ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1008

Complexity of Threshold Query Answering in Probabilistic Ontological Data Exchange

Thomas Lukasiewicz[1] **Livia Predoiu**[2]

Abstract. We study the complexity of threshold query answering in the logical framework for probabilistic ontological data exchange, which is an extension of the classical probabilistic data exchange framework with (1) probabilistic databases compactly encoded with several different annotations according to three different probability models used and (2) existential rules of different expressiveness. The ontological data exchange framework provides a logical formalization of exchanging probabilistic data and knowledge from one ontology to another via either deterministic or probabilistic mappings. We define the threshold query answering task in this framework and provide a thorough analysis of its computational complexity for different classes of existential rules and types of complexity. We also delineate several classes of existential rules and a probability model along with a compact encoding in which the threshold query answering problem can be solved in polynomial time in the data complexity.

1 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 can be considered to be in probabilistic databases [27]. At the same time, the field of databases enriched with ontological knowledge has gained importance through ontology-based data access (OBDA) [26]. Crucial challenges of such ontologically enhanced databases are the integration and the exchange of data and knowledge. There is currently also a huge need for combining the latter two areas in ontology-based probabilistic databases, especially due to important applications in the Semantic Web and in ontology-based access to Big Data.

In this paper, we tackle these challenges by studying an extension of the well-known framework of data exchange [14], which is an important and powerful theoretical framework used for studying data-interoperability tasks that require data to be transferred from source databases to a target database that comes with its own (independently created) ontological schema (and schema constraints). The data is translated from one database to the other via schema mappings, which are declarative specifications that describe the relationship between two database schemas. In [13], a probabilistic extension of the classical deterministic framework of data exchange has been proposed. Recently, the works [23, 24] have extended this probabilistic data exchange framework towards probabilistic ontological data exchange where source and target ontology-based data access systems have been considered for data exchange. While research in (deterministic and probabilistic) data exchange has only considered weakly acyclic existential rules (see, e.g., [6]), in [23, 24], several other classes of existential rules from Datalog+/– have been considered as ontology and source-to-target mapping languages. In addition, compared to [13], where only the elementary-event-independent probabilistic model has been considered together with full Boolean formulas as annotations, in [23, 24], an additional probabilistic model based on Bayesian networks, as well as annotations consisting of positive Boolean formulas without negation, and annotations consisting of a single positive literal have been considered.

This paper continues this line of research. We consider answering threshold queries in probabilistic ontological data exchange with different classes of existential rules, representing both ontological rules and mapping rules, as well as different probabilistic models and compact encodings for the probabilistic ontological data and mappings. Probabilities of annotations with Boolean events are specified via either pairwise independent random variables or with Bayesian networks. We study the data and combined complexity of threshold query answering, obtaining a detailed picture of the data complexity and the general, bounded-arity, and fixed-program combined complexity for the main classes of existential rules from Datalog+/– along with the considered probability models.

Note that annotations with Boolean events are widely used for encoding probabilities in probabilistic logical knowledge representation [15, 27] and are also known as data provenance and lineage [19, 15, 18, 27]. Note also that closely related to ontological data as studied in [23, 24] is exchanging incomplete databases as proposed in [3], 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 [2, 1].

The main contributions of this paper are briefly as follows.

- We define the problem of threshold query answering in the (probabilistic) ontological data exchange framework and study its data complexity, fixed-program-combined complexity, bounded-arity-combined complexity, and general combined complexity. For the complexity analysis, we consider the following main Datalog+/– languages: acyclic (full), weakly acyclic, linear (full), full, guarded (full), weakly guarded, sticky (full), and weakly sticky existential rules together with negative constraints.
- Besides considering different Datalog+/– languages, the complexity analysis for threshold query answering also investigates the impact of different probabilistic models on the complexity. More specifically, probabilities of annotations with Boolean events are specified via either pairwise independent random variables or

[1] University of Oxford, UK, email: thomas.lukasiewicz@cs.ox.ac.uk
[2] University of Oxford, UK, email: livia.predoiu@cs.ox.ac.uk; Otto-von-Guericke University, Magdeburg, Germany

with Bayesian networks. Furthermore, we investigate the impact of compact encodings for pairwise independent random variables: compact encodings via fully expressive Boolean formulas (elementary-event-independent) and via positive Boolean formulas without negation (PosBool and tuple-independence).

- We obtain a complete picture of the complexity of threshold query answering for the elementary-event-independent and the Bayesian-network encoding of probabilistic models. In particular, even in the data complexity, all considered Datalog+/- languages but weakly guarded existential rules (which have EXP-complete data complexity) for ontologies and mappings are PP-complete. In the fixed-program- and the bounded-arity-combined complexity, we obtain a complexity of PP^{NP} for Datalog+/- languages for which Boolean conjunctive query answering is NP-complete.
- For tuple-independent probabilistic databases and databases annotated with positive Boolean formulas, we obtain the same upper bounds as for the elementary-event-independent and the Bayesian-network encoding, and in many cases also the same lower bounds. In addition, we delineate the first-order rewritable classes of existential rules as an interesting case, where we obtain special-case tractability in the tuple-independent case in the data complexity, and where we conjecture a dichotomy of threshold query answering of either polynomial time or PP-hard queries.

