Probabilistic Data Exchange

RONALD FAGIN and BENNY KIMELFELD, IBM Research – Almaden
PHOKION G. KOLAITIS, University of California, Santa Cruz and IBM Research – Almaden

15

The work reported here lays the foundations of data exchange in the presence of probabilistic data. This requires rethinking the very basic concepts of traditional data exchange, such as solution, universal solution, and the certain answers of target queries. We develop a framework for data exchange over probabilistic databases, and make a case for its coherence and robustness. This framework applies to arbitrary schema mappings, and finite or countably infinite probability spaces on the source and target instances. After establishing this framework and formulating the key concepts, we study the application of the framework to a concrete and practical setting where probabilistic databases are compactly encoded by means of annotations formulated over random Boolean variables. In this setting, we study the problems of testing for the existence of solutions and universal solutions, materializing such solutions, and evaluating target queries (for unions of conjunctive queries) in both the exact sense and the approximate sense. For each of the problems, we carry out a complexity analysis based on properties of the annotation, for various classes of dependencies. Finally, we show that the framework and results easily and completely generalize to allow not only the data, but also the schema mapping itself to be probabilistic.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Systems—*Query processing; Relational databases*; H.2.5 [**Database Management**]: Heterogeneous Databases—*Data translation*

General Terms: Theory

Additional Key Words and Phrases: Data exchange, data integration, probabilistic database, probabilistic schema mapping, probabilistic solution, universal probabilistic solution, conjunctive query, certain answer, computational complexity

ACM Reference Format:
Fagin, R., Kimelfeld, B., and Kolaitis, P. G. 2011. Probabilistic data exchange. J. ACM 58, 4, Article 15 (July 2011), 55 pages.
DOI = 10.1145/1989727.1989729 http://doi.acm.org/10.1145/1989727.1989729

1. INTRODUCTION

Data exchange is the problem of transforming data that conform to one schema, the *source schema*, into data that conform to another schema, the *target schema*, in a way that is consistent with various *dependencies* (i.e., constraints expressed in some logical formalism over the two schemas). The source and target schemas, along with the dependencies, define a *schema mapping*, and the results of the transformation of a source instance are called *solutions*. Traditional data exchange is based on the

An abridged version of this article was published in *Proceedings of the 13th International Conference on Database Theory* [Fagin et al. 2010].
The research of P. G. Kolaitis is partially supported by NSF grants IIS-0430994 and ARRA-0905276.
Authors' addresses: R. Fagin and B. Kimelfeld, IBM Almaden Research Center, Dept. K53/B2, 650 Harry Road, San Jose, CA 95120-6099; email: {fagin, kimelfeld}@us.ibm.com; P. G. Kolaitis, Computer Science Department, University of California, Santa Cruz, E2-345A, Santa Cruz, CA 95064; email: kolaitis@cs.ucsc.edu.

DOI 10.1145/1989727.1989729 http://doi.acm.org/10.1145/1989727.1989729

assumption that source data are certain. However, the need to account for uncertainty in data has long been recognized [Barbará et al. 1992; Fuhr and Rölleke 1997]. In view of the advent of the Web and related modern applications, models of uncertain data (typically *probabilistic databases*) have recently gained significant renewed focus [Cohen et al. 2008; Dalvi and Suciu 2004, 2007a; Green and Tannen 2006; Kimelfeld and Sagiv 2007; Koch 2008; Sarma et al. 2008b; Senellart and Abiteboul 2007]. It is, therefore, essential to rethink the conceptual framework of data exchange in the context of uncertainty in the source data.

Our goal in this article is to lay the foundations of data exchange in the presence of probabilistic data. This is accomplished in two main parts. First, in Sections 2–4, we establish a framework that extends and generalizes traditional data exchange to probabilistic (source and target) databases. This framework is general, in the sense that it imposes essentially no restriction at all on the types of dependencies or on the probabilistic databases (which are finite or countably infinite spaces of ordinary finite databases, where each database is assigned a probability). Then, in Section 5, we apply our framework to a concrete and practical setting, where the dependencies are from widely-studied classes, and where the probabilistic databases are compactly encoded in various conventional manners (e.g., as in Agrawal et al. [2006], Boulos et al. [2005], Dalvi and Suciu [2004], Kimelfeld and Sagiv [2007], and Sarma et al. [2008b]).

Furthermore, in Section 6, we extend the framework and the results to allow not only the data, but also the schema mapping to be probabilistic. In principle, we could use this extended setting right from the beginning. The reason for not doing so is that it would significantly increase the complexity of the presentation, while the key challenges and ideas arise already when only the data are probabilistic.

Formally, a *schema mapping* is a triple $(\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are the source and target schemas, respectively, and Σ is a set of dependencies formulated as logical assertions over $\mathbf{S}$ and $\mathbf{T}$. A *source instance* is an instance I over $\mathbf{S}$, and a *target instance* is an instance J over $\mathbf{T}$; moreover, J is allowed to include *labeled nulls*, which are essentially variables that are not bound to specific values. A target instance J is a *solution* if the pair $\langle I, J\rangle$ satisfies Σ. In this article, source and target instances are replaced with *probabilistic instances* (abbreviated, *p-instances*): a *source p-instance* is a probability space $\tilde{\mathcal{I}}$ over the source instances, and a *target p-instance* is a probability space $\tilde{\mathcal{J}}$ over the target instances.

The first task is, naturally, to define a *probabilistic solution* (abbreviated, *p-solution*) for a source p-instance with respect to a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$. Essentially, we define a target p-instance $\tilde{\mathcal{J}}$ to be a *p-solution* for a p-instance $\tilde{\mathcal{I}}$ if there exists a probability space over source-solution pairs (I, J) (i.e., J is a solution for I with respect to Σ), such that the marginals coincide with the p-instance $\tilde{\mathcal{I}}$ on the one hand, and with the p-instance $\tilde{\mathcal{J}}$ on the other. Our definition of a p-solution is based on the classical concept of a *bivariate* (joint) probability space with given marginals (research of this concept goes back to the 1950s [Fréchet 1951; Morgenstern 1956]), but with the additional requirement that the *support* (i.e., the set of samples with a nonzero probability) is contained in a fixed relation (in this case, the source-solution relation). To explore the coherence of this definition, we formulate two intuitive properties that every reasonable concept of a solution should satisfy. Each of these properties says that a p-solution properly reflects the uncertainty of the source data. Rather surprisingly, we show that each of the two properties is actually a characterization of a p-solution.

We then proceed to the adaptation of the notion of a *universal solution*. Recall that a solution is universal if it can be homomorphically mapped into every other solution [Fagin et al. 2005a]. Our definition of a *universal p-solution* is similar to that of a p-solution (given previously), except that we require the existence of a probability space over pairs (I, J), such that J is a universal solution for I (and, again, the

marginals coincide with $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$). On the surface, this definition does not imply any desired semantic property. In traditional data exchange, a universal solution J is a "good" solution in the sense that it generalizes all the other solutions, since every solution contains a homomorphic image of J. We want a similar property to characterize a universal p-solution. For that, we need to figure out the meaning of *generalization* between p-instances.

There are various ways of formally modeling the generalization relationship between p-instances; we consider three natural definitions, where each of the three extends the traditional concept (existence of a homomorphism) to p-instances. One definition is (again) in terms of a bivariate distribution, and the other two are based on the notion of a *stochastic order* (see, e.g., Shaked and Shanthikumar [1994]). We show that the three are different from one another; moreover, in the finite case, the complexity of testing whether the relationship holds for the first notion is different from the complexity of testing for the other two. So, we do not have one robust formalization of the generalization relationship between p-solutions. A priori, each of the three relationships could imply a different alternative definition of a universal p-solution, namely, one that "generalizes" all the p-solutions. Quite remarkably, the three definitions are all equivalent to the preceding definition of a universal p-solution. Furthermore, as we show next, when we consider the concept of answering target queries, a universal p-solution is also characterized by its usefulness in answering target conjunctive queries (as in the deterministic case [Fagin et al. 2005a]). These results indicate that the concept of a universal p-solution is very robust.

Since a solution in our framework (namely, a p-solution) is inherently probabilistic, evaluating target queries amounts to querying probabilistic databases. In particular, for a source p-instance $\tilde{\mathcal{I}}$ and a query q, every p-solution $\tilde{\mathcal{J}}$ gives a (potentially different) confidence value for each possible answer $\mathbf{a}$. Consistent with the approach of "certain answers" in traditional data exchange, the *confidence* of $\mathbf{a}$ is defined to be the infimum of the confidence values for $\mathbf{a}$ over all p-solutions. We show that (when a p-solution exists) this is the same as the probability that $\mathbf{a}$ is a certain answer for a random source instance of $\tilde{\mathcal{I}}$. We show that a universal p-solution can be used for answering unions of conjunctive queries (UCQs), namely, evaluation thereon gives the correct confidence values. Moreover, if a p-solution can be used this way in the evaluation of conjunctive queries, then this p-solution is necessarily universal.

We then proceed to study algorithmic and computational aspects of data exchange for finite probabilistic databases. Specifically, we consider the following problems: testing for the existence of solutions and universal solutions, materializing such solutions, and evaluating target unions of conjunctive queries. It follows from our results that these problems are not harder than their counterparts in the traditional (deterministic) setting. That holds, though, under the assumption that the source p-instance is represented in an explicit manner (i.e., by specifying each possible world I along with its probability). This is at odds with conventional practice, which is to associate a measure of confidence (or a probabilistic event) with each fact. Such a representation (along with some statistical assumptions) is typically logarithmic-scale compact. So, following existing representations (e.g., *ULDBs* [Agrawal et al. 2006; Sarma et al. 2008b], *probabilistic c-tables* [Green and Tannen 2006] and *probabilistic trees* [Senellart and Abiteboul 2007]), we explore a setting where the source p-instance is represented compactly by *annotating* facts with *conditions*, which are formulas over a set of (Boolean and probabilistically independent) random variables. We consider two types of annotations. In a *DNF instance*, the annotation is in disjunctive normal form; in a *tuple-independent instance*, different facts are probabilistically independent, and the annotation effectively specifies the probability of each fact, as done in Boulos et al. [2005] and Dalvi and Suciu [2004, 2007a].

Our analysis is based on data complexity, which is common in studying the complexity aspects of data exchange, (e.g., Fagin et al. [2005a, 2005b, 2005c] and Gottlob and Nash [2008]). Thus, we hold fixed a schema mapping and a query (when relevant), and the input consists of an annotated (i.e., DNF or tuple-independent) source instance. In our analysis, we consider the types of dependencies that were studied in Fagin et al. [2005a]. Thus, we allow *st-tgds* (source-to-target tgds), *t-tgds* (target tgds),[1] and *t-egds* (target egds). We consider also the effect on the complexity when the st-tgds and/or t-tgds are restricted to being full. We divide the computational problems into categories that correspond to all possible combinations of dependency and annotation types. We start with the problems of testing whether a (universal) solution exists and of materializing one that is encoded as a DNF instance. For each category, we show that either the corresponding problem is tractable for all schema mappings (in the category) or that there exists a schema mapping (in the category) for which the problem is intractable. We then consider target-query evaluation and, in particular, show that every nontrivial UCQ is #P-hard in some schema mapping of the most restrictive category (namely, independent facts and full st-tgds). Due to this hardness, we study the complexity of *approximate* query evaluation (which, in practice, is often good enough), and give the following complete classification. For each category, we prove one of the following two alternatives.

—*Alternative* 1. For every schema mapping and for every target UCQ there exists an efficient algorithm (randomized or deterministic) for approximate query evaluation.
—*Alternative* 2. For every nontrivial[2] target UCQ there exists a schema mapping in which query evaluation is hard to approximate.

Finally, we show how to generalize the framework and all of the aforementioned results to accommodate probabilistic schema mappings (in addition to probabilistic data). The combination of a probabilistic schema mapping with a source p-instance requires having a joint probability distribution over sets of dependencies and source instances; that is, a probability space on pairs (Σ, I), where Σ is a set of dependencies and I is a source instance. We call such a probability distribution a *probabilistic problem* (*p-problem*, in short). In general, a p-problem allows for every correlation between the probabilistic mapping and the source p-instance; a special case is the *product space* where the probabilistic schema mapping and the source p-instance are assumed to be independent.

We show that the framework and all aforementioned results completely generalize to p-problems, under the proper adaptation of the definitions. In particular, we use the notions of a p-solution, a universal p-solution, and an answer confidence (for a target query) for a p-problem $\tilde{\mathcal{P}}$ rather than for a source p-instance $\tilde{\mathcal{I}}$. Moreover, the results of Section 5 are generalized by annotating the dependencies specifying the mapping similarly to source facts (i.e., using formulas over event variables); event variables can be shared between facts and dependencies, thereby allowing correlations between the probabilistic source data and mappings to be represented.

To the best of our knowledge, this work (which was originally reported in the abridged version of this article [Fagin et al. 2010]), is the first to study data exchange over probabilistic databases. In Dong et al. [2007, 2009] and Sarma et al. [2008a, 2009], data integration is studied in the context of deterministic databases and probabilistic mappings. The relationship between these articles and the present article is discussed in Section 6.2.

In order to simplify and shorten the presentation, some of the proofs appear in the Appendix.

[1]We make the now standard assumption of *weak acyclicity* [Fagin et al. 2005a].
[2]We actually use two natural notions of *triviality* (see Section 5).

2. PRELIMINARIES

2.1. Schemas and Instances

We assume fixed countably infinite sets Const of *constants* and Var of *nulls*, such that $\mathsf{Const} \cap \mathsf{Var} = \emptyset$. A *schema* is a finite sequence $\mathbf{R} = \langle R_1, \ldots, R_k \rangle$ of distinct relation symbols, where each R_i has a fixed arity $r_i > 0$. An *instance* I (*over* $\mathbf{R}$) is a sequence $\langle R_1^I, \ldots, R_k^I \rangle$, such that each R_i^I is a finite relation of arity r_i over $\mathsf{Const} \cup \mathsf{Var}$ (i.e., R_i^I is a finite subset of $(\mathsf{Const} \cup \mathsf{Var})^{r_i}$). We call R_i^I the R_i*-relation* of I. We may abuse this notation and use R_i to denote both the relation symbol and the relation R_i^I that interprets it. We use $\mathsf{dom}(I)$ to denote the set of all constants and nulls that appear in I. We say that I is a *ground instance* if $\mathsf{dom}(I)$ does not contain nulls. We denote by $\mathsf{Inst}(\mathbf{R})$ and $\mathsf{Inst}^{\mathsf{c}}(\mathbf{R})$ the classes of all instances and ground instances, respectively, over $\mathbf{R}$. We use $R(t_1, \ldots, t_r)$ to denote that $(t_1, \ldots, t_r)$ is a tuple in a relation R and call it a *fact*. We identify an instance with the set of its facts.

Let K_1 and K_2 be instances over the same schema. A *homomorphism* $h : K_1 \to K_2$ is a mapping from $\mathsf{dom}(K_1)$ to $\mathsf{dom}(K_2)$, such that (1) $h(c) = c$ for all constants $c \in \mathsf{dom}(K_1)$, and (2) for all facts $R(\mathbf{t})$ of K_1, the fact $R(h(\mathbf{t}))$ is in K_2 (for $\mathbf{t} = (t_1, \ldots, t_r)$, the tuple $h(\mathbf{t})$ is $(h(t_1), \ldots, h(t_r))$). By $K_1 \to K_2$ we denote the existence of a homomorphism $h : K_1 \to K_2$.

2.2. Schema Mappings

We now describe our formalism of a *schema mapping*, which follows that of Fagin et al. [2005a]. Suppose that $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ and $\mathbf{T} = \langle T_1, \ldots, T_m \rangle$ are two schemas with no relation symbols in common. We denote by $\langle \mathbf{S}, \mathbf{T} \rangle$ the schema that is obtained by concatenating $\mathbf{S}$ and $\mathbf{T}$. Similarly, if I and J are instances of $\mathbf{S}$ and $\mathbf{T}$, respectively, then $\langle I, J \rangle$ is the instance $K \in \mathsf{Inst}(\langle \mathbf{S}, \mathbf{T} \rangle)$ that satisfies $S_i^K = S_i^I$ and $T_j^K = T_j^J$ for $1 \leq i \leq n$ and $1 \leq j \leq m$; in other words, since we identify an instance with the set of its facts, $\langle I, J \rangle$ is essentially the union of I and J.

We assume some formalism for expressing constraints over a given schema $\mathbf{R}$. If $I \in \mathsf{Inst}(\mathbf{R})$ and Σ is a set of formulas in this formalism, then $I \models \Sigma$ denotes that I satisfies every formula of Σ.

A *schema mapping* is a triple $(\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ (the *source schema*) and $\mathbf{T}$ (the *target schema*) are schemas without common relation symbols, and Σ is a set of formulas over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$. Each formula of Σ is called a *dependency*. A *source instance* is a ground instance I over $\mathbf{S}$, and a *target instance* is an instance J over $\mathbf{T}$ (that is, $I \in \mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$ and $J \in \mathsf{Inst}(\mathbf{T})$). We say that the target instance J is a *solution for* I (*with respect to* Σ) if $\langle I, J \rangle \models \Sigma$. A solution J for I with respect to Σ is *universal* if $J \to J'$ for all solutions J' for I with respect to Σ (in other words, every solution contains a homomorphic image of J).

2.3. Probability Spaces

All the probability spaces we consider are countable (finite or countably infinite). We call such spaces *p-spaces* and use the following notation. A p-space is a pair $\tilde{\mathcal{U}} = (\Omega(\tilde{\mathcal{U}}), p_{\tilde{\mathcal{U}}})$, such that $\Omega(\tilde{\mathcal{U}})$ is a countable set and $p_{\tilde{\mathcal{U}}} : \Omega(\tilde{\mathcal{U}}) \to [0, 1]$ is a function that satisfies $\sum_{u \in \Omega(\tilde{\mathcal{U}})} p_{\tilde{\mathcal{U}}}(u) = 1$. Each member u of $\Omega(\tilde{\mathcal{U}})$ is a *sample*, and $\Omega(\tilde{\mathcal{U}})$ is the *sample space*. We say that the p-space $\tilde{\mathcal{U}}$ is *over* $\Omega(\tilde{\mathcal{U}})$. The *support* of $\tilde{\mathcal{U}}$, denoted $\Omega_+(\tilde{\mathcal{U}})$, is the set of all samples $u \in \Omega(\tilde{\mathcal{U}})$ such that $p_{\tilde{\mathcal{U}}}(u) > 0$. We say that $\tilde{\mathcal{U}}$ is *finite* if its support $\Omega_+(\tilde{\mathcal{U}})$ is finite. A subset $X \subseteq \Omega(\tilde{\mathcal{U}})$ is called an *event*. The *probability* of the event X, denoted $\Pr_{\tilde{\mathcal{U}}}(X)$, is the sum $\sum_{u \in X} p_{\tilde{\mathcal{U}}}(u)$. We may omit the subscript $\tilde{\mathcal{U}}$ from $\Pr_{\tilde{\mathcal{U}}}(X)$ when it is clear from the context. We use $\mathcal{U}$ (i.e., without the tilde sign) to denote the *random variable* that represents a sample of $\tilde{\mathcal{U}}$. An event is often represented by a logical formula over $\mathcal{U}$ (e.g., $\Pr_{\tilde{\mathcal{U}}}(\varphi(\mathcal{U}))$ is the same as $\Pr_{\tilde{\mathcal{U}}}(\{u \in \Omega(\tilde{\mathcal{U}}) \mid \varphi(u)\})$). We often abuse the above notation

and identify $\tilde{\mathcal{U}}$ with its sample space $\Omega(\tilde{\mathcal{U}})$ (e.g., $u \in \tilde{\mathcal{U}}$ means that u is a member of $\Omega(\tilde{\mathcal{U}})$).

