{t})$ in Pr_s w.r.t. $\mathcal{M}$.

4 Computational Complexity

We now analyze the computational complexity of deciding the existence of a (universal) probabilistic solution for deterministic and probabilistic ontological data exchange problems. We also delineate some tractable special cases, and we provide some complexity results for exact UCQ answering for ODE and PODE problems.

We assume some elementary background in complexity theory [15, 20]. We now briefly recall the complexity classes that we encounter in our complexity results. The complexity classes PSPACE (resp., P, EXP, 2EXP) contain all decision problems that can be solved in polynomial space (resp., polynomial, exponential, double exponential time) on a deterministic Turing machine, while the complexity classes NP and NEXP contain all decision problems that can be solved in polynomial and exponential time on a nondeterministic Turing machine, respectively; coNP and coNEXP are their complementary classes, where "Yes" and "No" instances are interchanged. The complexity class AC^0 is the class of all languages that are decidable by uniform families of Boolean circuits of polynomial size and constant depth. The inclusion relationships among the above (decision) complexity classes (all currently believed to be strict) are as follows:

$$\text{AC}^0 \subseteq \text{P} \subseteq \text{NP}, \text{coNP} \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq \text{NEXP}, \text{coNEXP} \subseteq 2\text{EXP}$$

The (function) complexity class $\#\text{P}$ is the set of all functions that are computable by a polynomial-time nondeterministic Turing machine whose output for a given input string I is the number of accepting computations for I.

4.1 Decidability Paradigms

The main (syntactic) conditions on TGDs that guarantee the decidability of CQ answering are guardedness [6], stickiness [8], and acyclicity. Each one of these conditions has its "weak" counterpart: weak guardedness [6], weak stickiness [8], and weak acyclicity [11], respectively.

A TGD σ is *guarded* if there exists an atom in its body that contains (or "guards") all the body variables of σ. The class of guarded TGDs, denoted G, is defined as the

	Data	Comb.	ba-comb.	fp-comb.
L, LF, AF	in AC^0	PSPACE	NP	NP
G	P	2EXP	EXP	NP
WG	EXP	2EXP	EXP	EXP
S, SF	in AC^0	EXP	NP	NP
F, GF	P	EXP	NP	NP
A	in AC^0	NEXP	NEXP	NP
WS, WA	P	2EXP	2EXP	NP

Fig. 3. Complexity of BCQ answering [18]. All entries except for "in AC^0" are completeness ones, where hardness in all cases holds even for ground atomic BCQs.

	Data	Comb.	ba-comb.	fp-comb.
L, LF, AF	coNP	PSPACE	coNP	coNP
G	coNP	2EXP	EXP	coNP
WG	EXP	2EXP	EXP	EXP
S, SF	coNP	EXP	coNP	coNP
F, GF	coNP	EXP	coNP	coNP
A	coNP	coNEXP	coNEXP	coNP
WS, WA	coNP	2EXP	2EXP	coNP

Fig. 4. Complexity of existence of a probabilistic (universal) solution (for both deterministic and probabilistic ODE). All entries are completeness results.

family of all possible sets of guarded TGDs. A key subclass of guarded TGDs are the so-called linear TGDs with just one body atom (which is automatically a guard), and the corresponding class is denoted L. *Weakly guarded* TGDs extend guarded TGDs by requiring only "harmful" body variables to appear in the guard, and the associated class is denoted WG. It is easy to verify that $\mathsf{L} \subset \mathsf{G} \subset \mathsf{WG}$.

Stickiness is inherently different from guardedness, and its central property can be described as follows: variables that appear more than once in a body (i.e., join variables) are always propagated (or "stick") to the inferred atoms. A set of TGDs that enjoys the above property is called *sticky*, and the corresponding class is denoted S. Weak stickiness is a relaxation of stickiness where only "harmful" variables are taken into account. A set of TGDs which enjoys weak stickiness is *weakly sticky*, and the associated class is denoted WS. Observe that $\mathsf{S} \subset \mathsf{WS}$.

A set Σ of TGDs is *acyclic* if its predicate graph is acyclic, and the underlying class is denoted A. In fact, an acyclic set of TGDs can be seen as a nonrecursive set of TGDs. We say Σ is *weakly acyclic* if its dependency graph enjoys a certain acyclicity condition, which actually guarantees the existence of a finite canonical model; the associated class is denoted WA. Clearly, $\mathsf{A} \subset \mathsf{WA}$.