The rest of this paper is organized as follows. In Section 2, we provide the preliminaries on Datalog+/- and the main terminology. In Section 3, we describe the ontological data exchange framework with probabilistic databases and with both deterministic mappings (Section 3.1) and probabilistic mappings (Section 3.2); we also present the probabilistic models and the compact encodings that we consider (Section 3.3). In Section 4, we define the threshold query entailment problem and provide its complexity analysis. Finally, in Section 5, we conclude with a summary and an outlook to future work.

2 PRELIMINARIES

We now recall the basics of Datalog+/– [8, 9], including especially relational databases, tuple-generating dependencies (TGDs, or existential rules), 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 $h : \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \to \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ 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*, or *existential rule*) σ is a first-order formula $\forall \mathbf{X} \forall \mathbf{Y}\, \varphi(\mathbf{X}, \mathbf{Y}) \to \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z})$, 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 $head(\sigma)$. A *copy TGD* is of the form $\forall \mathbf{X}\, p(\mathbf{X}) \to p(\mathbf{X})$, where $p(\mathbf{X})$ is an atom with the variables $\mathbf{X} \subseteq \mathbf{V}$ as pairwise different arguments. 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 $\forall \mathbf{X}\, \varphi(\mathbf{X}) \to \bot$, 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 $\mathbf{t} \in ans(q, D, \Sigma)$. 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 [29], 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.

3 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 $\mu \colon \mathcal{I} \to [0, 1]$ is a function that satisfies (i) $\mu(I) > 0$ for only finitely many $I \in \mathcal{I}$ and (ii) $\sum_{I \in \mathcal{I}} \mu(I) = 1$.

We next provide the definitions of *deterministic* and *probabilistic ontological data exchange* (as proposed in [23, 24]).

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 TGDs and NCs) over a schema $\mathbf{T}$ via a (source-to-target) mapping (also 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. 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.

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., $(I, J) \notin USol_{\mathcal{M}}$). Intuitively, this says that all non-solutions (I, J) have probability zero, and that even if a solution exists, there still may be some zero-probability source database(s) without corresponding target instance(s).

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 $\mathbf{S} = \{\mathit{Researcher}/2, \mathit{ResearchArea}/2, \mathit{Publication}/3\}$ (the number after the relation name denotes its arity), the target schema $\mathbf{T} = \{\mathit{UResearchArea}/3, \mathit{Lecture}/2\}$, 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:

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

Given the probabilistic source database in Table 1, two possible probabilistic solution instances are shown in Table 1: $Pr_{t_1} = (\mathcal{J}_1, \mu_{t_1})$ and $Pr_{t_2} = (\mathcal{J}_2, \mu_{t_2})$. While both Pr_{t_1} and Pr_{t_2} are probabilistic solutions, only Pr_{t_1} is also a probabilistic universal solution. ∎

For a deterministic source database D relative to an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ and a CQ $q(\mathbf{X}) = \exists \mathbf{Y}\, \phi(\mathbf{X}, \mathbf{Y})$ over $\mathbf{T}$, the set of *answers* for q to D relative to $\mathcal{M}$ is defined as $ans(q, D, \Sigma_s \cup \Sigma_t \cup \Sigma_{st})$. We now generalize this to probabilistic source databases relative to ODE problems and unions of CQs (UCQs).

A *union of CQs* (or *UCQ*) has the form $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i\, \phi_i(\mathbf{X}, \mathbf{Y}_i)$, where each $\exists \mathbf{Y}_i\, \phi_i(\mathbf{X}, \mathbf{Y}_i)$ with $i \in \{1, \ldots, k\}$ is a CQ with exactly the variables $\mathbf{X}$ and $\mathbf{Y}_i$. 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 $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i\, \phi_i(\mathbf{X}, \mathbf{Y}_i)$, 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 $\exists \mathbf{Y}_i\, \phi_i(\mathbf{t}, \mathbf{Y}_i)$ with $i \in \{1, \ldots, k\}$ evaluates to true in J).

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 now have a probability distribution over the set of all subsets of Σ_{st}.

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 $\mathbf{S}$ (*source ontology*), (iv) a finite set Σ_t of TGDs and NCs over $\mathbf{T}$ (*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 $\sum_{\Sigma' \subseteq \Sigma_{st}} \mu_{st}(\Sigma') = 1$ (*probabilistic (source-to-target) mapping*).

The notion of a probabilistic (universal) solution is defined as follows. 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 $\Sigma' \subseteq \Sigma_{st}$; and (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 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 $Pr_s = (\mathcal{I}, \mu_s)$, a UCQ $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i\, \phi_i(\mathbf{X}, \mathbf{Y}_i)$, 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 $conf_q(\mathbf{t})$, 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 $J \in \mathcal{J}$.

3.3 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.

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.

Possible source database facts

r_a	*Researcher*(Alice, UnivOfOxford)
r_p	*Researcher*(Paul, UnivOfOxford)
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 $Pr_s = (\mathcal{I}, \mu_s)$

$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, a_{aml}, a_{pdb}\}$	0.3
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, a_{aml}, a_{pai}\}$	0.3
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, a_{adb}, a_{pai}\}$	0.2
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, a_{adb}, a_{pdb}\}$	0.1
$I_5 = \{r_a, p_{adb}, a_{adb}\}$	0.1

Possible target instance facts

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

Prob. target instance $Pr_{t_1} = (\mathcal{J}_1, \mu_{t_1})$

$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.3
$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.3
$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.2
$J_4 = \{u_{db}, l_{db}\}$	0.2

Prob. target instance $Pr_{t_2} = (\mathcal{J}_2, \mu_{t_2})$

$J_5 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.35
$J_6 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
$J_7 = \{u_{ml}, u_{ai}, u_{db}, l_{ml}, l_{ai}, l_{db}\}$	0.45

Table 1. Probabilistic source database Pr_s and two probabilistic target instances Pr_{t_1} and Pr_{t_2} (with nulls N_i) for Example 1.