Let U and W be two countable sets, and let $\tilde{\mathcal{P}}$ be a p-space over $U \times W$ (that is, $\tilde{\mathcal{P}} = (\Omega(\tilde{\mathcal{P}}), p_{\tilde{\mathcal{P}}})$, where $\Omega(\tilde{\mathcal{P}}) = U \times W$). The *left marginal* of $\tilde{\mathcal{P}}$ is the p-space $\tilde{\mathcal{U}}$, such that $\Omega(\tilde{\mathcal{U}}) = U$ and for all $u \in U$ it holds that $p_{\tilde{\mathcal{U}}}(u) = \sum_{w \in W} p_{\tilde{\mathcal{P}}}(u, w)$. Similarly, the *right marginal* of $\tilde{\mathcal{P}}$ is the p-space $\tilde{\mathcal{W}}$, such that $\Omega(\tilde{\mathcal{W}}) = W$ and for all $w \in W$ it holds that $p_{\tilde{\mathcal{W}}}(w) = \sum_{u \in U} p_{\tilde{\mathcal{P}}}(u, w)$.

3. EXCHANGING PROBABILISTIC DATA

Our goal is to study data exchange in the presence of uncertainty in the source instance. We use the convention of modeling uncertain data as a probabilistic database [Dalvi and Suciu 2004, 2007a; Green and Tannen 2006; Koch 2008; Sarma et al. 2008b]. The challenges of this generalization of data exchange arise right in the beginning: What is the meaning of a solution for a probabilistic source instance? The first observation is that such a solution should by itself be probabilistic because if the source database is uncertain, then so is the target database. Next, we formalize the notion of a probabilistic database.

Let $\mathbf{R}$ be a schema. A *probabilistic database*, or a *probabilistic instance* (over $\mathbf{R}$), abbreviated p-instance, is a p-space $\tilde{\mathcal{I}}$ over $\mathsf{Inst}(\mathbf{R})$. If $\tilde{\mathcal{I}}$ is a p-space over $\mathsf{Inst}^{\mathsf{c}}(\mathbf{R})$, then $\tilde{\mathcal{I}}$ is a *ground p-instance*. Note that the sample space $\mathsf{Inst}(\mathbf{R})$ (or $\mathsf{Inst}^{\mathsf{c}}(\mathbf{R})$) is countable due to our assumption that Const and Var are countable (and that ordinary instances are finite).

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. A *source p-instance* is a ground p-instance $\tilde{\mathcal{I}}$ over $\mathbf{S}$, and a *target p-instance* is a p-instance $\tilde{\mathcal{J}}$ over $\mathbf{T}$. In other words, a source p-instance and a target p-instance are p-spaces over the source and target instances, respectively.

Example 3.1. Let $\mathcal{M}$ be the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are defined as follows. Note that, for convenience, the columns are named.

$$\mathbf{S} : \mathit{Researcher}(\text{name, university}),\ \mathit{RArea}(\text{researcher, topic})$$
$$\mathbf{T} : \mathit{UArea}(\text{university, department, topic})$$

and Σ contains the single dependency

$$\forall r, u, t(\mathit{Researcher}(r, u) \wedge \mathit{RArea}(r, t) \rightarrow \exists d\, \mathit{UArea}(u, d, t)).$$

Figure 1 depicts a set of possible facts (e.g., r_{e}, a_{eir} and u_{ir}) for each of the three relations. Note that $\perp_1$, $\perp_2$ and $\perp_3$ are nulls (and the rest of the data values are constants). Using the facts, the figure depicts four finite p-instances $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}_1$, $\tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_3$, where $\tilde{\mathcal{I}}$ is a source p-instance and each $\tilde{\mathcal{J}}_i$ is a target p-instance. Each sample of a p-instance is represented by a two-entry row, where the left entry shows the instance and the right one shows its probability. For example, the probability $p_{\tilde{\mathcal{I}}}(I_1)$ of $I_1 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{eir}}, a_{\mathsf{jdb}}\}$ is 0.3. Observe that in each of $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}_1$, $\tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_3$, the probabilities in the rows sum up to 1.

The challenge is to identify when a target p-instance constitutes a solution for a source p-instance. In principle, we have a binary relationship between deterministic instances I and J (namely, J is a solution for I), and we want to generalize it to p-instances $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$. We introduce the *probabilistic match*, which is a systematic way of extending a binary relationship between objects into a binary relationship between p-spaces thereof. Next, we give the formal definition of a probabilistic match. In Section 3.2, we apply it to define our notion of a solution in the probabilistic setting, which we call a p-solution. Then, we show that this definition is semantically coherent,

Possible *Researcher* facts

r_{e}	*Researcher*(Emma, UCSD)
r_{j}	*Researcher*(John, UCSD)

Possible *RArea* facts

a_{eir}	*RArea*(Emma, IR)
a_{edb}	*RArea*(Emma, DB)
a_{jdb}	*RArea*(John, DB)
a_{jai}	*RArea*(John, AI)

Source p-instance $\tilde{\mathcal{I}}$

$I_1 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{eir}}, a_{\mathsf{jdb}}\}$	0.3
$I_2 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{eir}}, a_{\mathsf{jai}}\}$	0.3
$I_3 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{edb}}, a_{\mathsf{jai}}\}$	0.2
$I_4 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{edb}}, a_{\mathsf{jdb}}\}$	0.1
$I_5 = \{r_{\mathsf{e}}, a_{\mathsf{edb}}\}$	0.1

Possible *UArea* facts

u_{ir}	*UArea*(UCSD, $\perp_1$, IR)
u_{ai}	*UArea*(UCSD, $\perp_2$, AI)
u_{db}	*UArea*(UCSD, $\perp_3$, DB)

Target p-instance $\tilde{\mathcal{J}}_1$

$J_1 = \{u_{\mathsf{db}}, u_{\mathsf{ir}}\}$	0.3
$J_2 = \{u_{\mathsf{ai}}, u_{\mathsf{ir}}\}$	0.3
$J_3 = \{u_{\mathsf{ai}}, u_{\mathsf{db}}\}$	0.2
$J_4 = \{u_{\mathsf{db}}\}$	0.2

Target p-instance $\tilde{\mathcal{J}}_2$

$J_5 = \{u_{\mathsf{db}}, u_{\mathsf{ir}}\}$	0.35
$J_6 = \{u_{\mathsf{ai}}, u_{\mathsf{db}}, u_{\mathsf{ir}}\}$	0.45
$J_7 = \{u_{\mathsf{ai}}, u_{\mathsf{ir}}\}$	0.2

Target p-instance $\tilde{\mathcal{J}}_3$

$J_8 = \{u_{\mathsf{db}}, u_{\mathsf{ir}}\}$	0.3
$J_9 = \{u_{\mathsf{ai}}, u_{\mathsf{db}}\}$	0.3
$J_{10} = \{u_{\mathsf{ai}}, u_{\mathsf{ir}}\}$	0.4

Fig. 1. Source and target p-instances for the schema mapping $\mathcal{M}$ of Example 3.1.

by considering two natural and desirable requirements for a notion of a solution and showing that each of these requirements actually characterizes a p-solution.

3.1. Probabilistic Match

Our notion of a probabilistic match between p-spaces is based on the classical concept of joint (or bivariate) probability spaces with specified marginals [Fréchet 1951; Morgenstern 1956]. Our new twist on this old notion is that we require the joint distribution to have a support contained in a given binary relation.

Definition 3.2 (*Probabilistic Match*). Let $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ be two p-spaces and let $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ be a binary relation. A *probabilistic match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ with respect to R* (or, for short, an *R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$*) is a p-space $\tilde{\mathcal{P}}$ over $\Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ that satisfies the following two conditions.

(1) The left and right marginals of $\tilde{\mathcal{P}}$ are $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$, respectively. That is,
   (i) $\sum_{w \in \Omega(\tilde{\mathcal{W}})} p_{\tilde{\mathcal{P}}}(u, w) = p_{\tilde{\mathcal{U}}}(u)$ for all $u \in \tilde{\mathcal{U}}$, *and*
   (ii) $\sum_{u \in \Omega(\tilde{\mathcal{U}})} p_{\tilde{\mathcal{P}}}(u, w) = p_{\tilde{\mathcal{W}}}(w)$ for all $w \in \tilde{\mathcal{W}}$.
(2) The support of $\tilde{\mathcal{P}}$ is contained in R (i.e., $\Pr(\mathcal{P} \in R) = 1$).

Note that an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ can be viewed as a probability space over R, whose marginals coincide with $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$. A special case of a probabilistic match is the *product space* $\tilde{\mathcal{U}} \times \tilde{\mathcal{W}}$, where R is the set $\Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ and the two coordinates are probabilistically independent (that is, $p_{\tilde{\mathcal{U}} \times \tilde{\mathcal{W}}}(u, w) = p_{\tilde{\mathcal{U}}}(u) \cdot p_{\tilde{\mathcal{W}}}(w)$ for all $u \in \tilde{\mathcal{U}}$ and $w \in \tilde{\mathcal{W}}$). Two other special cases, for a relation $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$, are the following.

—An R-match $\tilde{\mathcal{P}}$ is *left-trivial* if for every $u \in \Omega_+(\tilde{\mathcal{U}})$ there is exactly one $w \in \Omega(\tilde{\mathcal{W}})$ such that $p_{\tilde{\mathcal{P}}}(u, w) > 0$; equivalently, $\Pr_{\tilde{\mathcal{P}}}(u, w) = \Pr_{\tilde{\mathcal{U}}}(u)$ whenever $\Pr_{\tilde{\mathcal{P}}}(u, w) > 0$.
—Similarly, $\tilde{\mathcal{P}}$ is *right-trivial* if for every $w \in \Omega_+(\tilde{\mathcal{W}})$ there is exactly one $u \in \Omega(\tilde{\mathcal{U}})$ such that $p_{\tilde{\mathcal{P}}}(u, w) > 0$; equivalently, $\Pr_{\tilde{\mathcal{P}}}(u, w) = \Pr_{\tilde{\mathcal{W}}}(w)$ whenever $\Pr_{\tilde{\mathcal{P}}}(u, w) > 0$.

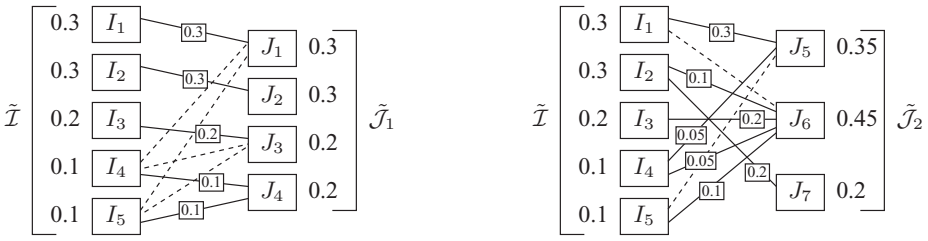

Fig. 2. $\mathsf{SOL}_{\mathcal{M}}$-matches of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_1$ and of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_2$ for the source p-instance $\tilde{\mathcal{I}}$ and the target p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ of Example 3.1.

Example 3.3. Consider the p-instances $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ of Figure 1. The two bipartite graphs of Figure 2 depict (finite) probabilistic matches of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_1$ (in the graph on the left side of Figure 2) and in $\tilde{\mathcal{J}}_2$ (in the graph on the right side of Figure 2). The relations R are the ones given by the (solid and dashed) edges that connect each of the two pairs of p-spaces. The probability of each pair (I, J) is written inside a rectangular box on the corresponding edge, unless this probability is zero and then the edge is represented as a dashed line. (Recall that the probability of (I, J) is necessarily zero if no edge connects I and J.)

Observe that the probabilistic match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_1$ on the left side of Figure 2 is left-trivial, since every node on the left side is incident to exactly one nonzero edge. Thus, it is immediate to verify that Item (i) in Part 1 of Definition 3.2 holds. Note that this match is not right-trivial (since J_4 is incident to two nonzero edges). Actually, there cannot be any right-trivial match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_1$, simply because $\Omega_+(\tilde{\mathcal{J}}_1)$ contains fewer samples than $\Omega_+(\tilde{\mathcal{I}})$.

A more complex example of a probabilistic match is the match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_2$ on the right side of Figure 2. Note that this match is neither left-trivial nor right-trivial. Consider the instance I_4 in the right side of Figure 2. In Item (i) of Part 1 of Definition 3.2, when the role of u is played by I_4, the sum on the left side, which is the sum of probabilities of the edges adjacent to I_4, is $0.05 + 0.05 = 0.1$, which is exactly the probability of I_4, which is the value on the right side. Consider now the instance J_6 in the right side of Figure 2. In Item (ii) of Part 1 of Definition 3.2, when the role of w is played by J_6, the sum on the left side, which is the sum of probabilities of the edges adjacent to J_6, is $0 + 0.1 + 0.2 + 0.05 + 0.1 = 0.45$, which is exactly the probability of J_6, which is the value on the right side.

3.2. p-Solution

We are now ready to define the concept of a p-solution. For a schema mapping $\mathcal{M}$, we denote by $\mathsf{SOL}_{\mathcal{M}}$ the binary relation that comprises pairs $(I, J) \in \mathsf{Inst}^{\mathsf{c}}(\mathbf{S}) \times \mathsf{Inst}(\mathbf{T})$, such that J is a solution for I.

Definition 3.4 (*p-Solution*). Let $\mathcal{M}$ be a schema mapping and let $\tilde{\mathcal{I}}$ be a source p-instance. A *p-solution* (for $\tilde{\mathcal{I}}$ *with respect to* Σ) is a target p-instance $\tilde{\mathcal{J}}$, such that there is a $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$.

Note that by a $\mathsf{SOL}_{\mathcal{M}}$-match we mean, of course, an R-match where the role of R is played by $\mathsf{SOL}_{\mathcal{M}}$.

Example 3.5. Consider again the schema mapping $\mathcal{M}$ of Example 3.1 and the source and target p-instances that are depicted in Figure 1 and described in Example 3.1. Figure 2 shows two $\mathsf{SOL}_{\mathcal{M}}$-matches of $\tilde{\mathcal{I}}$: the one on the left side is in $\tilde{\mathcal{J}}_1$, and the one on the right side is in $\tilde{\mathcal{J}}_2$. For example, there are edges from I_4 to J_1, J_3, and J_4, since J_1, J_3, and J_4 are each solutions for I_4 with respect to Σ (the edges from I_4 to J_1 and

J_3 each have probability 0, which is allowed). There is no edge from I_4 to J_2, since J_2 is not a solution for I_4 with respect to Σ. Thus, both $\tilde{J}_1$ and $\tilde{J}_2$ are p-solutions. Later, we will show that $\tilde{J}_3$ is not a p-solution (i.e., there is no $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{I}$ in $\tilde{J}_3$).

Defining a p-solution by means of a $\mathsf{SOL}_{\mathcal{M}}$-match is a straightforward application of the probabilistic-match mechanism. Next, we give a semantic justification to this definition. We start with an example. Consider the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and the source and target p-instances of Example 3.1. As shown in Examples 3.3 and 3.5, $\tilde{J}_1$ and $\tilde{J}_2$ are p-solutions. One may claim that $\tilde{J}_3$ should be deemed a p-solution as well (even though we later show that it is not) due to the following easily verifiable statement. For each sample I of $\tilde{I}$ (which has the probability $\Pr_{\tilde{I}}(I)$ of being the selected instance), there is a probability of $\Pr_{\tilde{I}}(I)$, or even higher, that a sample of $\tilde{J}$ is a solution for I. Next, we show that this property is not enough, and moreover, that $\tilde{J}_3$ should *not* be a p-solution.

For an arbitrary target p-instance $\tilde{J}$, let $p_{\mathrm{db}}(\tilde{J})$ be the probability that, in $\tilde{J}$, database (DB) research is done in UCSD. The source p-instance $\tilde{I}$ says that there is a probability of 0.7 that at least one researcher of UCSD is in the DB area (as obtained by summing up the probabilities of all the instances that contain a_{edb}, a_{jdb} or both). By the schema mapping $\mathcal{M}$, we would like a p-solution $\tilde{J}$ to say that DB research is done in UCSD with a probability of 0.7, that is, $p_{\mathrm{db}}(\tilde{J}) = 0.7$. Moreover, since Σ allows DB research at UCSD even if the source does not contain a DB researcher at UCSD, we should allow $p_{\mathrm{db}}(\tilde{J})$ to be larger than 0.7, in addition to allowing it to equal 0.7. Now, $p_{\mathrm{db}}(\tilde{J}_1)$ is exactly 0.7 and $p_{\mathrm{db}}(\tilde{J}_2)$ is 0.8, as desired. However, this is not the case for $\tilde{J}_3$, since $p_{\mathrm{db}}(\tilde{J}_3) = 0.6$.

To generalize this example, consider a schema mapping $\mathcal{M}$, a source p-instance $\tilde{I}$ and an event E of $\tilde{I}$ (e.g., the event "one or more researchers are in the DB area in UCSD," which means that a_{edb} or a_{jdb} or both are in the source instance). We say that a target instance J is *consistent* with E if J is a solution for at least one instance I of E. Then, as illustrated above, the following property is desired from a p-solution $\tilde{J}$. For all events E of $\tilde{I}$, the probability that $\tilde{J}$ is consistent with E is at least the probability of E. An analogous desired property is the following. For all events F of $\tilde{J}$ (e.g., the event "the DB area in UCSD is nonempty", which means that u_{db} is in the target instance), the probability that a random instance of $\tilde{I}$ has a solution in F is at least the probability of F. It can rather easily be shown that the existence of a $\mathsf{SOL}_{\mathcal{M}}$-match guarantees these two properties. Rather surprisingly, each of the two properties implies the existence of a $\mathsf{SOL}_{\mathcal{M}}$-match; thus, as shown in the next theorem, each of the two is a characterization of a p-solution. (Recall from Section 2.3 that for a probability space $\tilde{J}$, we denote by $\mathcal{J}$, without the tilde sign, the random variable representing a sample of $\tilde{J}$.)

