Basic Probabilistic Ontological Data Exchange with Existential Rules

Thomas Lukasiewicz,[1] **Maria Vanina Martinez,**[2] **Livia Predoiu,**[1,3] **Gerardo I. Simari**[2]
[1]Department of Computer Science, University of Oxford, UK
[2]Institute for Computer Science and Engineering (Universidad Nacional del Sur–CONICET), Bahia Blanca, Argentina
[3]Department of Computer Science, Otto-von-Guericke University, Magdeburg, Germany
{thomas.lukasiewicz, livia.predoiu}@cs.ox.ac.uk, {mvm, gis}@cs.uns.edu.ar

Abstract

We study the complexity of exchanging probabilistic data between ontology-based probabilistic databases. We consider the Datalog+/– family of languages as ontology and ontology mapping languages, and we assume different compact encodings of the probabilities of the probabilistic source databases via Boolean events. We provide an extensive complexity analysis of the problem of deciding the existence of a probabilistic (universal) solution for a given probabilistic source database relative to a (probabilistic) data exchange problem for the different languages considered.

Introduction

Being able to process uncertainty attached to data is becoming increasingly important in many areas such as information extraction, data cleaning, and Web data integration. Applications in these areas produce large volumes of uncertain data. At the moment, the best way to model, store, and process uncertain data is in probabilistic databases (Suciu et al. 2011). Enriching databases with ontological knowledge has recently gained more importance through the requirement of ontology-based data access (OBDA) (Poggi et al. 2008).

A crucial challenge of accessing distributed Web-based knowledge as found in the Semantic Web and created through distributed OBDA applications is to integrate and exchange knowledge. We may have to map complex concepts or queries over one or several ontologies to another one. Apart from the uncertainty attached to source data, a second kind of uncertainty may have to be considered as well, namely, the uncertainty of automatically (or semi-automatically) created ontology mappings.

Data exchange (Fagin et al. 2005) is an important and powerful theoretical framework used for studying data-interoperability tasks that require data to be transferred from existing source databases to a target database that comes with its own (independently created) schema (and schema constraints). The data is translated from one database to another one via schema mappings, which are declarative specifications that describe the relationship between two database schemas. Fagin, Kimelfeld, and Kolaitis (2011) proposed a probabilistic extension of the classical deterministic framework to probabilistic databases and a probabilistic source-to-target mapping. Recently, Lukasiewicz et al. (2015b) have extended this probabilistic data exchange framework towards probabilistic ontological data exchange. In addition to weakly acyclic existential rules, which are the only source-to-target mapping rules and constraints on the target database that have been considered in data exchange so far (see, e.g., (Barcelo 2009)), in (Lukasiewicz et al. 2015b), several other fragments of Datalog+/– are considered as ontology and source-to-target mapping languages.

In this paper, we continue this line of research. We consider a more basic probabilistic model where probabilities of annotations with Boolean events are specified via pairwise independent random variables (rather than Bayesian networks). We study the data and combined complexity of deciding the existence of (universal) probabilistic solutions, obtaining a complete picture of the data complexity and the general, bounded-arity, and fixed-program combined complexity for the main Datalog+/– languages.

Note that annotations with Boolean events are widely used for encoding probabilities in probabilistic logical knowledge representation (Fuhr and Rölleke 1997; Suciu et al. 2011) and are also known as data provenance and lineage (Imielinski and Witold Lipski 1984; Fuhr and Rölleke 1997; Green, Karvounarakis, and Tannen 2007; Suciu et al. 2011). Note also that closely related to exchanging uncertain and/or ontological data as studied in (Fagin, Kimelfeld, and Kolaitis 2011; Lukasiewicz et al. 2015b) is exchanging incomplete databases as proposed in (Arenas, Pérez, and Reutter 2013), which considers incomplete deterministic source and target databases in the data exchange problem and deterministic mappings. Also related is the approach to knowledge base exchange between deterministic *DL-Lite*$_{RDFS}$ and *DL-Lite*$_{\mathcal{R}}$ ontologies in (Arenas et al. 2012; 2013).

The main contributions of this paper are briefly as follows.