The compact encoding of probabilistic databases is then defined as follows. This encoding is also underlying our complexity analysis below. 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 $\{a_1 : \lambda_1, \ldots, a_m : \lambda_m\} \subseteq \mathbf{A}$ 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 all annotations are in disjunctive normal form (DNF), i.e., disjunctions of conjunctions of literals, and we consider the following four 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 $\nu(\ell_i) = \mu(e_i)$ for $\ell_i = e_i$, and $\nu(\ell_i) = 1 - \mu(e_i)$ for $\ell_i = \neg e_i$);

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.

Elementary-event-dependence encoded by a Bayesian network: Here, we assume that the probability distributions for the underlying elementary events are given by a Bayesian network.

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. The following example illustrates the 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. ∎

If also the mapping is probabilistic, 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 $\sigma : \lambda$, where σ is a TGD (resp., NC), and λ is an annotation. A set Σ of annotated TGDs and NCs $\sigma : \lambda$ with $\sigma \in \Sigma_{st}$ along with a probability $\mu(w) \in [0,1]$ for every world w *compactly encodes a probabilistic mapping* $\mu_{st} : 2^{\Sigma_{st}} \to [0,1]$ 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 $\{\sigma_1 : \lambda_1, \ \ldots, \sigma_k : \lambda_k\} \subseteq \Sigma$ 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).

4 COMPUTATIONAL COMPLEXITY

We now analyze the computational complexity of deciding threshold query answering for deterministic and probabilistic ontological data exchange problems. We also delineate some tractable special cases.

More precisely, we consider the following decision problem. We query the target ontology and ask whether a Boolean UCQ (BUCQ) is entailed with a probability of at least a given threshold $\tau \in [0,1]$.

Definition 3 (Threshold Query Answering) Given a (P)ODE problem $\mathcal{M}$, a probabilistic source database Pr_s, a UCQ $q(\mathbf{X})$, a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and a threshold $\tau \in (0,1]$, decide whether $conf_q(\mathbf{a}) \geq \tau$ in Pr_s relative to $\mathcal{M}$; we then say that $\mathbf{a}$ is a τ-*threshold answer* to $q(\mathbf{X})$. ∎

W.l.o.g., the underlying ontologies and the mapping are in the same language, as the more expressive one always defines the complexity class of the ontological data exchange problem as a whole.

	Possible source database facts	Annotation
r_a	*Researcher*(Alice, UnivOfOxford)	true
r_p	*Researcher*(Paul, UnivOfOxford)	$e_1 \vee e_2 \vee e_3 \vee e_4$
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)	$(\neg e_1 \wedge e_2) \vee (\neg e_1 \wedge e_3)$

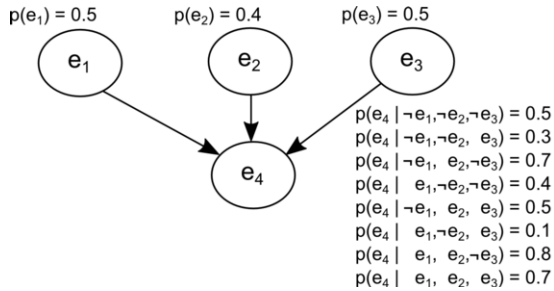

Table 2. Left: Annotation encoding of the probabilistic source database in Table 1. A possible elementary-event-independent interpretation is presented in Example 2. **Right:** An elementary-event-dependent interpretation represented by a polytree Bayesian network.

4.1 Complexity Classes

We assume some elementary background in complexity theory; see [20, 25]. 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. Finally, the complexity class PP (resp., PP^{NP}) contains the problems decidable by a polynomial-time Turing machine (resp., polynomial-time Turing machine with an oracle for NP) that accepts an input iff the majority of its runs halt in an accepting state. The functional analog of PP is the well-known class #P, which is the class of all functions f (from strings to the nonnegative integers) for which there exists a nondeterministic polynomial-time Turing machine T such that for every input string w, it holds that $f(w)$ is the number of accepting runs of T on w. The above (decision) complexity classes and their inclusion relationships (which are all currently believed to be strict) are shown below:

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

4.2 Decidability Paradigms

The main (syntactic) conditions on TGDs that guarantee the decidability of CQ answering are guardedness [7], stickiness [10], and acyclicity. Each of them has its "weak" counterpart: weak guardedness [7], weak stickiness [10], and weak acyclicity [14], 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. More specifically, weakly guardedness requires guards to cover all variables occurring in affected positions only, where affected positions are positions in predicates that may contain some fresh labeled nulls that are generated during the construction of the chase [7]. It is easy to verify that $\mathsf{L} \subset \mathsf{G} \subset \mathsf{WG}$. Note that guardedness and weak guardedness are generalized by the notions of frontier-guardedness and weak frontier-guardedness, respectively [4]. More precisely, *frontier-guardedness* relaxes the guardedness condition of TGDs by requiring the guard atom to contain only the frontier of a TGD, which is the set of all variables that appear in both the body and the head of the TGD. Generalizing frontier-guardedness, a set of TGDs is *weakly frontier-guarded*, if each TGD has an atom in its body that contains all affected variables from the frontier of the TGD.