THEOREM 3.6. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. Let $\tilde{I}$ be a source p-instance and let $\tilde{J}$ be a target p-instance. The following are equivalent:*

(1) *$\tilde{J}$ is a p-solution (that is, a $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{I}$ in $\tilde{J}$ exists).*
(2) *For all $E \subseteq \mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$,*

$$\Pr_{\tilde{J}}\left(\bigvee_{I \in E} \langle I, \mathcal{J}\rangle \models \Sigma\right) \geq \Pr_{\tilde{I}}(E)\,.$$

(3) *For all $F \subseteq \mathsf{Inst}(\mathbf{T})$,*

$$\Pr_{\tilde{I}}\left(\bigvee_{J \in F} \langle \mathcal{I}, J\rangle \models \Sigma\right) \geq \Pr_{\tilde{J}}(F)\,.$$

Note that, following the previous discussion about $\tilde{\mathcal{J}}_3$, the fact that Part 2 of Theorem 3.6 is necessary for being a p-solution shows that $\tilde{\mathcal{J}}_3$ is not a p-solution for $\tilde{\mathcal{I}}$ (by using the event E saying that there is a DB researcher in UCSD). Theorem 3.6 is proved via the following characterization of the existence of a probabilistic match in the spirit of Hall's Marriage Theorem [Hall 1935]. Note that $R(u, \mathcal{W})$ is a (conventional) shorthand notation for $(u, \mathcal{W}) \in R$.

LEMMA 3.7. *Let $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ be two p-spaces and let $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ be a binary relation. There exists an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ if and only if for all events U of $\tilde{\mathcal{U}}$ it holds that $\Pr_{\tilde{\mathcal{U}}}(U) \leq \Pr_{\tilde{\mathcal{W}}}(\bigvee_{u \in U} R(u, \mathcal{W}))$.*

PROOF. We first prove the "only if" direction, and then the "if" direction.

The "only if" Direction. Let $\tilde{\mathcal{P}}$ be an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ and let U be an event of $\tilde{\mathcal{U}}$. Let W be the set of all samples w of $\tilde{\mathcal{W}}$, such that there exists $u \in U$ where $(u, w) \in R$. We need to show that $\Pr_{\tilde{\mathcal{W}}}(W) \geq \Pr_{\tilde{\mathcal{U}}}(U)$. Since $\tilde{\mathcal{U}}$ is the marginal distribution on the left side of $\tilde{\mathcal{P}}$, it follows that $\Pr_{\tilde{\mathcal{U}}}(U)$ is the probability that a random sample (u, w) of $\tilde{\mathcal{P}}$ satisfies $u \in U$. However, if (u, w) satisfies $u \in U$, then it also satisfies $w \in W$. Consequently, $\Pr_{\tilde{\mathcal{U}}}(U)$ is not larger than the probability that a random sample of $\tilde{\mathcal{P}}$ has a right component in W, namely, $\Pr_{\tilde{\mathcal{W}}}(W)$ (since $\tilde{\mathcal{W}}$ is a marginal distribution of $\tilde{\mathcal{P}}$). We conclude that $\Pr_{\tilde{\mathcal{W}}}(W) \geq \Pr_{\tilde{\mathcal{U}}}(U)$, as claimed.

The "if" Direction. We assume that $\Pr_{\tilde{\mathcal{U}}}(U) \leq \Pr_{\tilde{\mathcal{W}}}(\bigvee_{u \in U} R(u, \mathcal{W}))$ holds for all events U of $\tilde{\mathcal{U}}$. This condition, with U being $\Omega(\tilde{\mathcal{U}})$, implies that the support of $\tilde{\mathcal{W}}$ is contained in the right side of R. We will show that there exists an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ by using a recent version of the max-flow min-cut theorem for countable networks [Aharoni et al. 2011]. For that, we construct a (directed) network graph G as illustrated in Figure 3. In particular, the node set of G contains the set $\Omega(\tilde{\mathcal{U}})$, the set $\Omega(\tilde{\mathcal{W}})$, a source s and a target t. For all samples u of $\tilde{\mathcal{U}}$, there is an edge from s to u with the capacity $p_{\tilde{\mathcal{U}}}(u)$. Similarly, for all samples w of $\tilde{\mathcal{W}}$, there is an edge from w to t with the capacity $p_{\tilde{\mathcal{W}}}(w)$. Finally, for every pair (u, w) of R, there is an edge with an infinite capacity from u to w.

We will show that there is a flow $f : E(G) \to \mathbb{R}$ of size 1 from s to t. Formally, this means that the sum of flows $f(e)$ along the edges e that emanate from s is equal to 1; we denote this sum by $|f|$. Note that such a flow is the maximal possible flow. Moreover, in such a flow, all the edges that enter $\Omega(\tilde{\mathcal{U}})$ and all the edges that emanate from $\Omega(\tilde{\mathcal{W}})$ are saturated. Then, we define the p-space $\tilde{\mathcal{P}}$ over $\Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ to be such that $p_{\tilde{\mathcal{P}}}(u, w)$ is $f(u, w)$ if $(u, w) \in R$, and zero if $(u, w) \notin R$. It can easily be verified that $\tilde{\mathcal{P}}$ is an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$. Next, we show that such f indeed exists.

For a flow f over G and two sets U and V of nodes of G, we denote by $|f(U, V)|$ the sum of flows along the edges from U to V. A *cut* of G is a pair (S, T), such that $s \in S$ and $t \in T$, and the two sets form a partition of the node set of G. A generalization of the max-flow min-cut theorem from finite to countable networks [Aharoni et al. 2011, Theorem 6.1] says that there exists a cut (S, T) and a flow f, such that for all edges e of G, if e is from S to T then $f(e)$ is the capacity of e, and if e is from T to S then $f(e) = 0$. So, let (S, T) and f be such cut and flow in our network G. We will show that $|f| \geq 1$; this is enough, since $|f| \leq 1$ always holds as we said previously. As illustrated in Figure 3, we define:

$$U^s \stackrel{\text{def}}{=} S \cap \Omega(\tilde{\mathcal{U}}) \quad U^t \stackrel{\text{def}}{=} T \cap \Omega(\tilde{\mathcal{U}}) \quad W^s \stackrel{\text{def}}{=} S \cap \Omega(\tilde{\mathcal{W}}) \quad W^t \stackrel{\text{def}}{=} T \cap \Omega(\tilde{\mathcal{W}})$$

The definition of $|f|$ implies that

$$|f| = |f(\{s\}, U^t)| + |f(\{s\}, U^s)| \,. \tag{1}$$

Since f saturates the cut (S, T), we have

$$|f(\{s\}, U^t)| = \Pr_{\tilde{\mathcal{U}}}\big(U^t\big) \,. \tag{2}$$

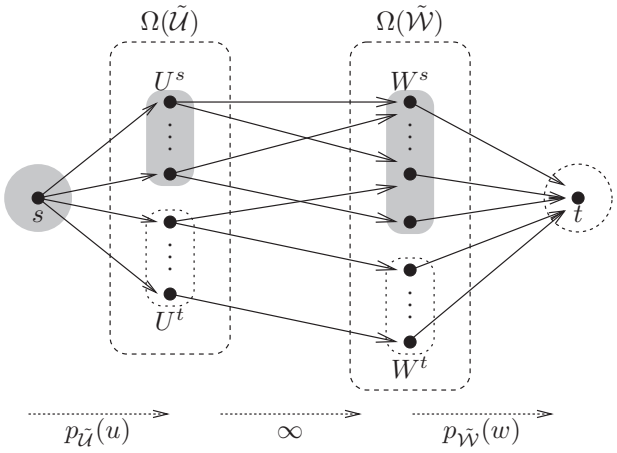

Fig. 3. The graph G.

Now, observe the following. If U^s and W^t contain samples u and w, respectively, such that $R(u, w)$ holds, then for $e = (u, w)$ we have $f(e) = \infty$ (since f saturates the edge e), which is clearly impossible (since only a finite flow enters u). So, W^s contains all the samples w of $\tilde{\mathcal{W}}$ such that $R(u, w)$ holds for some u of U^s. Therefore, since f is a flow, we have $|f(\{s\}, U^s)| = |f(U^s, W^s)|$. Recall that f is zero on the edges from U^t to W^s; therefore, $|f(U^s, W^s)| = |f(W^s, \{t\})|$. Also, recall that f is saturated on the edges from W^s to $\{t\}$, and hence $|f(W^s, t)| = \Pr_{\tilde{\mathcal{W}}}(W^s)$. Therefore, we have

$$|f(\{s\}, U^s)| = \Pr_{\tilde{\mathcal{W}}}(W^s)\,. \tag{3}$$

From the "if" condition, it follows that

$$\Pr_{\tilde{\mathcal{W}}}(W^s) \geq \Pr_{\tilde{\mathcal{U}}}(U^s) \tag{4}$$

(since W^s contains all the samples w of $\tilde{\mathcal{W}}$ such that $R(u, w)$ holds for some u of U^s).

Finally, by combining (1), (2), and (4), we conclude that $|f| \geq \Pr_{\tilde{\mathcal{U}}}(U^t) + \Pr_{\tilde{\mathcal{U}}}(U^s) = 1$, as claimed. □

4. UNIVERSAL P-SOLUTIONS AND QUERY ANSWERING

In this section, we generalize the concepts of a universal solution, and that of answering target queries, to the probabilistic setting.

4.1. Universal p-Solutions

Recall that the notion of a probabilistic match provides a systematic way of extending any binary relationship between (deterministic) objects to a relationship between probability spaces thereof. In the case of universal solutions, this is applied as follows. Consider a schema mapping $\mathcal{M}$. Denote by $\mathsf{USOL}_{\mathcal{M}}$ the relationship between pairs I and J of (ordinary) source and target instances, respectively, such that $\mathsf{USOL}_{\mathcal{M}}(I, J)$ holds if and only if J is a universal solution for I. Then, a *universal p-solution* is defined as follows.

Definition 4.1 (*Universal p-Solution*). Let $\mathcal{M}$ be a schema mapping. Let $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$ be source and target p-instances, respectively. We say that $\tilde{\mathcal{J}}$ is a *universal p-solution* (*for $\tilde{\mathcal{I}}$ with respect to* Σ) if there is a $\mathsf{USOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$.

Example 4.2. The $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_1$ (where $\mathcal{M}$, $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}_1$ are described in Example 3.1) on the left side of Figure 2 is actually a $\mathsf{USOL}_{\mathcal{M}}$-match, since an edge from I_m to J_n has a nonzero probability only if J_n is a universal solution for I_m. Thus, $\tilde{\mathcal{J}}_1$ is a universal p-solution for $\tilde{\mathcal{I}}$. The $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_2$ on the right side of Figure 2 is not a $\mathsf{USOL}_{\mathcal{M}}$-match since, for example, there is an edge (with probability 0.1) between I_2 and J_6, yet J_6 is not a universal solution for I_2. Later, we will show that $\tilde{\mathcal{J}}_2$ is, indeed, *not* a universal p-solution for $\tilde{\mathcal{I}}$.

We now give a simple proposition about the existence of a p-solution and of a universal p-solution.

PROPOSITION 4.3. *Let $\mathcal{M}$ be a schema mapping and let $\tilde{\mathcal{I}}$ be a source p-instance. A p-solution exists if and only if a solution exists for all $I \in \Omega_+(\tilde{\mathcal{I}})$. Similarly, a universal p-solution exists if and only if a universal solution exists for all $I \in \Omega_+(\tilde{\mathcal{I}})$.*

PROOF. The proposition states two special cases of the following general statement. Let U and W be two countable sets, let R be a subset of $U \times W$, and let $\tilde{\mathcal{U}}$ be a p-space over U. There exists a p-space $\tilde{\mathcal{W}}$ over W with an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ if and only if for every $u \in \Omega_+(\tilde{\mathcal{U}})$ there exists $w \in W$, such that $(u, w) \in R$. So let us prove this statement.

For the "if" direction, suppose that $\tilde{\mathcal{P}}$ is an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ for some p-space $\tilde{\mathcal{W}}$ over W, and let $u \in \Omega_+(\tilde{\mathcal{U}})$ be given. Since the left marginal of $\tilde{\mathcal{P}}$ is $\tilde{\mathcal{U}}$, we get that $\Omega_+(\tilde{\mathcal{P}})$ must contain the pair (u, w) for some $w \in W$. But the support of $\tilde{\mathcal{P}}$ is contained in R, and hence, (u, w) is in R as well.

For the "only if" direction, suppose that for every $u \in \Omega_+(\tilde{\mathcal{U}})$ there exists $w \in W$, such that $(u, w) \in R$. We will construct p-space $\tilde{\mathcal{W}}$ over W and a left-trivial R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$. For each $u \in \Omega_+(\tilde{\mathcal{U}})$, we choose an arbitrary member w_u of W, such that $(u, w_u) \in R$. The p-space $\tilde{\mathcal{W}}$ will assign to each $w \in W$ the probability that is the sum of the probabilities $p_{\tilde{\mathcal{U}}}(u)$ over all the elements u, such that $w_u = w$. The R-match $\tilde{\mathcal{P}}$ will assign to each pair of the form (u, w_u) the probability $p_{\tilde{\mathcal{U}}}(u)$, and to each other pair the probability 0. The reader can easily verify that $\tilde{\mathcal{W}}$ is indeed a p-space, and that $\tilde{\mathcal{P}}$ is indeed an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$. □

The proof of Proposition 4.3 shows that when a p-solution for a source p-instance $\tilde{\mathcal{I}}$ exists, there is a straightforward construction of a p-solution that is left-trivial. A similar comment holds for universal p-solutions.

In the deterministic case, a universal solution is deemed a good choice of a solution, since it is a most general one, where the notion of generality is defined by means of a homomorphism; that is, J_1 *generalizes* J_2 if $J_1 \to J_2$. We would like to have a similar characterization of a universal p-solution. For that, we need a notion for a relationship between p-instances that corresponds to that of homomorphism in ordinary data. One such definition can be obtained by applying the probabilistic match. Let $\mathbf{T}$ be a schema. We denote by $\mathsf{HOM}_{\mathbf{T}}$ the binary relation that includes all the pairs $(J_1, J_2) \in (\mathsf{Inst}(\mathbf{T}))^2$, such that $J_1 \to J_2$. Consider two p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ over $\mathbf{T}$. We use $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$ to denote that there is a $\mathsf{HOM}_{\mathbf{T}}$-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$.

Example 4.4. Consider again the p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ of Figure 1. A $\mathsf{HOM}_{\mathbf{T}}$-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$ is shown in Figure 4. Hence, since such a match exists, $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$ holds. Note that in this example, homomorphism is the same as containment (i.e., $J_i \to J_j$ if and only if $J_i \subseteq J_j$). But of course, homomorphism may exist without containment, and this is indeed the case in our later examples.

Remark 4.5. The definition of $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$, restricted to finite p-instances, is similar yet different from that of homomorphism given in Dong et al. [2009] where, in our

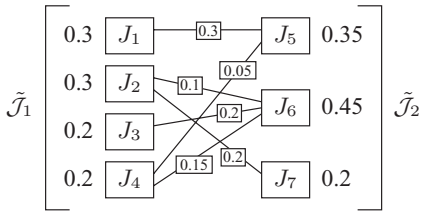

Fig. 4. A $\mathsf{HOM}_{\mathbf{T}}$-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$.

terminology, only right-trivial $\mathsf{HOM}_{\mathbf{T}}$-matches are allowed (in particular, there is no homomorphism from $\tilde{\mathcal{J}}_1$ to $\tilde{\mathcal{J}}_2$ in the sense of Dong et al. [2009] if the cardinality of $\Omega_+(\tilde{\mathcal{J}}_1)$ is strictly larger than that of $\Omega_+(\tilde{\mathcal{J}}_2)$).

The $\mathsf{HOM}_{\mathbf{T}}$-match extends the notion of homomorphism to p-instances in the systematic way of applying the probabilistic match. There are, though, other natural ways of generalizing this notion. Next, we consider two such ways, which are based on the classical notion of a stochastic order. We then explore their relationships to the existence of a $\mathsf{HOM}_{\mathbf{T}}$-match. First, we need some definitions.

A stochastic order is traditionally an order over numeric random variables (cf. Shaked and Shanthikumar [1994]). Here, we extend this notion from numbers to general preordered elements, in a straightforward manner. Formally, let O be a countable set and let $\preceq$ be a preorder over O (i.e., $\preceq$ is a reflexive and transitive binary relation $\preceq$ over O). We define the *stochastic extension* of $\preceq$ as the preorder $\preceq'$ over the set of all the p-spaces over O, where for all p-spaces $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$, the interpretation of $\tilde{\mathcal{U}} \preceq' \tilde{\mathcal{W}}$ is

$$\forall o \in O \left(\Pr(\mathcal{U} \preceq o) \geq \Pr(\mathcal{W} \preceq o)\right) .$$

Let $\mathbf{T}$ be a schema. It is well known that the existence-of-a-homomorphism relationship can be viewed as a preorder over $\mathsf{Inst}(\mathbf{T})$ (see, e.g., Hell and Nešetřil [2004]), and there are basically two ways to define this preorder. In the first, we use the preorder $\preceq_{\mathsf{sp}}$, where $J \preceq_{\mathsf{sp}} J'$ is interpreted as $J \to J'$, namely, "J is at most as specific as J'." The second preorder, $\preceq_{\mathsf{ge}}$, has the complementary interpretation: "J is at most as general as J'," that is, $J \preceq_{\mathsf{ge}} J'$ means $J' \to J$. Having the two preorders $\preceq_{\mathsf{sp}}$ and $\preceq_{\mathsf{ge}}$ over instances, we automatically obtain two preorders over p-instances, namely, the stochastic extensions, which we denote by $\overset{\preceq_{\mathsf{sp}}}{\longrightarrow}$ (with a forward arrow) and $\overset{\preceq_{\mathsf{ge}}}{\longleftarrow}$ (with a backward arrow), respectively.[3] Thus, $\tilde{\mathcal{J}}_1 \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}_2$ if $\Pr(\mathcal{J}_1 \to J) \geq \Pr(\mathcal{J}_2 \to J)$ for all instances J over $\mathbf{T}$ (where $\Pr(\mathcal{J} \to J)$ is the probability that there is a homomorphism from a random instance of $\tilde{\mathcal{J}}$ to J), and $\tilde{\mathcal{J}}_2 \overset{\preceq_{\mathsf{ge}}}{\longleftarrow} \tilde{\mathcal{J}}_1$ if $\Pr(J \to \mathcal{J}_2) \geq \Pr(J \to \mathcal{J}_1)$ for all instances J over $\mathbf{T}$. For uniformity of presentation, we write $\tilde{\mathcal{J}}_1 \overset{\succeq_{\mathsf{ge}}}{\longrightarrow} \tilde{\mathcal{J}}_2$ instead of $\tilde{\mathcal{J}}_2 \overset{\preceq_{\mathsf{ge}}}{\longleftarrow} \tilde{\mathcal{J}}_1$.

Example 4.6. Let $\mathbf{T}$ be the schema that contains a single unary relation R. Figure 5 shows two p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ over $\mathbf{T}$ (note that $\perp_1$ is a labeled null). We first show that $\tilde{\mathcal{J}}_1 \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}_2$. So, let J be an instance over $\mathbf{T}$; we need to show that $\Pr(\mathcal{J}_1 \to J) \geq \Pr(\mathcal{J}_2 \to J)$. We consider several cases.

—If J contains neither $R(a)$ nor $R(b)$, then $\Pr(\mathcal{J}_2 \to J) = 0$.
—If J contains either $R(a)$ or $R(b)$ but not both, then $J_1 \to J$ and either $J_3 \to J$ or $J_4 \to J$, but not both. Therefore, in this case $\Pr(\mathcal{J}_1 \to J) = 0.5 = \Pr(\mathcal{J}_2 \to J)$.
—If J contains both $R(a)$ and $R(b)$, then $\Pr(\mathcal{J}_1 \to J) = 1 = \Pr(\mathcal{J}_2 \to J)$.