- We consider a more basic probabilistic model than the one presented in (Lukasiewicz et al. 2015b). It involves probabilistically independent Boolean random variables representing possible worlds without dependencies. Our complexity analysis of the problem of solution existence reveals a lower data complexity, yielding tractability for nearly all Datalog+/– languages considered.

- We study three different kinds of annotations that provide compact encodings of the probabilities and analyze their impact on the complexity of the problem of deciding the existence of a solution. The most general encoding corresponds to fully expressive Boolean formulas over probabilistic events (Fuhr and Rölleke 1997; Green and Tannen 2006; Fagin, Kimelfeld, and Kolaitis 2011). The other two are PosBool and tuple-independent annotations (Green, Karvounarakis, and Tannen 2007; Suciu et al. 2011). It turns out that they differ only in the data complexity, and using PosBool has the same complexity as tuple-independent annotations, despite being more expressive and also expressive enough to compactly encode probabilistic databases.

Preliminaries

We now recall the basics of Datalog+/– (Calì, Gottlob, and Lukasiewicz 2012; Calì et al. 2010), including especially relational databases, tuple-generating dependencies (TGDs), and (Boolean) conjunctive queries ((B)CQs).

We assume infinite sets of *constants* $\mathbf{C}$, *(labeled) nulls* $\mathbf{N}$, and *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 mapping $h : \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \rightarrow \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ 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 $\forall \mathbf{X} \forall \mathbf{Y}\, \varphi(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \subseteq \mathbf{V}$, $\varphi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Z})$ is an atom. We call $\varphi(\mathbf{X}, \mathbf{Y})$ the *body* of σ, denoted $body(\sigma)$, and $p(\mathbf{X}, \mathbf{Z})$ 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 whenever there exists a homomorphism h such that $h(\varphi(\mathbf{X}, \mathbf{Y})) \subseteq I$, then there exists $h' \supseteq h|_{\mathbf{X} \cup \mathbf{Y}}$, where $h|_{\mathbf{X} \cup \mathbf{Y}}$ is the restriction of h to $\mathbf{X} \cup \mathbf{Y}$, such that $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. 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 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 $\mathbf{t} \in ans(q, D, \Sigma)$. It is well-known that such CQ answering can be reduced in LOGSPACE to BCQ answering, and we thus focus on BCQ answering only. Following Vardi's taxonomy (Vardi 1982), 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* (*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. In the *data complexity*, only the database is part of the input; the *fixed-program combined complexity* (*fp-combined complexity*) is calculated by considering the set of TGDs and NCs as fixed.

Ontological Data Exchange

The source (resp., target) of the ontological data exchange problem that we consider here in this paper is a probabilistic database (resp., probabilistic instance), each relative to a deterministic ontology.

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 $\sum_{I \in \mathcal{I}} \mu(I) = 1$.

The next two sections provide the definitions of *deterministic* and *probabilistic ontological data exchange* (as proposed in (Lukasiewicz et al. 2015b)).

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 a set of TGDs and NCs).

More specifically, an *ontological data exchange (ODE) problem* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ 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 $body(\sigma)$ and $head(\sigma)$ 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 the ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ iff $(I \cup J) \models \Sigma_s \cup \Sigma_t \cup \Sigma_{st}$. We denote by $Sol_{\mathcal{M}}$ the set of all such (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 $\mathcal{M}$ iff (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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ iff 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) $\sum_{J \in \mathcal{J}}(\mu(I,J)) = \mu_s(I)$ for all $I \in \mathcal{I}$ and (i.b) $\sum_{I \in \mathcal{I}}(\mu(I,J)) = \mu_t(J)$ 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 that 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 the relation name denotes its arity), the target schema $\mathbf{T} = \{$*UResearchArea*$/3$, *Lecture*$/2\}$, 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:

$$
\begin{aligned}
&\sigma_s\colon \mathit{Publication}(\mathrm{X},\mathrm{Y},\mathrm{Z}) \to \mathit{ResearchArea}(\mathrm{X},\mathrm{Y}),\\
&\nu_s\colon \mathit{Researcher}(\mathrm{X},\mathrm{Y}) \wedge \mathit{ResearchArea}(\mathrm{X},\mathrm{Y}) \to \bot,\\
&\sigma_t\colon \mathit{UResearchArea}(\mathrm{U},\mathrm{D},\mathrm{T}) \to \exists \mathrm{Z}\, \mathit{Lecture}(\mathrm{T},\mathrm{Z}),\\
&\nu_t\colon \mathit{Lecture}(\mathrm{X},\mathrm{Y}) \wedge \mathit{Lecture}(\mathrm{Y},\mathrm{X}) \to \bot,\\
&\sigma_{st}\colon \mathit{ResearchArea}(\mathrm{N},\mathrm{T}) \wedge\\
&\qquad\qquad \mathit{Researcher}(\mathrm{N},\mathrm{U}) \to \exists \mathrm{D}\, \mathit{UResearchArea}(\mathrm{U},\mathrm{D},\mathrm{T}),\\
&\nu_m\colon \mathit{ResearchArea}(\mathrm{N},\mathrm{T}) \wedge \mathit{UResearchArea}(\mathrm{U},\mathrm{T},\mathrm{N}) \to \bot.
\end{aligned}
$$

Given the probabilistic source database in Table 1, two possible probabilistic solution instances $Pr_{t1} = (\mathcal{J}_1, \mu_{t1})$ and $Pr_{t2} = (\mathcal{J}_2, \mu_{t2})$ are shown in Table 1: Pr_{t1} and Pr_{t2}. Note that while both Pr_{t1} and Pr_{t2} are probabilistic solutions, only Pr_{t1} is also a probabilistic universal solution. ■

Query answering in ontological data exchange is performed over the target ontology and is generalized from deterministic data exchange; see (Lukasiewicz et al. 2015b) for a formal definition.

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 now 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 $Pr_t = (\mathcal{J}, \mu_t)$ is a *probabilistic solution* (resp., *probabilistic universal solution*) for a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$ relative to a PODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$ iff there exists a probability space $Pr = (\mathcal{I} \times \mathcal{J} \times 2^{\Sigma_{st}}, \mu)$ such that: (i) The three marginals of μ are μ_s, μ_t, and μ_{st}, such that: (i.a) $\sum_{J \in \mathcal{J},\, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma') = \mu_s(I)$ for all $I \in \mathcal{I}$, (i.b) $\sum_{I \in \mathcal{I},\, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma') = \mu_t(J)$ for all $J \in \mathcal{J}$, and (i.c) $\sum_{I \in \mathcal{I},\, J \in \mathcal{J}} \mu(I, J, \Sigma') = \mu_{st}(\Sigma')$ for all $\Sigma' \subseteq \Sigma_{st}$; (ii) $\mu(I, J, \Sigma') = 0$ for all $(I, J) \notin Sol_{(\mathbf{S},\mathbf{T},\Sigma')}$ (resp., $(I,J) \notin USol_{(\mathbf{S},\mathbf{T},\Sigma')}$).

Using probabilistic (universal) solutions for probabilistic source databases relative to PODE problems, the semantics of UCQs can be lifted to PODE problems; cf. (Lukasiewicz et al. 2015b) for a formal definition.

Compact Encoding

We use a compact encoding of both probabilistic databases and probabilistic mappings, which is based on annotating database atoms, TGDs, and NCs by probabilistic Boolean events rather than explicitly specifying the whole probability space. That is, database atoms, TGDs, and NCs are annotated with Boolean combinations of elementary events, where every annotation describes when the annotated item is true and is associated with a probability. We first define general annotations and general annotated atoms.

Definition 1 (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 are annotations, then also $\neg\lambda_1$ and $\lambda_1 \wedge \lambda_2$); as usual, $\lambda_1 \vee \lambda_2$ abbreviates $\neg(\neg\lambda_1 \wedge \neg\lambda_2)$. 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 is then defined as follows. Note that this encoding is also underlying our complexity analysis below.