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 its class is denoted S. Weak stickiness is a relaxation of stickiness where only "harmful" variables are taken into account. A set of TGDs that 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. Note that $\mathsf{A} \subset \mathsf{WA} \subset \mathsf{WS}$.

Another key fragment of TGDs are *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. Note that $\mathsf{F} \subset \mathsf{WA}$ and $\mathsf{F} \subset \mathsf{WG}$.

4.3 Overview of Complexity Results

Our complexity results for deciding threshold query answering in the elementary-event-independent and the Bayesian-network case for both ODE and PODE problems are summarized in Table 4, while our complexity results for deciding threshold query answering in the tuple-independent and the PosBool case are summarized in Table 5.

More precisely, in the elementary-event-independent and the Bayesian-network case (see Table 4), threshold query answering is complete for PP (resp., PP^{NP}) in the data (resp., *fp*-combined) complexity for all fragments of existential rules, except for $\mathsf{WG}_\perp$, where it is complete for EXP. The *ba*-combined complexity in the elementary-event-independent and the Bayesian-network case is among PP^{NP} (for L, LF, AF, S, SF, F, and GF), EXP (for G and WG), NEXP (for A), and 2EXP (for WS and WA), while the combined complexity is among PSPACE (for L, LF, and AF), EXP (for S, SF, F, and GF), NEXP (for A), and 2EXP (for G, WG, WS, and WA).

Note that the same complexity results hold for other fragments of TGDs where standard BCQ answering has the same complexity, e.g., since standard BCQ answering in the frontier-guarded fragment is complete for P, 2EXP, and 2EXP (EXP, 2EXP, and 2EXP in the weakly frontier-guarded case) in the data, *ba*-combined, and combined complexity, respectively [4], in the elementary-event-independent and the Bayesian-network case, threshold BUCQ answering in the frontier-guarded fragment is complete for PP, 2EXP, and 2EXP (EXP, 2EXP, and 2EXP in the weakly frontier-guarded case) in the data, *ba*-combined, and combined complexity, respectively.

In the tuple-independent and the PosBool case, we obtain the same complexity classes, except that the matching lower bounds for the PP and PP^{NP} cases are still open; though, we were able to provide a #P completeness result for the function problem of computing the exact probability of a BUCQ for the PP cases in the data complexity.

Thus, as for the complexity cases above PSPACE, the complexity of threshold BUCQ answering coincides with the complexity of standard BCQ answering (see Table 3).

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

Table 3. Complexity of BCQ answering [22]. All entries except for the "in" ones are completeness results; hardness for all entries but the *fp*-combined ones holds even for ground atomic BCQs.

	Data	***fp*-comb.**	***ba*-comb.**	**Comb.**
L, LF, AF	PP	PP^{NP}	PP^{NP}	PSPACE
G	PP	PP^{NP}	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	PP	PP^{NP}	PP^{NP}	EXP
F, GF	PP	PP^{NP}	PP^{NP}	EXP
A	PP	PP^{NP}	NEXP	NEXP
WS, WA	PP	PP^{NP}	2EXP	2EXP

Table 4. Complexity of threshold query entailment (for both ODE and PODE problems) in the elementary-event-independent and the Bayesian-network case. All entries are completeness results.

	Data	***fp*-comb.**	***ba*-comb.**	**Comb.**
L, LF, AF	in PP	in PP^{NP}	in PP^{NP}	PSPACE
G	in PP	in PP^{NP}	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	in PP	in PP^{NP}	in PP^{NP}	EXP
F, GF	in PP	in PP^{NP}	in PP^{NP}	EXP
A	in PP	in PP^{NP}	NEXP	NEXP
WS, WA	in PP	in PP^{NP}	2EXP	2EXP

Table 5. Complexity of threshold query entailment (for both ODE and PODE problems) for tuple-independent and PosBool annotated probabilistic databases. All entries except for the "in" ones are completeness results.

4.4 Deterministic Ontological Data Exchange

We first consider threshold query answering on target ontologies relative to an ODE problem in Datalog+/– languages where BCQ answering is complete for $\mathcal{C} \supseteq$ PSPACE (see also Table 3 [22]). The following result shows that in these cases, threshold query answering is complete for $\mathcal{C}$ in all four annotation cases. This proves all completeness entries in Tables 4 and 5 above PSPACE.

Theorem 4 *Given (i) 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 complexity class $\mathcal{C} \supseteq$ PSPACE or nondeterministic complexity class $\mathcal{C} =$ NEXP, (ii) a probabilistic source database Pr_s w.r.t. Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and (v) $\tau \in (0, 1]$, deciding whether $\mathbf{a}$ is a τ-*threshold answer to $q(\mathbf{X})$ is complete for $\mathcal{C}$.*

Proof (sketch). As for membership, with annotations consisting of n variables, we create a full valuation of an annotation at a time and compute its probability, which is in PSPACE. Then, we check whether $q(\mathbf{a})$ is true in the corresponding world, which is in $\mathcal{C}$. As we examine one valuation after another, we also add up its probability, if $q(\mathbf{a})$ is true in the corresponding world, until we reach the threshold τ, or we have examined all valuations. Hence, if standard BCQ answering belongs to the deterministic complexity class $\mathcal{C} \supseteq$ PSPACE, then the upper bound is $\mathcal{C}$. If standard BCQ answering belongs to the nondeterministic complexity class $\mathcal{C} =$ NEXP, we guess a set of worlds where $q(\mathbf{a})$ evaluates to true, and verify this guess. Since both steps are in NEXP, the computation is overall also in NEXP.