[3]The choice of the notation $\overset{\preceq_{\mathsf{sp}}}{\longrightarrow}$ and $\overset{\preceq_{\mathsf{ge}}}{\longleftarrow}$ (rather than, e.g., $\preceq'_{\mathsf{sp}}$ and $\preceq'_{\mathsf{ge}}$) is for clarity of presentation.

$\tilde{\mathcal{J}}_1$	
0.5	$J_1 = \{R(\perp_1)\}$
0.5	$J_2 = \{R(a), R(b)\}$

$\tilde{\mathcal{J}}_2$	
0.5	$J_3 = \{R(a)\}$
0.5	$J_4 = \{R(b)\}$

Fig. 5. p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$.

On the other hand, $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq ge} \tilde{\mathcal{J}}_2$ does not hold since, for example, $\tilde{\mathcal{J}}_2$ does not include any instance J with $J_2 \to J$, and hence, $\Pr_{\tilde{\mathcal{J}}_1}(J_2 \to \mathcal{J}_1) > 0$ while $\Pr_{\tilde{\mathcal{J}}_2}(J_2 \to \mathcal{J}_2) = 0$.

Next, we will show that $\tilde{\mathcal{J}}_2 \xrightarrow{\succeq ge} \tilde{\mathcal{J}}_1$. So, let J be an instance over $\mathbf{T}$; we need to show that $\Pr(J \to \mathcal{J}_1) \geq \Pr(J \to \mathcal{J}_2)$. Again, we consider several cases.

—If neither $J \to J_3$ nor $J \to J_4$ holds, then $\Pr(J \to \mathcal{J}_2) = 0$.
—If exactly one of $J \to J_3$ and $J \to J_4$ holds, then $\Pr(J \to \mathcal{J}_2) = 0.5$. In this case, we have $J \to J_2$, and hence, $\Pr(J \to \mathcal{J}_1) \geq 0.5$.
—If both $J \to J_3$ and $J \to J_4$ holds, then it is easy to show that both $J \to J_1$ and $J \to J_2$ hold. Therefore, $\Pr(J \to \mathcal{J}_1) = 1 = \Pr(J \to \mathcal{J}_2)$.

On the other hand, $\tilde{\mathcal{J}}_2 \xrightarrow{\preceq sp} \tilde{\mathcal{J}}_1$ is false, since there are homomorphisms from none of the instances of $\tilde{\mathcal{J}}_2$ to J_1 (hence, we get a contradiction to $\tilde{\mathcal{J}}_2 \xrightarrow{\preceq sp} \tilde{\mathcal{J}}_1$ by using $J = J_1$).

Later on, we will give a more systematic way of testing for the satisfiability of $\xrightarrow{\preceq sp}$ and $\xrightarrow{\succeq ge}$ (Lemma 4.7). We will also show (in Theorem 4.9) that $\tilde{\mathcal{J}} \xrightarrow{mat} \tilde{\mathcal{J}}'$ implies both $\tilde{\mathcal{J}} \xrightarrow{\preceq sp} \tilde{\mathcal{J}}'$ and $\tilde{\mathcal{J}} \xrightarrow{\succeq ge} \tilde{\mathcal{J}}'$; hence, for this $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ we get that neither $\tilde{\mathcal{J}}_1 \xrightarrow{mat} \tilde{\mathcal{J}}_2$ nor $\tilde{\mathcal{J}}_2 \xrightarrow{mat} \tilde{\mathcal{J}}_1$ holds.

We now have three ways of extending the relationship $J_1 \to J_2$ (existence of a homomorphism) from instances J_1 and J_2 to p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$. The first is $\tilde{\mathcal{J}}_1 \xrightarrow{mat} \tilde{\mathcal{J}}_2$, namely, there exists a $\mathsf{HOM}_{\mathbf{T}}$-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$. The second is $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq sp} \tilde{\mathcal{J}}_2$, namely, $\tilde{\mathcal{J}}_1$ is at most as specific as $\tilde{\mathcal{J}}_2$. The third is $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq ge} \tilde{\mathcal{J}}_2$, namely, $\tilde{\mathcal{J}}_1$ is at least as general as $\tilde{\mathcal{J}}_2$. Observe that the three are indeed extensions of $\to$, in the following sense. If $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ are deterministic instances J_1 and J_2 (i.e., the probability of J_i in $\tilde{\mathcal{J}}_i$ is 1, for $i = 1, 2$), then each of $\tilde{\mathcal{J}}_1 \xrightarrow{mat} \tilde{\mathcal{J}}_2$, $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq sp} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq ge} \tilde{\mathcal{J}}_2$ is equivalent to $J_1 \to J_2$.

In Example 4.6, we showed that neither $\xrightarrow{\preceq sp}$ nor $\xrightarrow{\succeq ge}$ implies the other. Next, we further explore the connection among the three relationships $\xrightarrow{mat}$, $\xrightarrow{\preceq sp}$ and $\xrightarrow{\succeq ge}$. For that, some notation and intermediate results are needed. We start with characterizations of the relationships $\xrightarrow{\preceq sp}$ and $\xrightarrow{\succeq ge}$.

Let J_1 and J_2 be two instances over the same schema. The instance $J_1 \sqcap J_2$ is obtained by joining every two facts $R(t_1, \ldots, t_n) \in J_1$ and $R(u_1, \ldots, u_n) \in J_2$ into the fact $R(f(t_1, u_1), \ldots, f(t_n, u_n))$, where the function f is defined as follows.

—If $x \in \mathsf{Const}$, then $f(x, x) = x$;
—If $x \neq y$ or either x or y is a null, then $f(x, y)$ is a unique null (thus, f is a one-to-one function).

As an example, if $J_1 = \{R(a, b), S(b), S(c)\}$ and $J_2 = \{R(a, c), S(\perp_1), S(c)\}$, then $J_1 \sqcap J_2$ is as follows (we denote $f(x, y)$ by $\perp_{xy}$ whenever $f(x, y)$ is a null):

$$J_1 \sqcap J_2 = \{R(a, \perp_{bc}), S(\perp_{b\perp_1}), S(\perp_{bc}), S(\perp_{c\perp_1}), S(c)\}$$

It is rather easy to verify that the operation $\sqcap$ is commutative and associative up to isomorphism. If $O = \{J_1, \ldots, J_k\}$, then $\sqcap O$ denotes the instance $J_1 \sqcap \cdots \sqcap J_k$. The instance $\cup O$ is the one that contains the facts of all the relations of O.

LEMMA 4.7. *Let $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ be two p-instances over the same schema and suppose (without loss of generality) that different samples of $\tilde{\mathcal{J}}_2$ do not share common nulls. Then,*

(1) $\tilde{\mathcal{J}}_1 \stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ *if and only if for all* $O \subseteq \Omega(\tilde{\mathcal{J}}_2)$,

$$\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)\,.$$

(2) $\tilde{\mathcal{J}}_1 \stackrel{\succeq ge}{\Longrightarrow} \tilde{\mathcal{J}}_2$ *if and only if for all* $O \subseteq \Omega(\tilde{\mathcal{J}}_1)$,

$$\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O)\,.$$

PROOF. Throughout this proof, we assume that all the instances and p-instances are over the same schema. We begin with Part 1 and first prove the "only if" direction. Suppose that $\tilde{\mathcal{J}}_1 \stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ and let $O \subseteq \Omega(\tilde{\mathcal{J}}_2)$ be given. We need to show that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$. Now, since $J \to \cup O$ holds for all $J \in O$, we have that $\Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$. Consequently, since our assumption $\tilde{\mathcal{J}}_1 \stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ implies that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to \cup O)$, it follows that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$ also holds, as desired.

For the "if" direction, suppose that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$ for every $O \subseteq \Omega(\tilde{\mathcal{J}}_2)$, and let J be an instance. We need to show that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to J) \geq \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J)$. Let $O_J \subseteq \Omega(\tilde{\mathcal{J}}_2)$ be the set of all the instances J' of $\tilde{\mathcal{J}}_2$, such that $J' \to J$. From the assumption that the sets of nulls of the instances of $\tilde{\mathcal{J}}_2$ are pairwise disjoint it follows that $\cup O_J \to J$. So, from the fact that homomorphisms can be composed, we conclude that $\mathcal{J}_1 \to \cup O_J$ implies $\mathcal{J}_1 \to J$. Therefore,

$$\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to J) \geq \Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O_J)\,. \tag{5}$$

Now, the assumption implies that

$$\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O_J) \geq \Pr_{\tilde{\mathcal{J}}_2}(O_J)\,. \tag{6}$$

Finally, by the choice of O_J, we have that

$$\Pr_{\tilde{\mathcal{J}}_2}(O_J) = \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J)\,. \tag{7}$$

By combining (5), (6), and (7), we get that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to J) \geq \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J)$, as claimed.

We now prove Part 2 and begin with the "only if" direction. Suppose that $\tilde{\mathcal{J}}_1 \stackrel{\succeq ge}{\Longrightarrow} \tilde{\mathcal{J}}_2$ and consider a set $O \subseteq \Omega(\tilde{\mathcal{J}}_1)$. We need to show that $\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O)$. We will prove that $\sqcap O \to J$ holds for all $J \in O$. This is enough, since it then follows that $\Pr_{\tilde{\mathcal{J}}_1}(O) \leq \Pr_{\tilde{\mathcal{J}}_1}(\sqcap O \to \mathcal{J}_1)$ and, by our assumption, $\Pr_{\tilde{\mathcal{J}}_1}(\sqcap O \to \mathcal{J}_1) \leq \Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2)$, hence $\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O)$.

So, it suffices to prove the following claim. If J_1 and J_2 are instances, then there is a homomorphism $h : (J_1 \sqcap J_2) \to J_1$. (By induction, we then proceed to sets O with more than two members.) Let f be the one-to-one function that is used for constructing $J_1 \sqcap J_2$ (see the definition of $\sqcap$). So, we define h to be the projection of f over the first coordinate, that is, it maps $f(x, y)$ to x (since f is one-to-one, h is well defined). Note that, by definition, h preserves the facts and, moreover, h is identity on constants, since $f(x, y) \in \mathsf{Const}$ implies that $x = y = f(x, y)$.

For the "if" direction, suppose that $\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O)$ for every $O \subseteq \Omega(\tilde{\mathcal{J}}_1)$, and let J be an instance. We need to show that $\Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1) \leq \Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2)$. Let O_J be the set of all instances $J' \in \Omega(\tilde{\mathcal{J}}_1)$, such that $J \to J'$. We will first show that $J \to \sqcap O_J$ holds. For that, it is enough to show the following. For instances J, J_1 and J_2, if $J \to J_1$ and $J \to J_2$, then $J \to (J_1 \sqcap J_2)$. (By induction, we then proceed to sets O with more than two members.) Again, let f be the function used for constructing $J_1 \sqcap J_2$. Let $h_1 : J \to J_1$ and $h_2 : J \to J_2$ be given. The function h is defined by $h(z) = f(h_1(z), h_2(z))$. We now show that h is a homomorphism. First, note that h preserves facts, since if $R(t_1, \ldots, t_n) \in J$, then $R(h_1(t_1), \ldots, h_1(t_n)) \in J_1$ and $R(h_2(t_1), \ldots, h_2(t_n)) \in J_1$. It follows that $J_1 \sqcap J_2$ contains $R(f(h_1(t_1), h_2(t_1))), \ldots, f(h_1(t_n), h_2(t_n)))$. Observe that h also preserves constants since, for all constants $c \in \mathsf{Const}(J)$ it holds that $h(c) = f(h_1(c), h_2(c)) = f(c, c) = c$.

Now, since $J \to \sqcap O_J$, we have that $\sqcap O_J \to \mathcal{J}_2$ implies $J \to \mathcal{J}_2$ (since homomorphisms can be composed). Consequently,

$$\Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_2}(\sqcap O_J \to \mathcal{J}_2)\ . \tag{8}$$

By our assumption, it holds that

$$\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O_J \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O_J)\ . \tag{9}$$

From the definition of O_J, it follows that

$$\Pr_{\tilde{\mathcal{J}}_1}(O_J) = \Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1)\ . \tag{10}$$

Finally, $\Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1) \leq \Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2)$ is obtained by combining (8), (9), and (10). □

The next proposition shows that existence of a probabilistic match between two finite p-spaces is decidable in *strongly polynomial time* [Grötschel et al. 1988].

PROPOSITION 4.8. *Whether there exists an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$, given finite p-spaces $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ and a relation $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$, can be decided in strongly polynomial time.*

PROOF. Recall the proof of Lemma 3.7, and specifically the "if" direction for a finite R. There, we constructed from $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ a network graph G with a source node s and a target node t (see Figure 3), and showed how to obtain an R-match from a flow of size 1. The proof actually shows that such a flow exists if and only there is an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$. So, determining whether there exists an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ (efficiently) reduces to finding a maximal flow, and it is well known (e.g., Galil and Tardos [1986]) that a maximal flow can be found in strongly polynomial time. □

We can now proceed to prove the following theorem, showing that $\xrightarrow{\text{mat}}$ is a strictly stronger relationship than $\xrightarrow{\preceq\text{sp}}$ and $\xrightarrow{\succeq\text{ge}}$; that is, $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$ implies both $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_2$, and there are cases where neither of the opposite implications holds. Moreover, this theorem states that $\xrightarrow{\preceq\text{sp}}$ and $\xrightarrow{\succeq\text{ge}}$ are incomparable, which we already showed in Example 4.6 (and include in the theorem for presentation's sake). Finally, the theorem shows that for finite p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, testing $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$ and testing $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_2$ are not even in the same complexity class as testing $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$ (assuming NP $\neq$ coNP), since the first two tests are DP-hard,[4] yet decidable, while the third is NP-complete.

THEOREM 4.9. *The following holds.*

(1) *For all p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, if $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$ then $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_2$.*
(2) *There are p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, such that $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_1 \not\xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_2$; similarly, there are p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, such that $\tilde{\mathcal{J}}_1 \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_1 \not\xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$. Hence, due to Part 1, neither $\xrightarrow{\preceq\text{sp}}$ nor $\xrightarrow{\succeq\text{ge}}$ implies $\xrightarrow{\text{mat}}$.*
(3) *Testing $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$, given two finite p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, is in* NP*, and there is a schema over which this test is* NP*-complete.*
(4) *Testing each of $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$ and $\tilde{\mathcal{J}}_2 \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}_1$, given finite p-instances $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, is in EXPTIME and NEXPTIME, respectively, and there is a schema over which both tests are DP-hard.*

PROOF. We will prove each part separately.

Part 1. Suppose that $\tilde{\mathcal{J}}_1 \xrightarrow{\text{mat}} \tilde{\mathcal{J}}_2$. We first show that $\tilde{\mathcal{J}}_1 \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}_2$. Let J be an instance. We need to show that $\Pr_{\mathcal{J}_1}(\tilde{\mathcal{J}}_1 \to J) \geq \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J)$. Let O_J be the set of all samples

[4]Recall that DP is the class of problems that can be formed as a difference of two problems in NP [Papadimitriou and Yannakakis 1984]. In particular, if a problem is DP-hard, then it is both NP-hard and coNP-hard.

J' of $\tilde{\mathcal{J}}_2$, such that $J' \to J$. Then,

$$\Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J) = \Pr_{\tilde{\mathcal{J}}_2}(O_J). \tag{11}$$

From Lemma 3.7, it follows that

$$\Pr_{\tilde{\mathcal{J}}_2}(O_J) \leq \Pr_{\tilde{\mathcal{J}}_1}\left(\bigvee_{J' \in O_J} \mathcal{J}_1 \to J'\right). \tag{12}$$

But then, if $\vee_{J' \in O_J} J_1 \to J'$ holds for some $J_1 \in \Omega(\tilde{\mathcal{J}}_1)$, then $J_1 \to J$ also holds (since there is a homomorphism from each $J' \in O_J$ to J). Consequently,

$$\Pr_{\tilde{\mathcal{J}}_1}\left(\bigvee_{J' \in O_J} \mathcal{J}_1 \to J'\right) \leq \Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to J). \tag{13}$$

From (11), (12), and (13), we conclude that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to J) \geq \Pr_{\tilde{\mathcal{J}}_2}(\mathcal{J}_2 \to J)$, as desired.

Next, we prove that $\tilde{\mathcal{J}}_1 \overset{\succeq \mathsf{ge}}{\Longrightarrow} \tilde{\mathcal{J}}_2$. Let J be an instance. Now, we need to show that $\Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1) \leq \Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2)$. Let O_J be the set of all samples J' of $\tilde{\mathcal{J}}_1$, such that $J \to J'$. Then,

$$\Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1) = \Pr_{\tilde{\mathcal{J}}_1}(O_J). \tag{14}$$

From Part 2 of Lemma 3.7, it follows that

$$\Pr_{\tilde{\mathcal{J}}_1}(O_J) \leq \Pr_{\tilde{\mathcal{J}}_2}\left(\bigvee_{J' \in O_J} J' \to \mathcal{J}_2\right). \tag{15}$$

Since homomorphisms can be composed, it follows that for all instances $J'' \in \tilde{\mathcal{J}}_2$, if $J' \to J''$ holds for some $J' \in O_J$, then $J \to J''$ also holds (since $J \to J'$). Consequently,

$$\Pr_{\tilde{\mathcal{J}}_2}\left(\bigvee_{J' \in O_J} J' \to \mathcal{J}_2\right) \leq \Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2). \tag{16}$$

Then, $\Pr_{\tilde{\mathcal{J}}_1}(J \to \mathcal{J}_1) \leq \Pr_{\tilde{\mathcal{J}}_2}(J \to \mathcal{J}_2)$ follows from (14), (15), and (16), as required.

Part 2. This part has already been shown in Example 4.6.

Part 3. Membership in NP is proved as follows. Observe that, if $R \subseteq R'$, then every R-match is also an R'-match. Consequently, as evidence that $\tilde{\mathcal{J}}_1 \overset{\mathsf{mat}}{\Longrightarrow} \tilde{\mathcal{J}}_2$, one can provide a relation $R \subseteq \Omega(\tilde{\mathcal{J}}_1) \times \Omega(\tilde{\mathcal{J}}_2)$, such that $R \subseteq \mathsf{HOM}_{\mathbf{T}}$ (that is, every pair $(J_1, J_2) \in R$ is such that $J_1 \to J_2$), and an R-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$ exists. To show that R is indeed a subset of $\mathsf{HOM}_{\mathbf{T}}$, the evidence should also include a homomorphism $h : J_1 \to J_2$ for each $(J_1, J_2) \in R$. Finally, Proposition 4.8 shows that whether there is an R-match of $\tilde{\mathcal{J}}_1$ in $\tilde{\mathcal{J}}_2$ can be decided in polynomial time. For NP-hardness, recall that there are fixed schemas over which testing $\tilde{\mathcal{J}}_1 \overset{\mathsf{mat}}{\Longrightarrow} \tilde{\mathcal{J}}_2$ is NP-hard even if $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ are deterministic [Chandra and Merlin 1977].