Definition 2 (Compact Encoding of Probabilistic Databases). 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:

(1) the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and

(2) the probability μ of every subset-maximal database $\{a_1, \ldots, a_m\} \in \mathcal{I}$ 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

	Possible source database facts
r_a	*Researcher*(Alice, UniversityOfOxford)
r_p	*Researcher*(Paul, UniversityOfOxford)
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 $Pr_s = (I, \mu_s)$	
$I_1 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{aml}, \mathrm{p}_{pdb}, \mathrm{a}_{aml}, \mathrm{a}_{pdb}\}$	0.3
$I_2 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{aml}, \mathrm{p}_{pai}, \mathrm{a}_{aml}, \mathrm{a}_{pai}\}$	0.3
$I_3 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{adb}, \mathrm{p}_{pai}, \mathrm{a}_{adb}, \mathrm{a}_{pai}\}$	0.2
$I_4 = \{\mathrm{r}_a, \mathrm{r}_p, \mathrm{p}_{adb}, \mathrm{p}_{pdb}, \mathrm{a}_{adb}, \mathrm{a}_{pdb}\}$	0.1
$I_5 = \{\mathrm{r}_a, \mathrm{p}_{adb}, \mathrm{a}_{adb}\}$	0.1

	Possible target instance facts
u_{ml}	*UResearchArea*(UniversityOfOxford, N_1, ML)
u_{ai}	*UResearchArea*(UniversityOfOxford, N_2, AI)
u_{db}	*UResearchArea*(UniversityOfOxford, N_3, DB)
l_{ml}	*Lecture*(ML, N_4)
l_{ai}	*Lecture*(AI, N_5)
l_{db}	*Lecture*(DB, N_6)

Probabilistic target instance $Pr_{t1} = (J_1, \mu_{t1})$	
$J_1 = \{\mathrm{u}_{ml}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{db}\}$	0.3
$J_2 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{l}_{ml}, \mathrm{l}_{ai}\}$	0.3
$J_3 = \{\mathrm{u}_{ai}, \mathrm{u}_{db}, \mathrm{l}_{ai}, \mathrm{l}_{db}\}$	0.2
$J_4 = \{\mathrm{u}_{db}, \mathrm{l}_{db}\}$	0.2

Probabilistic target instance $Pr_{t2} = (J_2, \mu_{t2})$	
$J_5 = \{\mathrm{u}_{ml}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{db}\}$	0.35
$J_6 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{l}_{ml}, \mathrm{l}_{ai}\}$	0.2
$J_7 = \{\mathrm{u}_{ml}, \mathrm{u}_{ai}, \mathrm{u}_{db}, \mathrm{l}_{ml}, \mathrm{l}_{ai}, \mathrm{l}_{db}\}$	0.45

Table 1: Probabilistic source database and two probabilistic target instances for Example 1 ($N_1, \ldots, N_5$ are nulls); both are probabilistic solutions, but only *Pr$_{t1}$* is universal.

of $\lambda_1 \wedge \cdots \wedge \lambda_m$ (and the probability μ of every other database in $\mathcal{I}$ is 0).

We assume that all annotations are in disjunctive normal form (DNF), i.e., disjunctions of conjunctions of literals, and we consider the following three cases:

Elementary-event-independence: elementary events and their negations are pairwise probabilistically independent (i.e., the probability of worlds $\ell_1 \wedge \cdots \wedge \ell_n$ of elementary events ($\ell_i = e_i$) and their negations ($\ell_i = \neg e_i$) is defined as $\Pi_{i=1}^{n} \nu(\ell_i)$, where $\nu(\ell_i) = \mu(e_i)$ and $\nu(\ell_i) = 1 - \mu(e_i)$, respectively);

PosBool: a special case of elementary-event-independence where all annotations are arbitrary many disjunctions of arbitrary many conjunctions of positive elementary events. Again, elementary events are pairwise probabilistically independent (i.e., the probability of worlds $\ell_1 \wedge \cdots \wedge \ell_n$ of elementary events ($\ell_i = e_i$) is defined as $\Pi_{i=1}^{n} \nu(\ell_i)$, where $\nu(\ell_i) = \mu(e_i)$);

Tuple-independence: special case of PosBool where annotations are elementary and worlds have positive probability.

Note that in the tuple-independent case, annotations consist of as many elementary events as database atoms, and each database atom is annotated with a different single elementary event. Below in Example 2, we provide an example of an annotation encoding of a probabilistic database.