Another key fragment of TGDs, which deserves our attention, are the so-called *full* TGDs, i.e., TGDs without existentially quantified variables, and the corresponding class is denoted F. If we further assume that full TGDs enjoy linearity, guardedness, stickiness, or acyclicity, then we obtain the classes LF, GF, SF, and AF, respectively.

4.2 Overview of Complexity Results

Our complexity results for deciding the existence of a probabilistic (universal) solution for both ODE and PODE problems with annotations over events relative to an underlying Bayesian network are summarized in Fig. 4 for all classes of existential rules discussed above in the data, combined, ba-combined, and fp-combined complexity (all entries are completeness results). For L, LF, AF, S, SF, and A in the data complexity, we obtain tractability when the underlying Bayesian network is a polytree. For all other cases, hardness holds even when the underlying Bayesian network is a polytree. Finally, for all classes of existential rules discussed above except for WG, answering UCQs for both ODE and PODE problems is in $\#\text{P}$ in the data complexity.

4.3 Deterministic Ontological Data Exchange

The first result shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for a probabilistic source database relative to an ODE problem is complete for $\mathcal{C}$ (resp., co$\mathcal{C}$), if BCQ answering for the involved sets of TGDs and NCs is complete for a deterministic (resp., nondeterministic) complexity class $\mathcal{C} \supseteq$ PSPACE (resp., $\mathcal{C} \supseteq$ NP), and hardness holds even for ground atomic BCQs. As a corollary, by the complexity of BCQ answering with TGDs and NCs in Figure 3 [18], we immediately obtain the complexity results shown in Figure 4 for deciding the existence of a probabilistic (universal) solution (in deterministic ontological data exchange) in the combined, ba-combined, and fp-combined complexity, and for the class WG of TGDs and NCs in the data complexity. The hardness results hold even when the underlying Bayesian network is a polytree.

Theorem 1. *Given a probabilistic source database Pr_s relative to a source ontology Σ_s and an ODE problem $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_t \cup \Sigma_{st}$ belongs to a class of TGDs and NCs for which BCQ answering is complete for a deterministic (resp., nondeterministic) complexity class $\mathcal{C} \supseteq$ PSPACE (resp., $\mathcal{C} \supseteq$ NP), and hardness holds even for ground atomic BCQs, deciding the existence of a probabilistic (universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is complete for $\mathcal{C}$ (resp., co$\mathcal{C}$). Hardness holds even when the underlying Bayesian network is a polytree.*

The following result shows that deciding whether there exists a probabilistic (universal) solution for a probabilistic source database relative to an ODE problem is complete for coNP in the data complexity, for all classes of sets of TGDs and NCs considered in this paper, except for WG. Hardness for coNP for the classes G, F, GF, WS, and WA holds even when the underlying Bayesian network is a polytree.

Theorem 2. *Given a probabilistic source database Pr_s relative to a source ontology Σ_s and an ODE problem $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_t \cup \Sigma_{st}$ belongs to a class among* L, LF, AF, G, S, SF, F, GF, A, WS, *and* WA, *deciding whether there exists a probabilistic (or probabilistic universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is* coNP*-complete in the data complexity. Hardness for* coNP *for the classes* G, F, GF, WS, *and* WA *holds even when the underlying Bayesian network is a polytree.*

The following result shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for a probabilistic source database relative to an ODE problem is in P in the data complexity, if BCQ answering for the involved sets of TGDs and NCs is first-order rewritable as a Boolean UCQ, and the underlying Bayesian network is a polytree. As a corollary, by the complexity of BCQ answering with TGDs and NCs, deciding the existence of a solution is in P for the classes L, LF, AF, S, SF, and A in the data complexity, if the underlying Bayesian network is a polytree.

Theorem 3. *Given a probabilistic source database Pr_s relative to a source ontology Σ_s, with a polytree as Bayesian network, and an ODE problem $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_t \cup \Sigma_{st}$ belongs to a class of TGDs and NCs for which BCQ answering is first-order rewritable as a Boolean UCQ, deciding whether there exists a probabilistic (universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is in* P *in the data complexity.*

Finally, the following theorem shows that answering UCQs for probabilistic source databases relative to an ODE problem is complete for #P in the data complexity for all above classes of existential rules except for WG.