Part 4. The EXPTIME and NEXPTIME upper bounds are obtained from Lemma 4.7, as follows.

—Building on Lemma 4.7, to decide whether $\tilde{\mathcal{J}}_1 \overset{\preceq \mathsf{sp}}{\Longrightarrow} \tilde{\mathcal{J}}_2$, we traverse over the (exponentially many) sets $O \subseteq \Omega(\tilde{\mathcal{J}}_2)$, and, for each O, we test whether $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$. To compute $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O)$, we sum up all the numbers $p_{\tilde{\mathcal{J}}_1}(J)$ over all the instances $J \in \tilde{\mathcal{J}}_1$, where $J \to \cup O$ (and $J \to \cup O$ can obviously be decided in exponential time).

—Building on Lemma 4.7, to decide whether $\tilde{\mathcal{J}}_2 \stackrel{\succeq ge}{\Longrightarrow} \tilde{\mathcal{J}}_1$, we traverse over the (exponentially many) sets $O \subseteq \Omega(\tilde{\mathcal{J}}_1)$, and, for each O, we test whether $\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2) \geq \Pr_{\tilde{\mathcal{J}}_1}(O)$. To compute $\Pr_{\tilde{\mathcal{J}}_2}(\sqcap O \to \mathcal{J}_2)$, we sum up all the numbers $p_{\tilde{\mathcal{J}}_2}(J)$ over all the instances $J \in \tilde{\mathcal{J}}_2$, where $\sqcap O \to J$. Now, from the definition of the operator $\sqcap$, it easily follows that $|\sqcap O| \leq \prod_{J' \in O} |J'|$, and, in particular, that $\sqcap O$ is at most exponentially larger then the input of the problem. Therefore, $\sqcap O \to J$ can be decided nondeterministically by guessing a homomorphism (of exponential size) from $\sqcap O$ to J.

Next, we prove that deciding $\tilde{\mathcal{J}}_1 \stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ is coNP-hard. Later, we show how to extend the proof to show DP-hardness. The proof is by reduction from the complement of the problem *Independent Set* (IndSet), which is the following. Given an undirected graph $G = (V(G), E(G))$ and a natural number k, determine whether there is a set $U \subseteq V(G)$ of k nodes, such that no two nodes of U are connected by an edge.

Let (G, k) be an instance of IndSet, and suppose that $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$. We assume that $m > 0$, since otherwise IndSet is trivial. The schema $\mathbf{T}$ contains one unary relation symbol V. The p-instance $\tilde{\mathcal{J}}_2$ contains the instance $\{V(i)\}$ for all $1 \leq i \leq n$. In addition, $\tilde{\mathcal{J}}_2$ contains the instance $J_{\text{all}} = \{V(i) \mid 1 \leq i \leq n\}$. The probability of each $\{V(i)\}$ is p_0, and that of J_{all} is $1 - np_0$. We later determine the value of p_0.

The p-instance $\tilde{\mathcal{J}}_1$ contains the instance $\{V(i), V(i')\}$ for all edges $\{v_i, v_{i'}\} \in E(G)$. The probability of each $\{V(i), V(i')\}$ is np_0. In addition, $\tilde{\mathcal{J}}_1$ contains the empty instance $J_\emptyset = \emptyset$ with the probability $(k-1)p_0$. Since $\tilde{\mathcal{J}}_1$ is a probability space, this defines the value of p_0:

$$mnp_0 + (k-1)p_0 = 1 \quad \Rightarrow \quad p_0 = \frac{1}{mn + k - 1}$$

Next, we prove that $\tilde{\mathcal{J}}_1 \stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ if and only if no independent set of size k exists. We first prove the "if" part (so we assume that every independent set has fewer than k nodes). Let $O \subseteq \Omega(\tilde{\mathcal{J}}_2)$. By Lemma 4.7, it is enough to show that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$. We consider three cases.

Case 1. $J_{\text{all}} \in O$. Hence, $\cup O = J_{\text{all}}$. Since $J \to J_{\text{all}}$ for all $J \in \tilde{\mathcal{J}}_1$, it holds that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) = 1$.

Case 2. $J_{\text{all}} \notin O$ *and* $|O| < k$. In this case, the probability of O is at most $(k-1)p_0$ and, then, we can use the instance $J_\emptyset$ of $\tilde{\mathcal{J}}_1$ as follows. Since $J_\emptyset \to \cup O$, we have that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O)$ is at least $p_{\tilde{\mathcal{J}}_1}(J_\emptyset)$ which, by definition, is $(k-1)p_0$. Then, from the assumption of this case, we conclude that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O)$ is at least $|O|p_0$, that is, the probability $\Pr_{\tilde{\mathcal{J}}_2}(O)$.

Case 3. $J_{\text{all}} \notin O$ *and* $|O| \geq k$. Here, we use the fact that no independent set of size k exists to conclude that $\cup O$ contains two instances $\{V(i)\}$ and $\{V(i')\}$, such that v_i and $v_{i'}$ are connected by an edge of G. Consequently, there is a homomorphism from the instance $\{V(i), V(i')\}$ to $\cup O$. Therefore, $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq p_{\tilde{\mathcal{J}}_1}(\{V(i), V(i')\})$. Now, recall that $p_{\tilde{\mathcal{J}}_1}(\{V(i), V(i')\})$ is np_0. Since O is a subset of $\{v_1, \ldots, v_n\}$, we get that $|O| \leq n$ and, hence, $p_{\tilde{\mathcal{J}}_1}(\{V(i), V(i')\}) \geq |O|p_0$. Finally, since $J_{\text{all}} \notin O$ we conclude from the construction of $\tilde{\mathcal{J}}_1$ that $|O|p_0$ is exactly $\Pr_{\tilde{\mathcal{J}}_2}(O)$. It follows that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O) \geq \Pr_{\tilde{\mathcal{J}}_2}(O)$, as required.

We now prove the "only if" direction. Suppose that there is an independent set U of size k. Let O_U be the event of $\tilde{\mathcal{J}}_2$ that comprises all the instances $\{V(i)\}$, where $v_i \in U$. Using Part 1 of Lemma 4.7, we will prove that $\tilde{\mathcal{J}}_1 \not\stackrel{\preceq sp}{\Longrightarrow} \tilde{\mathcal{J}}_2$ by showing that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O_U) < \Pr_{\tilde{\mathcal{J}}_2}(O_U)$. Since U is independent, there is no homomorphism from any of the instances $\{V(i), V(i')\} \in \tilde{\mathcal{J}}_1$ to $\cup O_U$. Consequently, the only instance of $\tilde{\mathcal{J}}_1$

that has a homomorphism to $\cup O_U$ is $J_\emptyset$. We conclude that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O_U)$ is equal to $p_{\tilde{\mathcal{J}}_1}(J_\emptyset)$, namely, $(k-1)p_0$. However, $|U| = k$ and, hence, $\Pr_{\tilde{\mathcal{J}}_2}(O_U)$ is, kp_0. We conclude that $\Pr_{\tilde{\mathcal{J}}_1}(\mathcal{J}_1 \to \cup O_U) < \Pr_{\tilde{\mathcal{J}}_2}(O_U)$, as required.

This completes the proof that testing $\overset{\preceq \mathsf{sp}}{\Longrightarrow}$ is coNP-hard. Next, we show how to change the proof for getting DP-hardness.

Let $\mathbf{S}'$ be a schema, such that testing whether there exists a homomorphism between two given instances over $\mathbf{S}'$ is NP-complete (see, e.g., Chandra and Merlin [1977]). Assume that $\mathbf{S}'$ does not contain the relation symbol V of $\mathbf{T}$, and let $\mathbf{S} = \langle \mathbf{S}', \mathbf{T} \rangle$. In order to show that testing $\overset{\preceq \mathsf{sp}}{\Longrightarrow}$ is DP-hard, we will show how to modify the above reduction to obtain one from the following DP-hard problem: Given a pair (J, J') of instances over $\mathbf{S}'$, a graph G and an integer k, determine whether both $J \to J'$ and G has no independent set of size k. This problem is DP-hard, since the conjunction of an NP-hard problem and a coNP-hard problem is a DP-hard problem.

Given the input J, J', G and k, we construct p-instances $\tilde{\mathcal{J}}'_1$ and $\tilde{\mathcal{J}}'_2$ similarly to the way we constructed $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, except that each instance J_1 of $\tilde{\mathcal{J}}_1$ is replaced with $J_1 \cup J$ in $\tilde{\mathcal{J}}'_1$, and each instance J_2 of $\tilde{\mathcal{J}}_2$ is replaced with $J_2 \cup J'$ in $\tilde{\mathcal{J}}'_2$. Previously, we proved the claim that $\tilde{\mathcal{J}}_1 \overset{\preceq \mathsf{sp}}{\Longrightarrow} \tilde{\mathcal{J}}_2$ if and only if G has no independent set of size k; if $J \to J'$, then that claim becomes $\tilde{\mathcal{J}}'_1 \overset{\preceq \mathsf{sp}}{\Longrightarrow} \tilde{\mathcal{J}}'_2$ if and only if G has no independent set of size k. This is for the following reason. In the previous argument, we considered homomorphisms of the form $J_1 \to \cup O$. If we now let J'_1 be $J_1 \cup J$, and let O' be $\{J_2 \cup J' \mid J_2 \in O\}$, then because $J \to J'$, we have that $J_1 \to \cup O$ if and only if $J'_1 \to \cup O'$ (recall that $\mathbf{S}'$ and $\mathbf{T}$ do not have relation symbols in common).

On the other hand, if $J \to J'$ does not hold, then for all $J_1 \in \tilde{\mathcal{J}}'_1$ and $J_2 \in \tilde{\mathcal{J}}'_2$, there is no homomorphism from J_1 to J_2. This implies that $\tilde{\mathcal{J}}'_1 \overset{\preceq \mathsf{sp}}{\Longrightarrow} \tilde{\mathcal{J}}'_2$ is false, because in Part 1 of Lemma 4.7, where the roles of $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ are played by $\tilde{\mathcal{J}}'_1$ and $\tilde{\mathcal{J}}'_2$, respectively, we will never have $\mathcal{J}'_1 \to \cup O'$. From what we have said, it follows fairly easily that $\tilde{\mathcal{J}}'_1 \overset{\preceq \mathsf{sp}}{\Longrightarrow} \tilde{\mathcal{J}}'_2$ if and only if both $J \to J'$ and G has no independent set of size k. Consequently, testing $\overset{\preceq \mathsf{sp}}{\Longrightarrow}$ is DP-hard.

The proof that testing $\overset{\succeq \mathsf{ge}}{\Longrightarrow}$ is coNP-hard is done by changing the proof that $\overset{\preceq \mathsf{sp}}{\Longrightarrow}$ is coNP-hard, as follows. First, we construct $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ exactly as mentioed previously. Second, each instance J (of either $\tilde{\mathcal{J}}_1$ or $\tilde{\mathcal{J}}_2$) is replaced with $\overline{J}$ which is given by

$$\overline{J} \overset{\text{def}}{=} \{V(i) \mid 1 \le i \le n \text{ and } V(i) \notin J\}.$$

In particular, observe that $\overline{J_{\text{all}}} = J_\emptyset$ and $\overline{J_\emptyset} = J_{\text{all}}$. Let $\tilde{\mathcal{J}}'_1$ and $\tilde{\mathcal{J}}'_2$ be obtained from $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$, respectively, by applying this replacement. We will show that $\tilde{\mathcal{J}}'_2 \overset{\succeq \mathsf{ge}}{\Longrightarrow} \tilde{\mathcal{J}}'_1$ if and only if G does not have an independent set of size k.

We first prove the "if" direction. So, we assume that the size of every independent set is smaller than k. Let $O' \subseteq \Omega(\tilde{\mathcal{J}}'_2)$ be given. By Lemma 4.7, it is enough to show that $\Pr_{\tilde{\mathcal{J}}'_1}(\sqcap O' \to \mathcal{J}'_1) \ge \Pr_{\tilde{\mathcal{J}}'_2}(O')$. Similarly to the proof of coNP-hardness for $\overset{\preceq \mathsf{sp}}{\Longrightarrow}$, we consider three cases.

Case 1. $J_\emptyset \in O'$. In this case, $\sqcap O'$ is the empty relation $J_\emptyset$. Since $J_\emptyset \to J$ holds for all $J \in \tilde{\mathcal{J}}'_1$, we have that $\Pr_{\tilde{\mathcal{J}}'_1}(\sqcap O' \to \mathcal{J}'_1) = 1$.

Case 2. $J_\emptyset \notin O'$ *and* $|O'| < k$. In this case, the probability of O' is at most $(k-1)p_0$ and, then, we use the instance J_{all} of $\tilde{\mathcal{J}}'_1$ as follows. Since $\sqcap O' \to J_{\text{all}}$, we have that $\Pr_{\tilde{\mathcal{J}}'_1}(\sqcap O' \to \mathcal{J}'_1)$ is at least $p_{\tilde{\mathcal{J}}'_1}(J_{\text{all}})$ which, by definition, is $(k-1)p_0$. Then, from the assumption of this case we conclude that $\Pr_{\tilde{\mathcal{J}}'_1}(\sqcap O' \to \mathcal{J}'_1)$ is at least $|O'|p_0$, that is, the probability $\Pr_{\tilde{\mathcal{J}}'_2}(O')$.

Case 3. $J_\emptyset \notin O'$ *and* $|O'| \ge k$. Here, we use the fact that no independent set of size k exists. In particular, O' contains two instances $\{\overline{V(i)}\}$ and $\{\overline{V(i')}\}$, such that v_i and $v_{i'}$

are connected by an edge of G. As a result, $\sqcap O'$ does not contain the two facts $V(i)$ and $V(i')$. Consequently, there is a homomorphism from $\sqcap O'$ to $\{\overline{V(i), V(i')}\}$. Therefore,

$$\Pr_{\tilde{\mathcal{J}}'_1}\big(\sqcap O' \to \mathcal{J}'_1\big) \geq p_{\tilde{\mathcal{J}}'_1}(\{\overline{V(i), V(i')}\})\,.$$

Recall that $p_{\tilde{\mathcal{J}}'_1}(\{\overline{V(i), V(i')}\})$ is np_0. Since O' is a subset of $\{v_1, \ldots, v_n\}$, we get that $|O'| \leq n$ and, therefore,

$$p_{\tilde{\mathcal{J}}'_1}(\{V(i), V(i')\}) \geq |O'|p_0\,.$$

Since $J_\emptyset \notin O'$, we conclude from the construction of $\tilde{\mathcal{J}}'_1$ that $|O'|p_0$ is exactly $\Pr_{\tilde{\mathcal{J}}'_2}(O')$. It follows that $\Pr_{\tilde{\mathcal{J}}'_1}(\sqcap O' \to \mathcal{J}'_1) \geq \Pr_{\tilde{\mathcal{J}}'_2}(O')$, as required.

We generalize the proof to DP-hardness like it is done in the case of $\xrightarrow{\preceq \mathsf{sp}}$, except that J is added to all the instances of $\tilde{\mathcal{J}}'_2$, and J' is added to all those of $\tilde{\mathcal{J}}'_1$. Here, in the proof of coNP-hardness, we considered homomorphisms of the form $\sqcap O \to J_1$. If we now let J'_1 be $J_1 \cup J'$, and let O' be $\{J_2 \cup J \mid J_2 \in O\}$, then because $J \to J'$, we have that $\sqcap O \to J_1$ if and only if $\sqcap O' \to J'_1$. $\square$

Theorem 4.9 shows that the three relationships $\xrightarrow{\mathsf{mat}}$, $\xrightarrow{\preceq \mathsf{sp}}$ and $\xrightarrow{\succeq \mathsf{ge}}$ between p-instances are inherently different–not only do they not coincide, but also testing whether $\xrightarrow{\mathsf{mat}}$ holds is of different complexity (under conventional complexity-theoretic assumptions) from the complexity of testing whether $\xrightarrow{\preceq \mathsf{sp}}$ holds or whether $\xrightarrow{\succeq \mathsf{ge}}$ holds. So now, we can give three additional definitions of a universal p-solution as a most general p-solution, where generality is according to each of the three relationships $\xrightarrow{\mathsf{mat}}$, $\xrightarrow{\preceq \mathsf{sp}}$ and $\xrightarrow{\succeq \mathsf{ge}}$. And since the three relationships are inherently different, we might expect to get different definitions of a universal p-solution. Surprisingly, it turns out that all three definitions are equivalent to existence of a $\mathsf{USOL}_{\mathcal{M}}$-match! This is stated in the following theorem. The equivalence of Parts 1 and 5 in this theorem tells us that, for a solution $\tilde{\mathcal{J}}$, either all $\mathsf{SOL}_{\mathcal{M}}$-matches are $\mathsf{USOL}_{\mathcal{M}}$-matches (and then $\tilde{\mathcal{J}}$ is universal) or none of them is a $\mathsf{USOL}_{\mathcal{M}}$-match.

THEOREM 4.10. *Let $\mathcal{M}$ be a schema mapping. Let $\tilde{\mathcal{I}}$ be a source p-instance and let $\tilde{\mathcal{J}}$ be a p-solution. The following are equivalent:*

(1) *$\tilde{\mathcal{J}}$ is a universal p-solution (i.e., there is a $\mathsf{USOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$).*
(2) *$\tilde{\mathcal{J}} \xrightarrow{\mathsf{mat}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(3) *$\tilde{\mathcal{J}} \xrightarrow{\preceq \mathsf{sp}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(4) *$\tilde{\mathcal{J}} \xrightarrow{\succeq \mathsf{ge}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(5) *Every $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$ is a $\mathsf{USOL}_{\mathcal{M}}$-match.*

PROOF. Note that $2 \Rightarrow 3$ and $2 \Rightarrow 4$ follow directly from Part 1 of Theorem 4.9. Also note that $5 \Rightarrow 1$ is straightforward, since $\tilde{\mathcal{J}}$ is a p-solution. So, we have

$$2 \Rightarrow 3\,, \quad 2 \Rightarrow 4\,, \quad 5 \Rightarrow 1\,.$$

$1 \Rightarrow 2$. We assume that $\tilde{\mathcal{J}}$ is a universal p-solution (i.e., there is a $\mathsf{USOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$); we need to show that $\tilde{\mathcal{J}} \xrightarrow{\mathsf{mat}} \tilde{\mathcal{J}}'$ holds for all p-solutions $\tilde{\mathcal{J}}'$. Let $\tilde{\mathcal{J}}'$ be a p-solution and $O \subseteq \Omega(\tilde{\mathcal{J}})$ be an event of $\tilde{\mathcal{J}}$. We will show that we have

$$\Pr_{\tilde{\mathcal{J}}'}\left(\bigvee_{J \in O} J \to \mathcal{J}'\right) \geq \Pr_{\tilde{\mathcal{J}}}(O)\,. \tag{17}$$

Because of (17), Lemma 3.7 implies that $\mathsf{HOM}_{\mathbf{T}}(\tilde{\mathcal{J}}, \tilde{\mathcal{J}}')$ hold.