Example 2. In Table 2, an annotation encoding of a probabilistic source database is shown. It has four elementary events e_1, e_2, e_3, and e_4 along with their probabilities $p(e_1) = 3/10$, $p(e_2) = 3/7$, $p(e_3) = 1/2$, and $p(e_4) = 1/2$, respectively. The encoding compactly represents the probabilistic source database in Table 1. ■

	Possible source database facts	Annotation
r_a	*Researcher*(Alice, UniversityOfOxford)	true
r_p	*Researcher*(Paul, UniversityOfOxford)	$\mathrm{e}_1 \vee \mathrm{e}_2 \vee \mathrm{e}_3 \vee \mathrm{e}_4$
p_{aml}	*Publication*(Alice, ML, JMLR)	$\mathrm{e}_1 \vee \mathrm{e}_2$
p_{adb}	*Publication*(Alice, DB, TODS)	$\neg\, \mathrm{e}_1 \wedge \neg\, \mathrm{e}_2$
p_{pdb}	*Publication*(Paul, DB, TODS)	$\mathrm{e}_1 \vee (\neg\, \mathrm{e}_2 \wedge \neg\, \mathrm{e}_3 \wedge \mathrm{e}_4)$
p_{pai}	*Publication*(Paul, AI, AIJ)	$(\neg\, \mathrm{e}_1 \wedge \mathrm{e}_2) \vee (\neg\, \mathrm{e}_1 \wedge \mathrm{e}_3)$

Table 2: Annotation encoding of the prob. source DB in Table 1.

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.

Definition 3 (**Compact Encoding of Prob. 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 mapping* $\mu_{st}\colon 2^{\Sigma_{st}} \to [0, 1]$ whenever:

(1) the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and

(2) the probability μ_{st} of every subset-maximal $\{\sigma_1, \ldots, \sigma_k\} \subseteq \Sigma_{st}$ such that $\{\sigma_1\colon \lambda_1, \ \ldots, \sigma_k\colon \lambda_k\} \subseteq \Sigma$ 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).

We consider the following computational problem.

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

W.l.o.g., we consider ontologies in the same language because the more expressive one always defines the complexity class of the exchange problem as a whole.

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.

Complexity Classes

We assume some elementary background in complexity theory; see (Johnson 1990; Papadimitriou 1994). We now briefly recall the complexity classes that we encounter in our complexity results below. The complexity class PSPACE (resp., P, EXP, 2EXP) contains 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, and 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 above complexity classes and their inclusion relationships (which are all currently believed to be strict) are shown below:

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

Decidability Paradigms

The main (syntactic) conditions on TGDs that guarantee the decidability of CQ answering are guardedness (Calì, Gottlob, and Kifer 2013), stickiness (Calì, Gottlob, and Pieris 2012), and acyclicity. Each one of these conditions has its "weak" counterpart: weak guardedness (Calì, Gottlob, and Kifer 2013), weak stickiness (Calì, Gottlob, and Pieris 2012), and weak acyclicity (Fagin et al. 2005), 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 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 L $\subset$ G $\subset$ 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 S $\subset$ 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, A $\subset$ 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.

	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

Table 3: Complexity of BCQ answering (Lukasiewicz et al. 2015a). All entries except for "in AC^0" are completeness; hardness holds in all cases even for ground atomic BCQs.

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

Table 4: Complexity of existence of a (universal) probabilistic solution in the tuple-independent case and for posBool annotations (for both ODE and PODE problems). All entries except for "in AC^0" are completeness results.

Overview of Complexity Results

Our complexity results for deciding the existence of a (universal) probabilistic solution in the tuple-independent and the elementary-event-independent case for both ODE and PODE problems are summarized in Tables 4 and 5, respectively, which show the data complexity as well as the combined, the *ba*-combined, and the *fp*-combined complexity for different classes of existential rules, ranging from the classes LF, AF, and SF to the more general classes WG, WA, and WS. Note that the last column of Tables 4 and 5 is the same as the first column because data and *fp*-combined complexity differ only in the query being not fixed in the later case. As we only consider the existence of solutions and, hence, consistency here, we do not have any query as input.

Deterministic Ontological Data Exchange

We first focus on the tuple-independent case. The following result shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for a tuple-independent probabilistic source database relative to a source ontology and a deterministic mapping is $\mathcal{C}$-complete for a complexity class $\mathcal{C} \supseteq$ P (resp., in AC^0), if BCQ answering for the involved sets of TGDs and NCs has this complexity and is hard even for ground atomic BCQs. As a corollary, by the complexity of BCQ answering with TGDs

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

Table 5: Complexity of existence of a (universal) probabilistic solution in the elementary-event-independent case (for both ODE and PODE problems). All entries except "in P" are completeness results.

and NCs in Table 3 (Lukasiewicz et al. 2015a), we obtain the complexity results shown in Table 4 for deciding the existence of a (universal) probabilistic solution (in deterministic ontological data exchange) in the tuple-independent case.