Hardness is shown by a reduction from BCQ answering in Datalog+/– ontologies to threshold query answering. Consider a source schema $\mathbf{S}$ and a target schema $\mathbf{T}$, as well as a set of n source relations $R_{S,i}$, $1 \leq i \leq n$, and a set of n target relations $R_{T,i}$, $1 \leq i \leq n$, where each $R_{S,i}$ has the same arity as $R_{T,i}$. We also assume a source database with each tuple having the probability 1. The target database is empty. By employing a simple copy mapping from the source database to the target database $R_{S,i}(x_1, \ldots, x_{n_i}) \rightarrow R_{T,i}(y_1, \ldots, y_{n_i})$, and Σ_s being empty, while Σ_t contains a set of TGDs and NCs in the language we consider. Then, a ground instance $\mathbf{a}$ is a *τ-threshold answer* to $q(\mathbf{X})$ with $\tau = 1$ in the ODE problem iff the BCQ $q(\mathbf{a})$ is true for the ontology $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ with the source database. □

The next result shows that in the elementary-event-independent and in the Bayesian-network case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is complete for NP in the *fp*-combined and the *ba*-combined complexity, then threshold query answering is complete for PP^{NP} in the *fp*-combined and the *ba*-combined complexity. This proves all PP^{NP} completeness entries in Table 4.

Theorem 5 *Given (i) 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* NP*-complete in the fp-combined and the ba-combined complexity, and which includes copy TGDs, (ii) an elementary-event-independent or a Bayesian-network-annotated probabilistic source database Pr_s relative to Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and (v) $\tau \in (0, 1]$, deciding whether $\mathbf{a}$ is a τ-*threshold answer to $q(\mathbf{X})$ is PP^{NP}*-complete in the fp-combined and the ba-combined complexity.*

Proof (sketch). As for membership in PP^{NP}, intuitively, we first create multiples of each world (which then correspond to the nondeterministic branches of a Turing machine), so that the probability distribution over all thus generated worlds is the uniform distribution. Then,

for thresholds properly below (resp., above) 0.5, we introduce artificial success (resp., failure) worlds (which correspond to other nondeterministic success (resp., failure) branches of a Turing machine), so that satisfying the resulting threshold corresponds to having a majority of success worlds. We thus only have to verify whether for the majority of the worlds, the query evaluates to true. As query evaluation is in NP, the computation is overall in PP^{NP}.

Hardness for PP^{NP} holds by a reduction from the PP^{NP}-complete problem of deciding, given $n \geq 0$ and a quantified Boolean formula (QBF) $\Phi = \forall x_1 \dots x_l \exists y_1 \dots y_m \ \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_k$, where $l, m, k \geq 1$, and every ϕ_i is a disjunction of literals over $x_1, \dots, x_l, y_1, \dots, y_m$, whether there are at least n truth assignments τ to $x_1, \dots, x_l$ such that $\exists y_1 \dots y_m \ \tau(\phi_1) \wedge \tau(\phi_2) \wedge \dots \wedge \tau(\phi_k)$ is satisfiable [30]. W.l.o.g., every ϕ_i is a disjunction of three literals over $x_1, \dots, x_l, y_1, \dots, y_m$. For every ϕ_i, let u_i, v_i, w_i denote its variables, and let the source database contain the probabilistic facts $r(i, \nu(u_i), \nu(v_i), \nu(w_i))\colon \psi$ such that (i) ν is a truth assignment to u_i, v_i, w_i that satisfies ϕ_i, and (ii) ψ is a conjunction of literals over $\{x_1, \dots, x_l\} \cap \{u_i, v_i, w_i\}$ that exactly represents the restriction of ν to $x_1, \dots, x_l$. Here, every variable x_i has the probability $\mu(x_i) = 0.5$, and thus every world over $x_1, \dots, x_l$ has the probability 2^{-l}. Let Σ_s and Σ_t be empty, and let Σ_{st} contain the copy mapping rule $r(I, U, V, W) \rightarrow r'(I, U, V, W)$. Then, $\exists x_1 \dots x_l \exists y_1 \dots y_m \ r'(1, u_1, v_1, w_1) \wedge r'(2, u_2, v_2, w_2) \wedge \dots \wedge r'(k, u_k, v_k, w_k)$ holds with the probability of at least $n \cdot 2^{-l}$ iff there are at least n truth assignments τ to $x_1, \dots, x_l$ such that $\exists y_1 \dots y_m \ \tau(\phi_1) \wedge \tau(\phi_2) \wedge \dots \wedge \tau(\phi_k)$ is satisfiable. Observe that the above reduction can clearly be done in polynomial time in the size of Φ, and that the set of TGDs and NCs is fixed (*fp*-combined case), and that the arity of all predicates is 4 (*ba*-combined case). □

The next theorem shows that in the elementary-event-independent and in the Bayesian-network case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is in P in the data complexity, then threshold query answering is complete for PP in the data complexity. This proves all PP completeness entries in Table 4.