Consider the event $U \subseteq \tilde{\mathcal{I}}$ that consists of all the instances I, such that some $J \in O$ is a universal solution for I. Then, since there is a $\mathsf{USOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$, Lemma 3.7

implies that

$$\Pr_{\tilde{\mathcal{I}}}(U) \geq \Pr_{\tilde{\mathcal{J}}}(O)\,. \tag{18}$$

Now, since $\tilde{\mathcal{J}}'$ is a p-solution, we conclude from the second characterization of Theorem 3.6 that

$$\Pr_{\tilde{\mathcal{J}}'}\left(\bigvee_{I\in U} \langle I, \mathcal{J}'\rangle \models \Sigma\right) \geq \Pr_{\tilde{\mathcal{I}}}(U)\,. \tag{19}$$

Since every instance $I \in U$ has a universal solution in O, it follows that, for every J' that satisfies $\vee_{I\in U}(\langle I, J'\rangle \models \Sigma)$, there exists an instance $J \in O$, such that $J \to J'$. Thus,

$$\Pr_{\tilde{\mathcal{J}}'}\left(\bigvee_{J\in O} J \to \tilde{\mathcal{J}}'\right) \geq \Pr_{\tilde{\mathcal{J}}'}\left(\bigvee_{I\in U} \langle I, \mathcal{J}'\rangle \models \Sigma\right)\,. \tag{20}$$

Finally, (17) follows from (18), (19), and (20).

$3 \Rightarrow 5$. Suppose that $\tilde{\mathcal{J}} \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$. Also suppose, by way of contradiction, that $\tilde{\mathcal{P}}$ is a $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$ and, moreover, there exist I and J, such that $p_{\tilde{\mathcal{P}}}(I, J) > 0$ and J is not a universal solution for I. By the definition of a universal solution, there exists a solution J' for I, such that there is no homomorphism from J to J'.

We construct a counterexample $\tilde{\mathcal{J}}'$ (where $\tilde{\mathcal{J}}'$ is a solution but where it is false that $\tilde{\mathcal{J}} \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}'$) as follows. The probability $p_{\tilde{\mathcal{J}}'}(J_0)$ of each sample $J_0 \in \mathsf{Inst}(\mathbf{T})$ is given by

$$p_{\tilde{\mathcal{J}}'}(J_0) = \begin{cases} p_{\tilde{\mathcal{J}}}(J_0) & \text{if } J_0 \notin \{J, J'\};\\ p_{\tilde{\mathcal{J}}}(J) - p_{\tilde{\mathcal{P}}}(I, J) & \text{if } J_0 = J;\\ p_{\tilde{\mathcal{J}}}(J') + p_{\tilde{\mathcal{P}}}(I, J) & \text{if } J_0 = J'. \end{cases}$$

Observe that $p_{\tilde{\mathcal{J}}}(J) - p_{\tilde{\mathcal{P}}}(I, J) \geq 0$ since $\tilde{\mathcal{P}}$ is a probabilistic match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$. Now, let $\tilde{\mathcal{Y}}$ be the p-space that is identical to $\tilde{\mathcal{P}}$, except for the probabilities $p_{\tilde{\mathcal{Y}}}(I, J)$ and $p_{\tilde{\mathcal{Y}}}(I, J')$, which are set to 0 and $p_{\tilde{\mathcal{P}}}(I, J') + p_{\tilde{\mathcal{P}}}(I, J)$, respectively. It can easily be verified that $\tilde{\mathcal{Y}}$ is a $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}'$; hence, $\tilde{\mathcal{J}}'$ is a p-solution. Now, observe that $\Pr(\mathcal{J} \to J') < \Pr(\mathcal{J}' \to J')$ since $\tilde{\mathcal{J}}'$ is obtained from $\tilde{\mathcal{J}}$ by moving some positive part of the probability from J to J' (and there is no homomorphism from J to J'). This contradicts the assumption that $\tilde{\mathcal{J}} \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}'$.

$4 \Rightarrow 5$. The proof of $4 \Rightarrow 5$ is similar to that of the implication $3 \Rightarrow 5$, except that we get a contradiction from the observation that $\Pr(J \to \mathcal{J}) > \Pr(J \to \mathcal{J}^c)$. □

Example 4.11. Consider again the schema mapping $\mathcal{M}$ of Example 3.1, and the source and target p-instances that are depicted in Figure 1 (and described in Example 3.1). In Example 4.2, we showed that $\tilde{\mathcal{J}}_1$ is a universal p-solution for $\tilde{\mathcal{I}}$. In Example 4.4, we showed that $\tilde{\mathcal{J}}_1 \overset{\mathsf{mat}}{\longrightarrow} \tilde{\mathcal{J}}_2$ holds. Of course, due to Theorem 4.10, this is no coincidence, since $\tilde{\mathcal{J}}_1$ being universal is equivalent to saying that $\tilde{\mathcal{J}}_1 \overset{\mathsf{mat}}{\longrightarrow} \tilde{\mathcal{J}}'$ (as well as $\tilde{\mathcal{J}}_1 \overset{\preceq_{\mathsf{sp}}}{\longrightarrow} \tilde{\mathcal{J}}'$ and $\tilde{\mathcal{J}}_1 \overset{\succeq_{\mathsf{ge}}}{\longrightarrow} \tilde{\mathcal{J}}'$) holds for all p-solutions $\tilde{\mathcal{J}}'$.

In Section 4.2.1, we give a query-based characterization of a universal p-solution (Proposition 4.14). Taken together with Theorem 4.10, these results show that the notion of a universal p-solution is remarkably robust.

4.2. Query Answering

We now generalize the concept of answering target queries in data exchange. A k-ary *query* over a schema $\mathbf{R}$ is a function Q that maps every instance $J \in \mathsf{Inst}(\mathbf{R})$ to a

set $Q(J) \subseteq \mathsf{dom}(J)^k$, such that Q is invariant under isomorphism of instances (i.e., if $\varphi : J \to J'$ is an isomorphism between the instances J and J', then $\varphi(Q(J)) = Q(J')$). Note that for $k = 0$, the result $Q(J)$ is either $\{()\}$ (denoted **true**) or $\emptyset$ (denoted **false**). Such a query is called *Boolean*. A *conjunctive query* (abbreviated, *CQ*) and a *union of conjunctive queries* (abbreviated, *UCQ*) are special cases of queries. For completeness, we next formally define a CQ and a UCQ.

A CQ has the form $\exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$, where $\mathbf{x}$ and $\mathbf{y}$ are tuples of variables, $\mathbf{c}$ is a tuple of constants (from Const) and $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is a conjunction of atomic formulas over the schema $\mathbf{R}$. We make the safety requirement that all the variables of $\mathbf{x}$ must participate in $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$. A UCQ has the form $\exists \mathbf{y}(\varphi_1(\mathbf{x}, \mathbf{y}, \mathbf{c}) \vee \cdots \vee \varphi_k(\mathbf{x}, \mathbf{y}, \mathbf{c}))$, where $\exists \mathbf{y} \varphi_i(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is a CQ for $1 \leq i \leq k$. Given an instance K over $\mathbf{R}$, the set $Q(K)$ of answers comprises all the possible assignments for $\mathbf{x}$ that result in a clause that is true over K.

We follow the conventional notion [Cohen et al. 2008; Dalvi and Suciu 2004, 2007a] of querying probabilistic databases. Thus, for a k-ary query Q and a p-instance $\tilde{\mathcal{K}}$ (where both Q and $\tilde{\mathcal{K}}$ are over the same schema $\mathbf{R}$), every tuple $\mathbf{a} \in (\mathsf{Const} \cup \mathsf{Var})^k$ has a *confidence* value, which is the probability $\Pr(\mathbf{a} \in Q(\mathcal{K}))$. In practice, the tuples $\mathbf{a}$ often come from some finite[5] set of possible answers, which can be given to the user (along with the confidence values); alternatively, the user may request a few answers having the top probabilities [Re et al. 2007].

Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping and let Q be a k-ary query over $\mathbf{T}$. In the deterministic case, *answering* Q means that, given a (deterministic) source instance I, we produce the certain answers, namely, the tuples $\mathbf{a} \in \mathsf{Const}^k$ that belong to $Q(J)$ for all solutions J for I. We denote this set by $\mathrm{certain}(Q, I, \Sigma)$. Next, we generalize the concept of certain answers to the case of probabilistic source instances. Let $\tilde{\mathcal{I}}$ be a source p-instance. Given $\mathbf{a}$, each p-solution $\tilde{\mathcal{J}}$ gives a (possibly different) probability $\Pr(\mathbf{a} \in Q(\mathcal{J}))$. Consistent with the deterministic case, we would like to characterize $\mathbf{a}$ with a property that is guaranteed in every p-solution. Therefore, we define the *confidence* of $\mathbf{a}$, denoted $conf_Q(\mathbf{a})$, as follows. If there are no solutions, then $conf_Q(\mathbf{a}) = 1$. Otherwise, it is the infimum of the confidences (probabilities) of $\mathbf{a}$ over all the p-solutions, namely,

$$conf_Q(\mathbf{a}) \stackrel{\text{def}}{=} \inf_{\text{p-solutions } \tilde{\mathcal{J}}} \Pr\left(\mathbf{a} \in Q(\mathcal{J})\right) .$$

If Q is Boolean, we write $conf_Q$ instead of $conf_Q(())$.

The following proposition shows that the confidence of an answer $\mathbf{a}$ is the same as the probability that $\mathbf{a}$ is certain in a random source instance (given that a p-solution exists). This equality is interesting, because the two numbers describe apparently different quantities: one is the infimum, over all p-solutions, of the probability of an event defined over the p-solutions (specifically, the probability of having $\mathbf{a}$ as an answer), whereas the other is the probability of an event defined over the source p-instance (specifically, the probability of having $\mathbf{a}$ in the certain answers). In particular, this proposition shows the robustness of our generalization of the notion of target-query answering.

PROPOSITION 4.12. *Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping, let Q be a query over $\mathbf{T}$, and let $\tilde{\mathcal{I}}$ be a source p-instance, such that a p-solution exists. For all tuples $\mathbf{a}$ of constants,*

$$conf_Q(\mathbf{a}) = \Pr_{\tilde{\mathcal{I}}}\left(\mathbf{a} \in \mathrm{certain}(Q, \mathcal{I}, \Sigma)\right) .$$

PROOF. Let $\mathbf{a}$ be a tuple of constants. Denote by $\mathbf{I^a}$ the set of all instances $I \in \mathsf{Inst}(\mathbf{S})$ such that $\mathbf{a} \in \mathrm{certain}(Q, I, \Sigma)$. We need to show that $conf_Q(\mathbf{a}) = \Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$

[5]This is the case when $\tilde{\mathcal{K}}$ is a finite p-space.

We first prove that $conf_Q(\mathbf{a}) \leq \Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$. Let $\tilde{\mathcal{J}}$ be a p-solution. By Characterization 2 of Theorem 3.6,

$$\Pr_{\tilde{\mathcal{I}}}\left(\mathbf{I^a}\right) \leq \Pr_{\tilde{\mathcal{J}}}\left(\bigvee_{I \in \mathbf{I^a}} \langle I, \mathcal{J}\rangle \models \Sigma\right). \tag{21}$$

Every sample J of $\tilde{\mathcal{J}}$ that is a solution for some instance I of $\mathbf{I^a}$ satisfies $\mathbf{a} \in Q(J)$. Therefore,

$$\Pr_{\tilde{\mathcal{J}}}\left(\bigvee_{I \in \mathbf{I^a}} \langle I, \mathcal{J}\rangle \models \Sigma\right) \leq \Pr_{\tilde{\mathcal{J}}}\left(\mathbf{a} \in Q(\mathcal{J})\right). \tag{22}$$

Since $\tilde{\mathcal{J}}$ is an arbitrary p-solution, it follows from (22) and the definition of $conf_Q(\mathbf{a})$ (as the infimum of $\Pr_{\tilde{\mathcal{J}}}\left(\mathbf{a} \in Q(\mathcal{J})\right)$ over all p-solutions $\tilde{\mathcal{J}}$) that

$$\Pr_{\tilde{\mathcal{J}}}\left(\bigvee_{I \in \mathbf{I^a}} \langle I, \mathcal{J}\rangle \models \Sigma\right) \leq conf_Q(\mathbf{a}). \tag{23}$$

So, $\Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a}) \leq conf_Q(\mathbf{a})$ is proved by combining (21) and (23).

To show that $conf_Q(\mathbf{a}) \leq \Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$, we will construct a p-solution $\tilde{\mathcal{J}}$, such that $\Pr_{\tilde{\mathcal{J}}}(\mathbf{a} \in Q(\mathcal{J})) = \Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$. This is sufficient due to the definition of $conf_Q(\mathbf{a})$. For all $I \in \Omega_+(\tilde{\mathcal{I}})$, the target instance J_I is defined as follows. If $I \in \mathbf{I^a}$, then J_I is an arbitrary solution for I (in particular, $\mathbf{a} \in Q(J_I)$). Otherwise (i.e., if $I \notin \mathbf{I^a}$), J_I is a solution J for I such that $\mathbf{a} \notin Q(J)$. Observe that the definitions of a p-solution and that of a certain answer imply that every J_I exists.

To construct the p-solution $\tilde{\mathcal{J}}$, we define a p-space $\tilde{\mathcal{P}}$ over $\mathsf{SOL}_{\mathcal{M}}$, such that the marginal distribution on its left side coincides with $\tilde{\mathcal{I}}$. As a result, its marginal distribution on the right side is, by definition, a p-solution, and $\tilde{\mathcal{J}}$ will be chosen to be that p-solution. The p-space $\tilde{\mathcal{P}}$ is defined as follows. For all $I \in \Omega_+(\tilde{\mathcal{I}})$ we define $p_{\tilde{\mathcal{P}}}(I, J_I) = p_{\tilde{\mathcal{I}}}(I)$; each of the other pairs (I, J) has the probability 0. Clearly, $\tilde{\mathcal{I}}$ coincides with the left marginal distribution of $\tilde{\mathcal{P}}$; hence, we get the p-solution $\tilde{\mathcal{J}}$. Note that $\tilde{\mathcal{J}}$ is left-trivial.

By the definition of a marginal distribution, sampling $\tilde{\mathcal{J}}$ is the same as sampling $\tilde{\mathcal{P}}$ and taking the right-hand element. Therefore, the probability $\Pr_{\tilde{\mathcal{J}}}(\mathbf{a} \in Q(\mathcal{J}))$ is the probability that a random sample (I, J) of $\tilde{\mathcal{P}}$ is one such that $\mathbf{a} \in Q(J)$ which, by the construction of $\tilde{\mathcal{P}}$, is the same as saying $I \in \mathbf{I^a}$. But, since the marginal distribution of $\tilde{\mathcal{P}}$ coincides with $\tilde{\mathcal{I}}$, this probability is exactly $\Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$. We conclude that $\Pr_{\tilde{\mathcal{J}}}(\mathbf{a} \in Q(\mathcal{J})) = \Pr_{\tilde{\mathcal{I}}}(\mathbf{I^a})$, as claimed. □

As a part of the proof of Proposition 4.12, we constructed a p-solution $\tilde{\mathcal{J}}$, such that $\Pr(\mathbf{a} \in Q(\mathcal{J}))$ is equal to the probability on the right-hand side of the equality. Thus, the infimum in the definition of confidence is always realized by some p-solution (hence, it can be replaced with *minimum*, even in the infinite case).

Example 4.13. Consider again the schema mapping $\mathcal{M}$, and the p-instances $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ of Example 3.1. Recall from Example 3.5 that both $\tilde{\mathcal{J}}_1$ and $\tilde{\mathcal{J}}_2$ are p-solutions. Let Q be the following target CQ, which extracts all the universities where both IR and AI research is conducted.

$$Q(u) \text{:-} \exists d_1, d_2 \left(UArea(u, d_1, \mathsf{IR}) \wedge UArea(u, d_2, \mathsf{AI})\right)$$

For $\tilde{\mathcal{J}}_1$, there is only one possible answer, which is $\mathbf{a} = (\mathsf{UCSD})$. Since $\Pr(\mathbf{a} \in Q(\mathcal{J}_1)) = 0.3$, we get that $conf_Q(\mathbf{a}) \leq 0.3$. Hence, the value of the left-hand side of the equality in

Proposition 4.12 is at most 0.3. What about the right-hand side, which is the probability that $\mathbf{a}$ is a certain answer? Since $\mathbf{a}$ is a certain answer only for I_2, and I_2 has the probability 0.3, the right-hand side of the equality in Proposition 4.12 is 0.3. Hence, by Proposition 4.12, $conf_Q(\mathbf{a})$ is 0.3, and so is realized by $\tilde{\mathcal{J}}_1$; that is, $\tilde{\mathcal{J}}_1$ is a p-solution $\tilde{\mathcal{J}}$ such that $\Pr(\mathbf{a} \in Q(\mathcal{J}))$ is minimal. In contrast, for $\tilde{\mathcal{J}}_2$ we have $\Pr(\mathbf{a} \in Q(\mathcal{J}_2)) = 0.65$, which is strictly larger than $conf_Q(\mathbf{a})$.

Earlier, in Example 4.2, we noted that the $\mathsf{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}_2$ on the right side of Figure 2 is not a $\mathsf{USOL}_{\mathcal{M}}$-match. Thus, by the equivalence of Parts 1 and 5 of Theorem 4.10, $\tilde{\mathcal{J}}_2$ is *not* a universal p-solution for $\tilde{\mathcal{I}}$. Moreover, recall from Example 4.13 that $\Pr(\mathbf{a} \in Q(\mathcal{J}_2))$ is strictly larger than $conf_Q(\mathbf{a})$. The following section shows how this latter fact gives another proof that $\tilde{\mathcal{J}}_2$ is not universal.

4.2.1. UCQs over Universal p-Solutions. In the deterministic case, a universal solution can be used for answering target UCQs in the sense that the result of applying the query to the universal solution (and then restricting to the tuples of constants) is the set of all certain answers [Fagin et al. 2005a]. Moreover, a solution that has this property for every CQ is necessarily universal [Fagin et al. 2005a]. The following proposition shows that, although the concepts of deterministic and probabilistic query answering are inherently different, this property of universal solutions generalizes to universal p-solutions. That is, the confidence of an answer for a UCQ is obtained by querying a universal p-solution (when one exists); also, a p-solution that has this property for every (Boolean) CQ is necessarily universal.