Theorem 4. *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 $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is complete for a complexity class $\mathcal{C} \supseteq$ P (resp., in AC^0), and hardness holds even for ground atomic BCQs, deciding whether there exists a probabilistic (or probabilistic universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is co$\mathcal{C}$-complete (resp., in AC^0) in the tuple-independent case.*

The next theorem considers PosBool (Green, Karvounarakis, and Tannen 2007) annotations allowing only Boolean connectives $\wedge$ and $\vee$. Note that the tuple-independent case is a special case of PosBool (only $\wedge$) and both are special cases of general Boolean formulas (allowing negation as well) as annotations.

Theorem 5. *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 $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is complete for a complexity class $\mathcal{C} \supseteq$ P (resp., in AC^0), and hardness holds even for ground atomic BCQs, deciding whether there exists a probabilistic (or probabilistic universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is co$\mathcal{C}$-complete (resp., in AC^0) with annotations in PosBool.*

The next theorem shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for an elementary-event-independent probabilistic source database relative to a source ontology and a deterministic mapping 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 Table 3 (Lukasiewicz et al. 2015a), we obtain the complexity results shown in Table 4 for deciding the existence of a (universal) probabilistic 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, in the elementary-event-independent case.

Theorem 6. *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 $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ 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 whether there exists a probabilistic (or probabilistic universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is complete for $\mathcal{C}$ (resp., co$\mathcal{C}$) in the elementary-event-independent case.*

The following result shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for an elementary-event-independent probabilistic source database relative to a source ontology and a deterministic mapping is complete for coNP, if the involved sets of TGDs and NCs belong to a class among G, F, GF, WS, and WA.

Theorem 7. *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 $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class among* G, F, GF, 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 elementary-event-independent case in the data complexity.*

The following result shows that deciding whether there exists a probabilistic (or probabilistic universal) solution for an elementary-event-independent probabilistic source database relative to a source ontology and a deterministic mapping is in P, if BCQ answering for the involved sets of TGDs and NCs is first-order rewritable as a Boolean UCQ. As a corollary, by the complexity of BCQ answering with TGDs and NCs, we obtain the complexity results in Table 5 for the classes L, LF, AF, S, SF, and A in the data complexity for deciding the existence of a (universal) probabilistic solution (in deterministic ontological data exchange) in the elementary-event-independent case.

Theorem 8. *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 $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ 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 (or probabilistic universal) solution for Pr_s relative to Σ_s and $\mathcal{M}$ is in* P *in the elementary-event-independent case in the data complexity.*

Probabilistic Data Exchange

All the results in Theorems 4, 5, 6, 7, and 8 carry over to PODE problems. Clearly, the hardness results carry over immediately, as deterministic ontological data exchange is a special case of probabilistic ontological data exchange. As for the membership results, in the tuple-independent case, rather than looking only at the maximal database, we also include the maximal set of TGDs and NCs from the probabilistic mapping, while in the elementary-event-independent case, we also consider the worlds for the probabilistic mapping, which are iterated through in the data complexity and guessed in the combined and *ba*/*fp*-combined complexity.

Summary and Outlook

We have studied the impact of a more basic probabilistic model to the problem of solution existence in ontological data exchange as defined in (Lukasiewicz et al. 2015b). We have also considered three different kinds of compact encodings for probabilistic atoms, TGDs, and NCs. In particular, we have given a precise analysis of the computational complexity of deciding the existence of a (universal) probabilistic solution for different classes of existential rules in both deterministic and probabilistic ontological data exchange and under elementary-event-independent, posBool, and tuple-independent annotations. The data complexity has turned out to be tractable in all cases but one: we have shown tractability via many FO-rewritable and polynomial cases, both in the data and the fixed-program combined complexity, thus yielding a more application-oriented framework—note that the work in (Lukasiewicz et al. 2015b) did not identify any FO-rewritable or polynomial cases.