Theorem 6 *Given (i) 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 in* P *in the data complexity, and which includes copy TGDs, (ii) an elementary-event-independent or a Bayesian-network-annotated probabilistic source database Pr_s relative to Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and (v) $\tau \in (0, 1]$, deciding whether $\mathbf{a}$ is a τ-*threshold answer *to $q(\mathbf{X})$ is* PP*-complete in the data complexity.*

Proof (sketch). The PP-membership proof is similar to the PP^{NP}-membership proof for Theorem 5, except that the standard BCQ query answering oracle is now in P and not in NP.

Hardness for PP holds by a reduction from the PP-complete #3SAT($\geq 2^{n/2}$) decision problem [5]: given a 3CNF formula ϕ with n variables, does ϕ have at least $2^{n/2}$ satisfying truth assignments? Given an instance ϕ of #3SAT($\geq 2^{n/2}$), we construct an ODE problem as follows. We assume a source database with a source schema $\mathbf{S}$ consisting of a binary relation symbol R_S and to contain a single tuple with ϕ as annotation in 3CNF. Each of the n variables of the annotation has the probability 0.5. The schema $\mathbf{T}$ of the target database consists of a binary relation symbol R_T; the target database is empty. The sets Σ_s and Σ_t are empty as well. The set Σ_{st} contains the following mapping rule $R_S(x, y) \rightarrow R_T(x, y)$, which copies the tuple of the source relation to the target relation. Then, the transferred tuple has a probability higher than $2^{-n/2}$ iff the 3CNF formula has at least $2^{n/2}$ satisfying truth assignments, which proves PP-hardness. □

The following theorem says that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is complete for NP in the *fp*-combined and the *ba*-combined complexity, then threshold query answering is in PP^{NP} in the *fp*-combined and the *ba*-combined complexity. The theorem follows from the membership in PP^{NP} of the problem in the more general elementary-event-independent case (see Theorem 5). This result proves all PP^{NP} membership entries in Table 5.

Theorem 7 *Given (i) 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 in* NP *in the fp-combined and the ba-combined complexity, (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and (v) $\tau \in (0, 1]$, deciding whether $\mathbf{a}$ is a τ-*threshold answer *to $q(\mathbf{X})$ is in* PP^{NP} *in the fp-combined and the ba-combined complexity.*

The next theorem shows that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is in P in the data complexity, then threshold query answering is in PP in the data complexity, proving all PP membership entries in Table 5. The theorem also shows that the function problem of computing the exact probability is #P-complete in the data complexity for ontologies and mappings encoded as NCs and full and guarded TGDs (i.e., this #P-completeness holds for the classes G, F, GF, WS, and WA).

Theorem 8 *Given (i) 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 in* P *in the data complexity, (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, and (v) $\tau \in (0, 1]$, deciding whether $\mathbf{a}$ is a τ-*threshold answer *to $q(\mathbf{X})$ is in* PP *in the data complexity. Given (i) to (iv), where $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ is full and guarded, computing $conf_q(\mathbf{a})$ in Pr_s relative to $\mathcal{M}$ is* #P*-complete in the data complexity.*

Proof (sketch). Membership in PP is immediate by the membership in PP of the problem in the more general elementary-event-independent case (see Theorem 6). Membership in #P of the function problem follows by a similar line of argumentation.

Hardness for #P follows from a reduction from the #P-complete monotone 2SAT problem [28]: given a Boolean formula $\phi = \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_k$ over the variables $x_1, x_2, \dots, x_l$, where each ϕ_i is a disjunction of two variables, compute the number of truth assignments to $x_1, x_2, \dots, x_l$ that satisfy ϕ. For every variable x_i, let the source database contain the probabilistic facts $s(x_i)\colon x_i$, along with the probability $\mu(x_i) = 0.5$. Furthermore, for every ϕ_i, let u_i and v_i denote its variables, and let the source database contain the probabilistic facts $r(i, u_i)\colon e_{r(i,u_i)}$ and $r(i, v_i)\colon e_{r(i,v_i)}$, along with the probabilities $\mu(e_{r(i,u_i)}) = 1$ and $\mu(e_{r(i,v_i)}) = 1$. For every $i, j \in \{1, \dots, k\}$ with $i < j$, let the source database contain the probabilistic facts $succ(i, j)\colon e_{succ(i,j)}$, along with the probability $\mu(e_{succ(i,j)}) = 1$. Furthermore, let the source database contain the probabilistic fact $max(k)\colon e_{max(k)}$, along with the probability $\mu(e_{max(k)}) = 1$. Let Σ_t be empty, and let Σ_s contain the full and

guarded rules $r(I,X) \wedge s(X) \rightarrow t(I)$, $max(X) \wedge t(X) \rightarrow t'(X)$, and $t'(I) \wedge succ(J,I) \wedge r(J) \rightarrow t'(J)$. Let Σ_{st} contain the mapping rule $t'(X) \rightarrow t''(X)$. Observe that the query and the set of TGDs and NCs are both fixed, and that the TGDs are full and guarded. Furthermore, $t''(1)$ holds with the probability m iff ϕ has $m \cdot 2^l$ satisfying truth assignments τ to $x_1, \ldots, x_l$. □

The following theorem shows that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings is in a class of TGDs and NCs where standard BCQ answering is in AC^0 in the data complexity, then the function problem of computing the exact probability is also #P-complete in the data complexity (i.e., this #P-completeness holds for the classes L, LF, AF, S, SF, and A).