PROPOSITION 4.14. *Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping and let $\tilde{\mathcal{I}}$ be a source p-instance. The following holds.*

(1) *If $\tilde{\mathcal{J}}$ is a universal p-solution and Q is a UCQ over $\mathbf{T}$, then $conf_Q(\mathbf{a}) = \Pr(\mathbf{a} \in Q(\mathcal{J}))$ for all tuples $\mathbf{a}$ of constants.*
(2) *If $\tilde{\mathcal{J}}$ is a p-solution such that $conf_Q = \Pr(Q(\mathcal{J}))$ holds for all Boolean CQs Q, then $\tilde{\mathcal{J}}$ is a universal p-solution.*

PROOF. We begin with Part 1. Since $\tilde{\mathcal{J}}$ is a p-solution, $\Pr(\mathbf{a} \in Q(\mathcal{J})) \geq conf_Q(\mathbf{a})$ holds by definition. So we need to show that $\Pr(\mathbf{a} \in Q(\mathcal{J})) \leq conf_Q(\mathbf{a})$. Let $O \subseteq \Omega(\tilde{\mathcal{J}})$ be the set of all instances J that satisfy $\mathbf{a} \in Q(J)$. Then, $\Pr(\mathbf{a} \in Q(\mathcal{J})) = \Pr_{\tilde{\mathcal{J}}}(O)$. Now, consider a p-solution $\tilde{\mathcal{J}}'$. From the fact that $\tilde{\mathcal{J}} \xrightarrow{\text{mat}} \tilde{\mathcal{J}}'$ and Lemma 3.7, we get that

$$\Pr_{\tilde{\mathcal{J}}}(O) \leq \Pr_{\tilde{\mathcal{J}}'}\left(\bigvee_{J \in O} J \to \mathcal{J}'\right).$$

In Fagin et al. [2005a], it is shown that if $\mathbf{a} \in Q(J)$ and $J \to J'$, then $\mathbf{a} \in Q(J')$ as well. Therefore, a homomorphism from one (or more) of the instances of O in $\mathcal{J}'$ implies membership of $\mathbf{a}$ in $Q(\mathcal{J}')$. Consequently,

$$\Pr\left(\bigvee_{J \in O} J \to \mathcal{J}'\right) \leq \Pr\left(\mathbf{a} \in Q(\mathcal{J}')\right).$$

We conclude that $\Pr(\mathbf{a} \in Q(\mathcal{J})) \leq \Pr(\mathbf{a} \in Q(\mathcal{J}'))$. Since it holds for all p-solutions $\tilde{\mathcal{J}}'$, we get that

$$\Pr(\mathbf{a} \in Q(\mathcal{J})) \leq conf_Q(\mathbf{a}).$$

For Part 2, consider a p-solution $\tilde{\mathcal{J}}$ such that $\Pr(Q(\mathcal{J})) = conf_Q$ holds for all Boolean conjunctive queries Q. By Theorem 4.10, it is enough to show that $\tilde{\mathcal{J}} \xrightarrow{\geq \text{ge}} \tilde{\mathcal{J}}'$ for all

p-solutions $\tilde{\mathcal{J}}'$. Let $\tilde{\mathcal{J}}'$ be a p-solution, and let J be an instance over $\mathbf{T}$. We need to show that $\Pr(J \to \tilde{\mathcal{J}}) \leq \Pr(J \to \tilde{\mathcal{J}}')$. Let Q^J be the *canonical* Boolean conjunctive query associated with J (as defined in, e.g., Chandra and Merlin [1977]). As shown in Chandra and Merlin [1977], for all instances J over $\mathbf{T}$, it holds that $Q^J(J') = \mathbf{true}$ if and only if $J \to J'$. Consequently,

$$\Pr(J \to \tilde{\mathcal{J}}) = \Pr(Q^J(\mathcal{J})) \text{ and } \Pr(J \to \tilde{\mathcal{J}}') = \Pr(Q^J(\mathcal{J}')).$$

Now, $\Pr(Q^J(\mathcal{J})) = conf_Q$ implies that

$$\Pr(Q^J(\mathcal{J})) \leq \Pr(Q^J(\mathcal{J}')).$$

We conclude that $\Pr(J \to \tilde{\mathcal{J}}) \leq \Pr(J \to \tilde{\mathcal{J}}')$, as claimed. □

In the next section, we study computational aspects of probabilistic data exchange. In particular, we consider the tasks of testing whether a (universal) p-solution exists, materializing one (when it exists), and evaluating target UCQs. By Proposition 4.3, a p-solution exists if and only if there is a solution for I with respect to Σ for all $I \in \Omega_+(\tilde{\mathcal{I}})$. By the discussion that follows. Proposition 4.3, if a p-solution exists, then we can materialize one (a left-trivial one) using solutions for the instances of $\Omega_+(\tilde{\mathcal{I}})$, by a straightforward construction. A similar comment applies to universal (p-) solutions. Proposition 4.12 implies that we can compute $conf_Q(\mathbf{a})$ by determining whether $\mathbf{a} \in \text{certain}(Q, I, \Sigma)$ for each $I \in \Omega_+(\tilde{\mathcal{I}})$, and taking the sum of the probabilities of the instances I for which the answer is "yes."

Consequently, in the case of finite p-instances, these tasks in the probabilistic setting are not harder than their traditional counterparts. Nevertheless, this analysis is based on the assumption that source p-instances are represented in an explicit manner (i.e., by specifying each possible instance along with its probability). This is not a practical assumption, as evidenced by existing models of probabilistic databases (e.g., Agrawal et al. [2006], Boulos et al. [2005], Dalvi and Suciu [2004, 2007a], and Sarma et al. [2008b]) that usually employ a (typically logarithmic-scale) compact encoding of the possible worlds. So, the next section studies these computational problems under some typical compact representations of probabilistic databases.

5. COMPACT REPRESENTATION

In this section, we explore complexity aspects of data exchange in a concrete setting where dependencies are in the form of *tgd*s and *egd*s [Beeri and Vardi 1984; Fagin et al. 2005a] (the formal definitions are in Section 5.2), and p-instances are represented compactly by annotating facts with probabilistic conditions [Fuhr and Rölleke 1997; Green and Tannen 2006; Green et al. 2007b] rather than explicitly specifying the whole probability space.

5.1. Annotated Instances

We consider p-instances that are represented by means of *Boolean pc-tables* [Green and Tannen 2006] (which are the probabilistic version of *c-tables* [Imielinski and Lipski 1984]) where the condition assigned to each fact is a logical formula over *event variables*—probabilistically independent Boolean (Bernoulli) random variables. In pc-tables, conditions can be phrased as arbitrary propositional-logic formulas, which renders the most basic operations as intractable since, for one, it is NP-complete even to decide whether a given fact occurs with a nonzero probability. Thus, our focus is on two restricted representations that correspond to (or subsume) various representations in the literature. In the first, conditions are in disjunctive normal form (DNF), and in the second, the facts are probabilistically independent. Next, we give the formal definitions.

I_p^α

	Fact f	Condition $\alpha(f)$
r_{e}	*Researcher*(Emma, UCSD)	**true**
r_{j}	*Researcher*(John, UCSD)	$e_1 \vee e_2 \vee e_3 \vee e_4$
a_{eir}	*RArea*(Emma, IR)	$e_1 \vee e_2$
a_{edb}	*RArea*(Emma, DB)	$\neg e_1 \wedge \neg e_2$
a_{jdb}	*RArea*(John, DB)	$e_1 \vee (\neg e_2 \wedge \neg e_3 \wedge e_4)$
a_{jai}	*RArea*(John, AI)	$(\neg e_1 \wedge e_2) \vee (\neg e_1 \wedge e_3)$

$$\mathsf{EVar}(\alpha) = \{e_1, e_2, e_3, e_4\}$$

$$p : \mathsf{EVar}(\alpha) \to [0,1]$$
$$p(e_1) = 3/10, \quad p(e_2) = 3/7, \quad p(e_3) = p(e_4) = 1/2$$

Fig. 6. A DNF instance I_p^α.

We assume an infinite set EVar of event variables. A *DNF formula* (over EVar) is a formula of the form $\varphi_1 \vee \cdots \vee \varphi_m$, where each φ_i is a conjunction of variables in EVar and negations of variables in EVar. Let $\mathbf{R}$ be a schema. A *DNF instance* (*over* $\mathbf{R}$) comprises an instance I over $\mathbf{R}$, a function α that maps every fact f of I to a DNF formula $\alpha(f)$ over EVar, and a function $p : \mathsf{EVar}(\alpha) \to [0,1]$ where $\mathsf{EVar}(\alpha)$ is the set of all the event variables that appear in the image of α. The DNF instance given by I, α and p is denoted by I_p^α. A DNF instance I_p^α naturally encodes a p-instance, which we denote by p-space(I_p^α), where a sample I' is obtained as follows: First, a random truth assignment $\tau : \mathsf{EVar}(\alpha) \to \{\mathbf{true}, \mathbf{false}\}$ is chosen for the event variables of I; this assignment is obtained by independently picking a random Boolean value $\tau(e)$, with probability $p(e)$ for **true**, for each member e of $\mathsf{EVar}(\alpha)$. Second, all the facts f such that τ satisfies the formula $\alpha(f)$ are selected as members of I' (alternatively, I' is obtained from I by removing all the facts f such that $\alpha(f)$ is violated). Thus, p-space(I_p^α) is the finite p-instance $\tilde{\mathcal{I}}$ such that $\Omega_+(\tilde{\mathcal{I}})$ comprises instances with facts from I, and for all $I' \subseteq I$ the probability $p_{\tilde{\mathcal{I}}}(I')$ is that of obtaining I' in this process (namely, the sum of the probabilities of all the assignments that satisfy every formula of I' and none of $I \setminus I'$).

Example 5.1. Figure 6 depicts a DNF instance I_p^α. The table on the top of the figure has a row for each fact, where the right column contains the condition of the corresponding fact. For example, the third row shows a_{eir}, which is the fact *RArea*(Emma, IR), and the condition $\alpha(f)$ is $e_1 \vee e_2$. As shown in the middle part of the figure, $\mathsf{EVar}(\alpha)$ contains the four event variables $e_1, \ldots, e_4$. Finally, the function p is specified at the bottom.

Note that the facts of I are those that are depicted in Figure 1, that is, the possible facts of the p-instance $\tilde{\mathcal{I}}$. The reader can verify that I_p^α encodes[6] exactly the p-instance $\tilde{\mathcal{I}}$ of Figure 1; that is, $\tilde{\mathcal{I}} =$ p-space(I_p^α) (which means that $\tilde{\mathcal{I}}$ and p-space(I_p^α) have the same support, and the same probability for each instance in their support). As an example, let us compute the probability of the instance $I_5 = \{r_{\mathsf{e}}, a_{\mathsf{edb}}\}$ (from Figure 1). In general, an instance can be produced by multiple truth assignments, but I_5 is produced by

[6]The translation of $\tilde{\mathcal{I}}$ into I_p^α follows standard techniques of encoding finite p-spaces by annotations (see, e.g., Green and Tannen [2006] and Senellart and Abiteboul [2007]).

I_p^α

	Fact f	$\alpha(f)$	$p(\alpha(f))$
r_{e}	*Researcher*(Emma, UCSD)	e'_0	1.0
r_{j}	*Researcher*(John, UCSD)	e'_1	0.9
a_{eir}	*RArea*(Emma, IR)	e'_2	0.6
a_{edb}	*RArea*(Emma, DB)	e'_3	0.4
a_{jdb}	*RArea*(John, DB)	e'_4	0.4
a_{jai}	*RArea*(John, AI)	e'_5	0.5

Fig. 7. A tuple-independent instance I_p^α.

only the assignment that maps all four variables to **false**, because $r_{\mathsf{j}} \notin I_5$. Let τ be that assignment. Observe that τ indeed produces I_5 since τ violates the condition of every fact other than r_{e} and a_{edb}. Therefore, the probability of I_5 is the probability of τ, namely, $\frac{7}{10} \times \frac{4}{7} \times \frac{1}{2} \times \frac{1}{2} = \frac{28}{280} = 0.1$. As another example, the reader can verify that the assignments τ that map e_1 to **true** are exactly those that result in the instance $I_1 = \{r_{\mathsf{e}}, r_{\mathsf{j}}, a_{\mathsf{eir}}, a_{\mathsf{jdb}}\}$; therefore, the probability of I_1 is $p(e_1) = 0.3$.

In Green and Tannen [2006], it is shown that every finite p-instance can be represented by means of Boolean pc-tables (i.e., Boolean pc-tables are "complete"). In particular, every finite p-instance $\tilde{\mathcal{I}}$ is equal to p-space(I_p^α) for some DNF instance I_p^α, since every formula in propositional logic can be transformed into DNF. Note that this translation may entail an exponential blowup. But, one can efficiently translate into DNF instances other representations like *block-independent disjoint* databases [Re and Suciu 2007a, 2007b; Re et al. 2007] and *probabilistic rdb's* [Kimelfeld and Sagiv 2007].

A special case of a DNF instance is one where tuples are probabilistically independent. Formally, a *tuple-independent* instance is a DNF instance I_p^α such that for all facts $f \in I$, the condition $\alpha(f)$ is a distinct atomic event variable e_f (i.e., $e_f \neq e_g$ for $f \neq g$); in particular, the facts of I_p^α are probabilistically independent. We require a tuple-independent instance I_p^α to be such that the function p is strictly positive (i.e., $p(e) > 0$ for all $e \in \mathsf{EVar}(\alpha)$). This is not a restriction, since a fact with zero-probability event can simply be removed.

Example 5.2. Figure 7 depicts a tuple-independent instance I_p^α. Each row shows a fact f, the unique variable $e'_i = \alpha(f)$ and the probability $p(e'_i)$. The facts of Figure 7 are the same as those of Figure 1 (and those of Figure 6, which is discussed in Example 5.1). Let $\tilde{\mathcal{I}}$ be as in Figure 1. The probabilities $p(e'_i)$ are chosen to be such that the probability of each fact f in I_p^α is the marginal probability of f in $\tilde{\mathcal{I}}$ of Figure 1 (i.e., $p(\alpha(f))$ is the sum of the probabilities of the instances $I \in \Omega_+(\tilde{\mathcal{I}})$ with $f \in I$). However, unlike Figure 6, the instance I_p^α of Figure 7 does *not* encode $\tilde{\mathcal{I}}$ (that is, p-space(I_p^α) $\neq \tilde{\mathcal{I}}$). Moreover, no tuple-independent instance encodes $\tilde{\mathcal{I}}$, simply because the facts of $\tilde{\mathcal{I}}$ are not independent. As an example, the facts a_{eir} and a_{edb} are mutually exclusive in $\tilde{\mathcal{I}}$ (hence, they are not independent).

In terms of representations of probabilistic data in the literature, tuple-independent instances are sets of *p-?-tables* [Green and Tannen 2006], and they are the same as the *tuple-independent probabilistic structures* of Dalvi and Suciu [2007a] (called *probabilistic databases* in Dalvi and Suciu [2007b]). We could avoid using event variables in tuple-independent instances, and just write a number next to each fact (as done

in Dalvi and Suciu [2007a, 2007b]). However, it is convenient for us to syntactically view these instances as special cases of DNF instances.

Consider a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$. A *source DNF instance* is a DNF instance I_p^α over $\mathbf{S}$, such that I is a ground instance, and a *target DNF instance* is a DNF instance J_q^β over $\mathbf{T}$ (J is not necessarily ground). Special cases are source and target tuple-independent instances. Clearly, if I_p^α and J_q^β are source and target DNF instances, then p-space(I_p^α) and p-space(J_q^β) are source and target p-instances, respectively.

5.2. Tuple/Equality-Generating Dependencies

We consider two specific types of dependencies that were studied in past research on data exchange (e.g., Fagin et al. [2005a, 2005b]); each dependency is a *tuple-generating dependency* (*tgd*) or an *equality-generating dependency* (*egd*) [Beeri and Vardi 1984]. More particularly, let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. A *source-to-target tgd* (*st-tgd*) is a formula of the form

$$\forall \mathbf{x}(\varphi_{\mathbf{S}}(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})),$$

a *target tgd* (*t-tgd*) is one of the form

$$\forall \mathbf{x}(\varphi_{\mathbf{T}}(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})),$$

and a *target egd* (*t-egd*) has the form

$$\forall \mathbf{x}(\varphi_{\mathbf{T}}(\mathbf{x}) \rightarrow (x_1 = x_2)).$$

In the preceding formulas, $\varphi_{\mathbf{S}}(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{S}$, and each of $\varphi_{\mathbf{T}}(\mathbf{x})$ and $\psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas over $\mathbf{T}$. Moreover, all the variables of $\mathbf{x}$ appear in both $\varphi_{\mathbf{S}}(\mathbf{x})$ and $\varphi_{\mathbf{T}}(\mathbf{x})$, and $\mathbf{x}$ contains the variables x_1 and x_2. As a special case, *full* st-tgds and *full* t-tgds are ones that do not contain existentially quantified variables (i.e., $\mathbf{y}$ is empty).

5.3. Complexity Results

We use data complexity for analyzing the computational problems that we address. In particular, we assume that the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is fixed, and the input consists of the source DNF instance I_p^α. If a query Q is involved, then it is fixed as well. For all variables $e \in \mathsf{EVar}(\alpha)$, the number $p(e)$ is a rational number represented by a pair of integers (the numerator and the denominator). Finally, we consider only schema mappings where the set Σ of dependencies is the union of finite sets Σ_1 and Σ_2, such that Σ_1 contains only st-tgds and t-egds, and Σ_2 is a *weakly acyclic* set of t-tgds (see Fagin et al. [2005a] for the formal definition of weak acyclicity). Such schema mappings are called *standard schema mappings* in Arenas et al. [2010].

The complexity results are shown in Table I. We study five computational problems, and give their complexity for each of the two types of source p-instances: the top five rows consider source DNF instances, and the bottom five rows are for source tuple-independent instances. Each row is associated with a specific problem. Each column corresponds to a class of schema mappings. For example, the column entitled "full st-tgds, t-egds" considers schema mappings $(\mathbf{S}, \mathbf{T}, \Sigma)$ such that Σ contains only full st-tgds and t-egds. Though not all the possible combinations of (full) st-tgds, (full) t-tgds and t-egds are mentioned in Table I, this table actually covers all possible combinations, in the following sense. Each missing combination lies between two combinations that have the same complexity results in the table. For example, the combination "st-tgds, full t-tgds" (which is not in the table) is between "full st-tgds, full t-tgds" and "st-tgds, w.a. t-tgds," and the complexity results for these two combinations are exactly the same; hence, these results also hold for the missing "st-tgds, full t-tgds."