Theorem 4. *Given (i) an ODE problem $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_t, \Sigma_s, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class among* L, LF, AF, G, S, SF, F, GF, A, WS, *and* WA, *and (ii) a probabilistic source database Pr_s relative to Σ_s such that there exists a solution for Pr_s relative to $\mathcal{M}$, (iii) a UCQ $Q = q(\boldsymbol{X})$ over $\boldsymbol{T}$, and (iv) a tuple* $\mathbf{a}$*, computing $conf_Q(\mathbf{a})$ is* #P*-complete in the data complexity.*

4.4 Probabilistic Ontological Data Exchange

All the results of Section 4.3 in Theorems 1 and 4 carry over to the case of probabilistic ontological data exchange. Clearly, the hardness results carry over immediately, since deterministic ontological data exchange is a special case of probabilistic ontological data exchange. As for the membership results, we additionally consider the worlds for the probabilistic mapping, which are iterated through in the data complexity and guessed in the combined, the ba-combined, and the fp-combined complexity.

5 Summary and Outlook

We have defined deterministic and probabilistic ontological data exchange problems, where probabilistic knowledge is exchanged between two ontologies. The two ontologies and the mapping between them are defined via existential rules, where the rules for the mapping are deterministic and probabilistic, respectively. We have given a precise analysis of the computational complexity of deciding the existence of a probabilistic (universal) solution for different classes of existential rules in both deterministic and probabilistic ontological data exchange. We also have delineated some tractable special cases, and we have provided some complexity results for exact UCQ answering.

An interesting topic for future research is to further explore the tractable cases of probabilistic solution existence and whether they can be extended, e.g., by slightly generalizing the type of the mapping rules. Another issue for future work is to further analyze the complexity of answering UCQs for different classes of existential rules in deterministic and probabilistic ontological data exchange.

Acknowledgments. This work was supported by an EU (FP7/2007-2013) Marie-Curie Intra-European Fellowship ("PRODIMA"), the UK EPSRC grant EP/J008346/1 ("PrOQAW"), the ERC grant 246858 ("DIADEM"), a Yahoo! Research Fellowship, and funds from Universidad Nacional del Sur and CONICET, Argentina. This paper is a short version of a paper that appeared in Proc. RuleML 2015 [19].

References

1. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V.: Exchanging OWL2 QL knowledge bases. In: Proc. IJCAI. pp. 703–710 (2013)
2. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V., Sherkhonov, E.: Exchanging description logic knowledge bases. In: Proc. KR. pp. 563–567 (2012)

3. Arenas, M., Pérez, J., Reutter, J.L.: Data exchange beyond complete data. J. ACM 60(4), 28:1–28:59 (2013)
4. Baader, F.: Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In: Proc. IJCAI. pp. 364–369 (2003)
5. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. IJCAI. pp. 364–369 (2005)
6. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. J. Artif. Intell. Res. 48, 115–174 (2013)
7. Cali, A., Gottlob, G., Lukasiewicz, T., Marnette, B., Pieris, A.: Datalog+/–: A family of logical knowledge representation and query languages for new applications. In: Proc. LICS. pp. 228–242 (2010)
8. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: The query answering problem. Artif. Intell. 193, 87–128 (2012)
9. Calvanese, D., De Giacomo, G., 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)
10. Fagin, R., Kimelfeld, B., Kolaitis, P.G.: Probabilistic data exchange. J. ACM 58(4), 15:1–15:55 (2011)
11. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
12. Fuhr, N., Rölleke, T.: A probabilistic relational algebra for the integration of information retrieval and database systems. ACM Trans. Inf. Sys. 15(1), 32–66 (1997)
13. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Proc. PODS. pp. 31–40 (2007)
14. Imielinski, T., Witold Lipski, J.: Incomplete information in relational databases. J. ACM 31(4), 761–791 (1984)
15. Johnson, D.S.: A catalog of complexity classes. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. A, chap. 2, pp. 67–161. MIT Press (1990)
16. Krisnadhi, A., Lutz, C.: Data complexity in the $\mathcal{EL}$ family of description logics. In: Proc. LPAR, LNCS, vol. 4790, pp. 333–347. Springer (2007)
17. Lenzerini, M.: Data integration: A theoretical perspective. In: Proc. PODS. pp. 233–246 (2002)
18. Lukasiewicz, T., Martinez, M.V., Pieris, A., Simari, G.I.: From classical to consistent query answering under existential rules. In: Proc. AAAI. pp. 1546–1552 (2015)
19. Lukasiewicz, T., Martinez, M.V., Predoiu, L., Simari, G.I.: Existential rules and Bayesian networks for probabilistic ontological data exchange. In: Proc. RuleML. LNCS, vol. 9202, pp. 294–310. Springer (2015)
20. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1994)
21. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Sem. 10, 133–173 (2008)
22. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. M & C (2011)
23. Vardi, M.Y.: The complexity of relational query languages (extended abstract). In: Proc. STOC. pp. 137–146 (1982)