Theorem 9 *Given (i) an ODE problem $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ is in a class of TGDs and NCs for which BCQ answering is in AC^0, (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s, (iii) a UCQ $q(\mathbf{X})$ over $\boldsymbol{T}$, and (iv) a ground instance $\mathbf{a}$ of $\mathbf{X}$ over $\mathbf{C}$, computing $conf_q(\mathbf{a})$ in Pr_s relative to $\mathcal{M}$ is #P-complete in the data complexity.*

Proof (sketch). Membership in #P of the function problem follows by a similar line of argumentation as in the proof of Theorem 8.

Hardness for #P follows from the observation that the #P-complete evaluation of unsafe UCQs in [12] is actually a special case of our function problem of computing the probability of a query. □

4.5 Tractable Cases

In the tuple-independent case, if the language of the ontologies and the mappings is in a class of TGDs and NCs where standard BCQ answering is in AC^0 in the data complexity, we arrive at a tuple-independent source database with a rewritten first-order UCQ q_Σ (e.g., by applying the algorithm XRewrite from [17] to the initial query $q(\mathbf{X})$ on the target database and $\Sigma = \Sigma_t \cup \Sigma_{st} \cup \Sigma_s$). This is now exactly the problem handled in [27, 12], which has important tractable cases (called *safe queries*): they are those queries that can be computed by extensional query evaluation, i.e., solely by the query syntax or its annotation. Extensional query evaluation consists of extending the relational operators with a probabilistic semantics. An example of tractable queries are hierarchical queries. A query is hierarchical, if for each existential sub-query expression $\exists x\, q_{\text{sub}}$, the quantified variable x occurs in all atoms of q_{sub}. Every hierarchical query has a read-once Boolean formula as query annotation. Further examples for tractable cases can be found in [27, 12]. In particular, [27] delineates six syntactical rules to check whether UCQs are tractable, and calls a UCQ q R_6-safe, if it adheres to one of them; if a UCQ is not R_6-safe, it is #P-hard. In [27, 12], a polynomial algorithm for R_6-safe UCQs is given involving Möbius' inversion function, which ensures that it also covers UCQs, and not just CQs. If the query is not R_6-safe, it may be approximated.

4.6 Probabilistic Ontological Data Exchange

All the results in Theorems 4, 5, 6, 7, 8 and 9 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, we also have to consider and iterate through the worlds for the probabilistic mapping. However, iterating through these worlds in addition to the worlds for the probabilistic source database does not increase the overall complexity of threshold query answering.

5 SUMMARY AND OUTLOOK

We have studied the impact of different probabilistic models and compact encodings on the computational complexity of the problem of threshold query answering in ontological data exchange with the main Datalog+/– languages for representing the source and target ontologies as well as the mappings. We have considered the data complexity, the *fp*-combined complexity, the *ba*-combined complexity, and the combined complexity. We have provided a complete picture for the elementary-event-independent and the Bayesian network case with a compact encoding with Boolean formulas. For tuple-independent and PosBool-annotated probabilistic databases, we have provided either completeness results or upper bounds.

While ontology and mapping languages with BCQ answering complexity above PSPACE dominate the complexity of threshold query answering, for ontology languages with BCQ answering complexity below PSPACE, we obtain interesting results, one of them being a potential dichotomy of either an upper bound of P or PP-hardness in the data complexity for threshold query answering in the first-order-rewritable cases. A similar dichotomy of either an upper bound of P or #P-hardness in the data complexity exists for query answering in probabilistic databases (see [27, 12]) and has been lifted to OWL QL in [21], which is also first-order-rewritable. Note that OWL QL corresponds to *DL-Lite*$_R$, which is strictly less expressive than most of the Datalog+/– languages considered here.

Another interesting result is PP-completeness for threshold query answering in the data complexity for most of the Datalog+/– languages that we considered for the elementary-event-independent and the Bayesian network case. Furthermore, we have obtained PP^{NP}-completeness in the *fp*-combined and *ba*-combined complexity for the elementary-event-independent and the Bayesian-network case for threshold query answering for several Datalog+/– languages. This complexity class is mostly known from a blow-up in succinct representations of problems (see [30]).

There is no related work on threshold query entailment in probabilistic data exchange or probabilistic ontological data exchange. Most other works on probabilistic databases and ontologies do not consider threshold query answering, but consider the function problem of query answering, mostly in the data complexity, such as the works [27, 12, 21]. Perhaps closest to our work is [11], where threshold query answering has been studied under the name "probabilistic query entailment" for the ontology language $\mathcal{EL}$, annotated with annotations that are related to our Bayesian-network annotations. Our work goes beyond that, as most of the ontology languages that we are considering here are strictly more expressive than $\mathcal{EL}$. In addition, we also consider a probabilistic model where the events are elementary-event-independent annotations and PosBool annotations, as well as tuple-independent and mapping-independent annotations. In addition, our complexity analysis for threshold query entailment with Bayesian networks as probabilistic model contains also completeness results for the languages we considered, while this is not the case in [11]. Threshold query answering with probabilistic ontologies has also been studied in [16], but the probabilistic uncertainty models used there are Markov logic networks.

An interesting topic for future research is a more detailed analysis of the tractable case for threshold query answering, especially also in combination with ontological query rewriting. Based on the complexity results and membership proofs of this paper, another topic for future research is to explore whether there are other (special-case or approximation) tractable cases for threshold query answering.