Interesting topics for future research are further explorations of the tractable cases of probabilistic solution existence as well as extensions, e.g., by generalizing the type of the mapping rules. Another issue for future work is to analyze the complexity of answering UCQs and to also consider languages combining (chase-style) forward and backward chaining (Baget et al. 2009), as well as additional frameworks, such as (weakly) frontier-guarded sets of rules (Baget, Leclère, and Mugnier 2010) and approaches considering acyclicity notions, as, e.g., proposed in (Krötzsch and Rudolph 2011; Grau et al. 2013; Baget et al. 2014).

Acknowledgments. This work was supported by the EU FP7/2007-2013 Marie-Curie IE Fellowship PRODIMA, the UK EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1, and funds from Universidad Nacional del Sur and CONICET, Argentina.

References

Arenas, M.; Botoeva, E.; Calvanese, D.; Ryzhikov, V.; and Sherkhonov, E. 2012. Exchanging description logic knowledge bases. In *Proc. of KR*, 563–567.

Arenas, M.; Botoeva, E.; Calvanese, D.; and Ryzhikov, V. 2013. Exchanging OWL2 QL knowledge bases. In *Proc. of IJCAI*, 703–710.

Arenas, M.; Pérez, J.; and Reutter, J. L. 2013. Data exchange beyond complete data. *J. ACM* 60(4):28:1–28:59.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2009. Extending decidable cases for rules with existential variables. In *Proc. of IJCAI*, 677–682.

Baget, J.-F.; Garreau, F.; Mugnier, M.-L.; and Rocher, S. 2014. Extending acyclicity notions for existential rules. In *Proc. of ECAI*, 39–44.

Baget, J.-F.; Leclère, M.; and Mugnier, M.-L. 2010. Walking the decidability line for rules with existential variables. In *Proc. of KR*, 466–476.

Barcelo, P. 2009. Logical foundations of relational data exchange. *SIGMOD Record* 38(1):49–58.

Calì, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; and Pieris, A. 2010. Datalog+/–: A family of logical knowledge representation and query languages for new applications. In *Proc. of LICS*, 228–242.

Calì, A.; Gottlob, G.; and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.* 48:115–174.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14:57–83.

Calì, A.; Gottlob, G.; and Pieris, A. 2012. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.* 193:87–128.

Fagin, R.; Kolaitis, P. G.; Miller, R. J.; and Popa, L. 2005. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* 336(1):89–124.

Fagin, R.; Kimelfeld, B.; and Kolaitis, P. G. 2011. Probabilistic data exchange. *J. ACM* 58(4):15:1–15:55.

Fuhr, N., and Rölleke, T. 1997. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Sys.* 15(1):32–66.

Grau, B. C.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity notions for existential rules and their application to query answering in ontologies. *J. Artif. Intell. Res.* 47:741–808.

Green, T. J.; and Tannen, V. 2006. Models for incomplete and probabilistic information. *IEEE Data Eng. Bull.* 29:17–24.

Green, T. J.; Karvounarakis, G.; Tannen, V. 2007. Provenance semirings. In *Proc. of PODS*, 31–40.

Imielinski, T., and Witold Lipski, J. 1984. Incomplete information in relational databases. *J. ACM* 31(4):761–791.

Johnson, D. S. 1990. A catalog of complexity classes. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science*, volume A. MIT Press. chapter 2, 67–161.

Krötzsch, M., and Rudolph, S. 2011. Extending decidable existential rules by joining acyclicity and guardedness. In *Proc. of IJCAI*, 963–968.

Lukasiewicz, T.; Martinez, M. V.; Pieris, A.; and Simari, G. I. 2015a. From classical to consistent query answering under existential rules. In *Proc. of AAAI*, 1546–1552.

Lukasiewicz, T.; Martinez, M. V.; Predoiu, L.; and Simari, G. I. 2015b. Existential rules and Bayesian networks for probabilistic ontological data exchange. In *Proc. of RuleML*, 294–310.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

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

Suciu, D.; Olteanu, D.; Ré, C.; and Koch, C. 2011. *Probabilistic Databases*. Morgan & Claypool.

Vardi, M. Y. 1982. The complexity of relational query languages (extended abstract). In *Proc. of STOC*, 137–146.