ACKNOWLEDGEMENTS

This work was supported by the UK EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1, and the EU FP7 Marie-Curie IE Fellowship PRODIMA.

REFERENCES

[1] Marcelo Arenas, Elena Botoeva, Diego Calvanese, and Vladislav Ryzhikov, 'Exchanging OWL 2 QL knowledge bases', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, (2013).

[2] Marcelo Arenas, Elena Botoeva, Diego Calvanese, Vladislav Ryzhikov, and Evgeny Sherkhonov, 'Exchanging description logic knowledge bases', in *Proceedings of 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*, pp. 563–567, (2012).

[3] Marcelo Arenas, Jorge Pérez, and Juan Reutter, 'Data exchange beyond complete data', *Journal of the ACM*, **60**(4), (2013).

[4] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo, 'Walking the complexity lines for generalized guarded existential rules', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pp. 712–717. IJCAI/AAAI Press, (2011).

[5] Delbert D. Bailey, Victor Dalmau, and Phokion G. Kolaitis, 'Phase transitions of PP-complete satisfiability problems', *Discrete Applied Mathematics*, **155**(12), (2007).

[6] Pablo Barcelo, 'Logical foundations of relational data exchange', *SIGMOD Record*, **38**(1), 49–58, (2009).

[7] Andrea Calì, Georg Gottlob, and Michael Kifer, 'Taming the infinite chase: Query answering under expressive relational constraints', *Journal of Artificial Intelligence Research*, **48**, (2013).

[8] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz, 'A general Datalog-based framework for tractable query answering over ontologies', *Journal of Web Semantics*, **14**, 57–83, (2012).

[9] Andrea Calì, Georg Gottlob, Thomas Lukasiewicz, Bruno Marnette, and Andreas Pieris, 'Datalog+/–: A family of logical knowledge representation and query languages for new applications', in *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)*, (2010).

[10] Andrea Calì, Georg Gottlob, and Andreas Pieris, 'Towards more expressive ontology languages: The query answering problem', *Artificial Intelligence*, **139**, (2012).

[11] Ismail Ilkan Ceylan and Rafael Penaloza, 'Probabilistic query answering in the Bayesian description logic $\mathcal{BEL}$', in *Proceedings of the 9th International Conference on Scalable Uncertainty Management (SUM 2015)*, (2015).

[12] Nilesh Dalvi and Dan Suciu, 'The dichotomy of probabilistic inference for union of conjunctive queries', *Journal of the ACM*, **59**(6), (2012).

[13] Ronald Fagin, Benny Kimelfeld, and Phokion G. Kolaitis, 'Probabilistic data exchange', *Journal of the ACM*, **58**(4), (2011).

[14] Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian Popa, 'Data exchange: Semantics and query answering', *Theoretical Computer Science*, **336**(1), 89–124, (2005).

[15] Norbert Fuhr and Thomas Rölleke, 'A probabilistic relational algebra for the integration of information retrieval and database systems', *ACM Transactions on Information Systems*, **15**(1), 32–66, (1997).

[16] Georg Gottlob, Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari, 'Query answering under probabilistic uncertainty in Datalog+/– ontologies', *Annals of Mathematics and Artificial Intelligence*, **69**(1), (2013).

[17] Georg Gottlob, Giorgio Orsi, and Andreas Pieris, 'Query rewriting and optimization for ontological databases', *ACM Transactions on Database Systems*, **39**(3), (2014).

[18] Todd J. Green, Grigoris Karvounarakis, and Val Tannen, 'Provenance semirings', in *Proceedings of the 26th Symposium on Principles of Database Systems (PODS 2007)*, (2007).

[19] Tomasz Imielinski and Witold Lipski, 'Incomplete information in relational databases', *Journal of the ACM*, **31**(4), 761–791, (1984).

[20] David S. Johnson, 'A catalog of complexity classes', in *Handbook of Theoretical Computer Science*, 67–161, MIT Press, (1990).

[21] Jean Christoph Jung and Carsten Lutz, 'Ontology-based access to probabilistic data with OWL QL', in *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, (2012).

[22] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris, and Gerardo I. Simari, 'From classical to consistent query answering under existential rules', in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, (2015).

[23] Thomas Lukasiewicz, Maria Vanina Martinez, Livia Predoiu, and Gerardo I. Simari, 'Existential rules and Bayesian networks for probabilistic ontological data exchange', in *Rule Technologies: Foundations, Tools, and Applications*, pp. 294–310, (2015).

[24] Thomas Lukasiewicz, Maria Vanina Martinez, Livia Predoiu, and Gerardo I. Simari, 'Basic probabilistic ontological data exchange with existential rules', in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, (2016).

[25] Christos H. Papadimitriou, *Computational Complexity*, Addison Wesley Longman, 1994.

[26] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', *Journal on Data Semantics*, **10**, 133–173, (2008).

[27] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch, *Probabilistic Databases*, Morgan & Claypool, 2011.

[28] Leslie G. Valiant, 'The complexity of enumeration and reliability problems', *SIAM Journal on Computing*, **8**(3), (1979).

[29] Moshe Y. Vardi, 'The complexity of relational query languages (extended abstract)', in *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC 1982)*, pp. 137–146, (1982).

[30] Klaus W. Wagner, 'The complexity of combinatorial problems with succinct input representation', *Acta Informatica*, **23**(3), 325–356, (1986).