Table I. Complexity of Testing for the Existence of a (Universal) p-Solution, Materializing a Candidate (Universal) p-Solution as a DNF Instance, and (Exact and Approximate) Evaluation of Target UCQs

I_p^α	Problem	*st-tgds, t-egds, w.a. t-tgds*	*st-tgds, t-egds*	*st-tgds, w.a. t-tgds*	*full st-tgds, t-egds, full t-tgds*	*full st-tgds, full t-tgds*	*full st-tgds, t-egds*	*st-tgds*	*full st-tgds*
DNF	*Existence of (U.) p-Solutions*	coNP-complete [*Prop 5.3, Prop 5.5, Thm 5.6*]	coNP-complete [*Prop 5.3, Prop 5.5, Thm 5.6*]	trivial [*Prop 4.3*]	coNP-complete [*Prop 5.3, Prop 5.5, Thm A.4*]	trivial [*Prop 4.3*]	PTIME [*Prop 5.3, Thm A.3*]	trivial [*Prop 4.3*]	trivial [*Prop 4.3*]
	Materialize a p-Solution	$\notin$ FP if P$\neq$NP [*Thm A.10*]	$\notin$ FP if P$\neq$NP [*Thm A.10*]	FP [*Prop A.9*]	$\notin$ FP if P$\neq$NP [*Thm A.10*]	FP [*Prop A.9*]	FP [*Thm A.8*]	FP [*Thm A.6*]	FP [*Thm A.6*]
	Materialize a U. p-Solution	$\notin$ FP if P$\neq$NP [*Prop 5.3, Thm A.10*]	$\notin$ FP if P$\neq$NP [*Prop 5.3, Thm A.10*]	$\notin$ FP if P$\neq$NP [*Thm A.11*]	$\notin$ FP if P$\neq$NP [*Prop 5.3, Thm A.10*]	$\notin$ FP if P$\neq$NP [*Thm A.11*]	FP [*Thm A.8*]	FP [*Thm A.6*]	FP [*Thm A.6*]
	Target UCQ: Exact	FP$^{\#P}$-complete [*Prop 5.8, Prop A.13*]							
	Target UCQ: Approx.	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm A.16*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	FPRAS [*Thm A.15*]	FPRAS [*Thm A.15*]	FPRAS [*Thm A.15*]
Tuple-Independent	*Existence of (U.) p-Solutions*	PTIME [*Prop 5.3, Prop 5.4*]	PTIME [*Prop 5.3, Prop 5.4*]	trivial [*Prop 4.3*]	PTIME [*Prop 5.3, Prop 5.4*]	trivial [*Prop 4.3*]	PTIME [*Prop 5.3, Prop 5.4*]	trivial [*Prop 4.3*]	trivial [*Prop 4.3*]
	Materialize a p-Solution	FP [*Prop A.9*]							
	Materialize a U. p-Solution	$\notin$ FP if RP$\neq$NP [*Thm A.12*]	$\notin$ FP if RP$\neq$NP [*Thm A.12*]	$\notin$ FP if RP$\neq$NP [*Thm 5.7*]	$\notin$ FP if RP$\neq$NP [*Thm 5.7*]	$\notin$ FP if RP$\neq$NP [*Thm 5.7*]	FP [*Thm A.8*]	FP [*Thm A.6*]	FP [*Thm A.6*]
	Target UCQ: Exact	FP$^{\#P}$-complete [*Prop 5.8, Prop A.13*]							
	Target UCQ: Approx.	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm A.18*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	inapprox. if RP$\neq$NP [*Thm 5.9*]	FPAS [*Thm A.15*]	FPAS [*Thm A.15*]	FPAS [*Thm A.15*]

The problem entitled "*Existence of (U.) p-Solutions*" is that of deciding whether a p-solution exists. Later, we will show that this problem is the same as deciding whether a universal p-solution exists (hence the "U." that stands for "Universal"). By "trivial" we mean that a p-solution always exists. Here and later, an upper bound (e.g., "PTIME") refers to all schema mappings in the corresponding column, whereas a lower bound (e.g., "coNP-complete") means that there exists a schema mapping, in the corresponding column, for which the result holds. Furthermore, "coNP-complete" means that the problem is in coNP for all schema mappings in the corresponding column, and there is a schema mapping, in the corresponding column, where the problem is coNP-hard.

The problem entitled "*Materialize a p-Solution*" is that of materializing a *candidate p-solution*, namely, a target p-instance $\tilde{J}$ that forms a p-solution if one exists. We

restrict to generation of candidate p-solutions $\tilde{J}$ that are represented as DNF instances J_q^β (i.e., $\tilde{J} = \text{p-space}(J_q^\beta)$). The problem entitled "*Materialize a U. p-Solution*." is the universal version of the second, namely, generation of a *candidate universal p-solution* as a DNF instance. Note that FP is the class of polynomial-time computable *functions* (while PTIME is the class of polynomial-time solvable decision problems). The table contains three types of results for the two materialization problems: FP, not in FP unless P = NP, and not in FP unless RP = NP.[7]

The problem entitled "*Target UCQ: Exact*" is that of evaluating unions of conjunctive queries. Recall that $\text{FP}^{\#\text{P}}$ is the class of functions that are efficiently computable using an oracle to some function in #P.[8] Note that a function F is $\text{FP}^{\#\text{P}}$-hard[9] if there is a polynomial-time *Turing reduction* (or *Cook reduction*) from every function in $\text{FP}^{\#\text{P}}$ to F.

Finally, the problem entitled "*Target UCQ: Approx.*" is that of evaluating unions of conjunctive queries in an approximate manner. We postpone the precise definition of "approximate manner," as well as those of the corresponding complexity results in the table, to Section 5.3.4. For now, we just mention that "FPRAS" and "FPAS" are upper bounds indicating that the problem is efficiently approximable (and actually in a strong sense), and that "inapprox. if RP $\neq$ NP" is a lower bound indicating inapproximability.

Table I shows that the studied problems are often hard. On the positive side, observe that for the rightmost three columns, all the problems (except for exact query answering) are tractable. Next, we discuss the complexity results in detail. To simplify the presentation, we prove some of the results of the table in the body of the article; the remaining proofs are in the Appendix.

5.3.1. Notation. Before we proceed to discuss the complexity results, let us first introduce some useful notation. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. We say that $\mathcal{M}$ is a *standard schema mapping* [Arenas et al. 2010] if Σ is the union of finite sets Σ_1 and Σ_2, such that Σ_1 contains only st-tgds and t-egds, and Σ_2 is a *weakly acyclic* set of t-tgds. In this section, we consider only standard schema mappings. Note that if Σ contains only st-tgds, t-tgds, and t-egds, and all the t-tgds of Σ are full, then $\mathcal{M}$ is necessarily standard.

Let $\mathcal{M}$ be a schema mapping and let I_p^α be source p-instance. Let τ be a truth assignment for $\mathsf{EVar}(\alpha)$. We denote by I_τ^α the source instance $I' \subseteq I$ that consists of all the facts f, such that $\alpha(f)$ is satisfied by τ. We say that τ is *feasible* if for all event symbols $e \in \mathsf{EVar}(\alpha)$, if $p(e) = 1$, then $\tau(e)$ is **true**, and if $p(e) = 0$, then $\tau(e)$ is **false** (in any other case, $\tau(e)$ can be either **true** or **false**); in other words, τ is feasible if it has a nonzero probability. Note that an instance I' is in the support of $\text{p-space}(I_p^\alpha)$ if and only if there exists a feasible truth assignment τ such that $I' = I_\tau^\alpha$. An instance $I' \subseteq I$ is said to be *feasible* if there exists a feasible truth assignment τ such that $I' \subseteq I_\tau^\alpha$; that is, the set $\{\alpha(f) \mid f \in I'\}$ is satisfiable by some feasible truth assignment.

5.3.2. Existence of p-Solutions. The first problem is that of deciding whether a p-solution exists. The following proposition shows that this problem is the same as deciding whether a universal p-solution exists. This proposition follows immediately from Fagin

[7]RP comprises the sets that are efficiently recognizable by a randomized algorithm with a bounded one-sided error (i.e., the answer may mistakenly be "no"). NP = RP is equivalent to NP $\subseteq$ BPP [Ko 1982] (where BPP comprises the sets that are efficiently recognizable by a randomized algorithm with a bounded two-sided error) and implies that BPP contains the whole polynomial hierarchy [Zachos 1988].

[8]#P [Valiant 1979] is the class of functions that count the number of accepting paths of the input of an NP machine.

[9]Using an oracle to a #P-hard (or $\text{FP}^{\#\text{P}}$-hard) function, one can efficiently solve every problem in the polynomial hierarchy [Toda and Ogiwara 1992].

et al. [2005a] (showing that for standard schema mappings, a solution exists if and only if a universal solution exists) and Proposition 4.3.

PROPOSITION 5.3. *Let $\mathcal{M}$ be a standard schema mapping. For a source p-instance $\tilde{\mathcal{I}}$, a p-solution exists if and only if a universal p-solution exists.*

The problem of deciding whether a p-solution exists corresponds to the rows of Table I entitled "Existence of (U.) p-Solutions." Recall that "trivial" means that a p-solution always exists. These are the cases where Σ is the union of a set of st-tgds and a weakly acyclic set of t-tgds (and Σ has no t-egds). Observe that for tuple-independent instances, existence of p-solutions is always tractable or trivial. For DNF instances, however, the nontrivial cases are coNP-complete, except for the tractable case where Σ contains full st-tgds and t-egds. Let us now prove some of these complexity results.

We first consider source tuple-independent instances. Such a source instance I_p^α has the special property that I itself is feasible (due to our requirement that $p(e) > 0$ for all $e \in \mathsf{EVar}(\alpha)$). The following proposition shows how this fact can be used for efficiently testing for the existence of a p-solution.

PROPOSITION 5.4. *Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a standard schema mapping. Given a source tuple-independent instance I_p^α, a p-solution exists if and only if a solution for I exists. Hence, due to Fagin et al. [2005a], testing whether a p-solution exists is in polynomial time.*

PROOF. Since $(\mathbf{S}, \mathbf{T}, \Sigma)$ is a standard schema mapping, it follows easily that a solution for a source instance I is also a solution for every source instance $I' \subseteq I$. This, along with Proposition 4.3, gives the "if" part. The "only if" part is due to Proposition 4.3 and the fact that I is feasible. □

Next, we consider DNF instances. The following proposition shows that (or of a universal p-solution) is in coNP. This proposition is an immediate corollary of Proposition 4.3 and the fact that, in ordinary data exchange, the existence of a solution can be tested in polynomial time (when the schema mapping is standard) [Fagin et al. 2005a].

PROPOSITION 5.5. *Let $\mathcal{M}$ be a standard schema mapping. Given a source DNF instance I_p^α, a p-solution exists if and only if a solution for I_τ^α exists for all feasible truth assignments τ. Thus, due to Fagin et al. [2005a], deciding whether a p-solution exists is in* coNP.

The following theorem shows the existence of a schema mapping that contains only st-tgds and t-egds (in which case a solution does not necessarily exist), where testing whether a p-solution exists is coNP-complete.

THEOREM 5.6. *There is a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$, such that Σ has only st-tgds and t-egds, and deciding whether a p-solution exists, given a source DNF instance I_p^α, is* coNP*-complete.*

PROOF. Membership in coNP is shown in Proposition 5.5. To prove coNP-hardness, we will show a reduction from the complement of SAT, namely, the problem of deciding whether a given CNF formula is not satisfiable. Let $\varphi = c_1 \wedge \cdots \wedge c_m$ be an instance of SAT (where each c_i is a disjunction of literals). Assume, without loss of generality, that the variables of φ belong to EVar. We will construct a fixed schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ that is independent of φ, while φ will define the source DNF instance I_p^α. The construction is as follows.

The source schema $\mathbf{S}$ has two binary relation symbols $E_\mathbf{S}$ and $L_\mathbf{S}$. For all integers i where $1 \leq i \leq m$, the instance I contains the fact $f_i = E_\mathbf{S}(i-1, i)$, and the condition $\alpha(f_i)$ is the conjunct c_i (which is a disjunction of literals). In addition, I contains the

facts $L_{\mathbf{S}}(0, \mathtt{a})$ and $L_{\mathbf{S}}(m, \mathtt{b})$, each with the condition **true**. The function p assigns to each variable of φ the probability 0.5.

Similarly to $\mathbf{S}$, the target schema $\mathbf{T}$ contains two binary relation symbols $E_{\mathbf{T}}$ and $L_{\mathbf{T}}$. The set Σ contains the following dependencies.

—d_E^{st}: $\forall x, y(E_{\mathbf{S}}(x, y) \rightarrow E_{\mathbf{T}}(x, y))$
—d_{EL}^{st}: $\forall x, y(E_{\mathbf{S}}(x, y) \rightarrow \exists u_1, u_2(L_{\mathbf{T}}(x, u_1) \wedge L_{\mathbf{T}}(y, u_2)))$
—d_L^{st}: $\forall x, u(L_{\mathbf{S}}(x, u) \rightarrow L_{\mathbf{T}}(x, u))$
—d_l^{egd}: $\forall x, u, v(L_{\mathbf{T}}(x, u) \wedge L_{\mathbf{T}}(x, v) \rightarrow u = v)$
—d_e^{egd}: $\forall x, y, u, v\ (E_{\mathbf{T}}(x, y) \wedge L_{\mathbf{T}}(x, u) \wedge L_{\mathbf{T}}(y, v) \rightarrow u = v)$

Next, we will show that there is a p-solution if and only if φ is not satisfiable. First, note that every truth assignment for φ is a feasible truth assignment for α. Consider such a truth assignment τ. We denote by G_τ the directed graph with the node set $\{0, 1, \ldots, m\}$ and the edge $(i-1, i)$ for each conjunct c_i of φ that is satisfied by τ. Note that the $E_{\mathbf{S}}$ relation of the instance I_τ^α contains exactly the edges of G_τ. Also, observe that G_τ contains a path from 0 to m if and only if all the conjuncts of φ are satisfied (i.e., τ is a satisfying assignment for φ). So, by Proposition 5.5, it is enough to prove the following. There exists a solution for I_τ^α if and only if G_τ does not contain a path from 0 to m. This is done next.

We view a solution for I_τ^α (when it exists) as a directed graph G', where the edges are given by $E_{\mathbf{T}}$. In addition, some of the nodes of G' are labeled, where the labeling function is given by $L_{\mathbf{T}}$. The t-egd d_l^{egd} requires $L_{\mathbf{T}}$ to act as a function.[10] The st-tgd d_E^{st} implies that the solution G' must contain all the edges of G_τ, and the st-tgd d_{EL}^{st} implies that $L_{\mathbf{T}}$ must label all the nodes that are incident to the edges of G'. The st-tgd d_L^{st}, together with the deterministic $L_{\mathbf{S}}$-relation of I, implies that in G', the label of 0 is a and that of m is b. The t-egd d_e^{egd} says that if G' has an edge from node v_1 to node v_2 and both nodes are labeled, then v_1 and v_2 should have the same label. In particular, the labeling function must propagate through paths of G_τ. So, if G_τ contains a path from 0 to m, then we get a contradiction to the fact that 0 and m have different labels. On the other hand, if G_τ does not contain a path from 0 to m, then we can obtain G' from G_τ by labeling all the nodes that are reachable from 0 with a, and the rest of the nodes with b. It follows that a solution G' exists if and only if no path of G_τ leads from 0 to m. □

5.3.3. Materializing a (Universal) p-Solution. The second problem corresponds to the rows entitled "Materialize a p-Solution," and is that of materializing a candidate p-solution as a DNF instance. Recall that a candidate p-solution is a target p-instance $\tilde{\mathcal{J}}$ that forms a p-solution if one exists. The third problem is the universal version of the second, namely, generation of a candidate universal p-solution as a DNF instance, and it corresponds to the rows entitled "Materialize a U. p-Solution."

Table I shows that, for source DNF instances, we can sometimes efficiently materialize a candidate universal p-solution (e.g., when Σ has only st-tgds) whereas in other cases we cannot efficiently materialize even a (not necessarily universal) candidate p-solution. If source instances are tuple-independent, then materializing a candidate p-solution is always tractable. However, for materializing candidate universal p-solutions, the intractable cases for source DNF instances remain intractable for tuple-independent instances. Next, we give some proofs, and we focus on the problem of materializing a universal p-solution.

[10] Actually, we do not really need d_l^{egd}; it is given just for the clarity of the proof.

The positive results are obtained by combining the chase algorithm [Beeri and Vardi 1984; Fagin et al. 2005a; Maier et al. 1979] with the known concept of maintaining conditions (or provenance) in relational operators, which is used in Green and Tannen [2006], Green et al. [2007b], and Imielinski and Lipski [1984] for showing closure of annotated databases under relational algebra.[11] More details can be found in Section A.2 of the Appendix.

The lower bounds are proved using the inapproximability of determining the number of assignments satisfying a monotone 2-CNF formula (see, e.g., Zuckerman [1996]), and the Monte-Carlo algorithm of Karp et al. [1989] as a reduction technique. As an example, we will now prove the existence of a schema mapping that contains only full st-tgds and full t-tgds, such that it is intractable to materialize a universal p-solution as a DNF instance, given a source tuple-independent instance.

THEOREM 5.7. *Assume* RP $\neq$ NP. *Then there is a schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma)$*, such that* Σ *has only full st-tgds and full t-tgds, and there is no polynomial-time algorithm that constructs a universal p-solution as a DNF instance* J_q^β*, given a tuple-independent instance* I_p^α.

PROOF. We will construct a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ (that contains only full st-tgds and full t-tgds), and show a reduction from the problem #monotone-2SAT, which is defined as follows. A formula in CNF is *monotone* if all its literals are positive, that is, it does not include negations. (Note that such a formula is trivially satisfiable.) The problem #monotone-2SAT is that of counting the number of satisfying assignment for a 2-CNF formula. It is well known that for all positive constants θ, this problem has no efficient randomized θ-approximation,[12] unless RP $\neq$ NP (see, e.g., Zuckerman [1996]).

So, let $\varphi = c_1 \wedge \cdots \wedge c_m$ be a monotone 2-CNF formula. Without loss of generality, assume that the variables of φ are the event variables $e_1, \ldots, e_n$.

We first construct the source schema $\mathbf{S}$ and the source tuple-independent instance I_p^α. The source schema $\mathbf{S}$ contains a unary relation symbol $V_\mathbf{S}$ (storing the variables), and a quaternary relation symbol $C_\mathbf{S}$ (storing the conjuncts). The tuple-independent instance I_p^α contains:

—The fact $V_\mathbf{S}(i)$ with $\alpha(V_\mathbf{S}(i)) = e_i$, for each variable e_i.
—The fact $C_\mathbf{S}(i_1, i_2, j-1, j)$, with $\alpha(C_\mathbf{S}(i_1, i_2, j-1, j)) = \mathbf{true}$, for each conjunct $c_j = e_{i_1} \vee e_{i_2}$.[13]

The function p assigns the probability 0.5 to each e_i.

To complete the definition of the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$, the schema $\mathbf{T}$ consists of one binary symbol $E_\mathbf{T}$, and Σ contains the following dependencies.

—d_l^{st}: $\forall v, u, x, y(V_\mathbf{S}(v) \wedge C_\mathbf{S}(v, u, x, y) \rightarrow E_\mathbf{T}(x, y))$
—d_r^{st}: $\forall v, u, x, y(V_\mathbf{S}(u) \wedge C_\mathbf{S}(v, u, x, y) \rightarrow E_\mathbf{T}(x, y))$
—d_E^{ttgd}: $\forall x, y, z(E_\mathbf{T}(x, y) \wedge E_\mathbf{T}(y, z) \rightarrow E_\mathbf{T}(x, z))$.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, let $\tilde{\mathcal{J}}$ be a p-solution for p-space(I_p^α), and let $\tilde{\mathcal{P}}$ be a $\mathsf{USOL}_\mathcal{M}$-match of p-space$(I_p^\alpha)$ in $\tilde{\mathcal{J}}$. Consider a pair (I_τ^α, J') in the support of $\tilde{\mathcal{P}}$. The combination of the dependencies d_l^{st} and d_r^{st} say that J' (being a solution for I_τ^α) must include the fact $E_\mathbf{T}(j-1, j)$ if the source contains a fact of the form $C_\mathbf{S}(i_1, i_2, j-1, j)$ and least one

[11] A similar construction is used in Green et al. [2007a] for the task of propagating *trust conditions* through data exchange between peers in a network.
[12] We say that r is a θ-approximation of s if $s/(1+\theta) \leq r \leq (1+\theta)s$
[13] More precisely, since the condition **true** is technically not allowed in a tuple-independent instance, $\alpha(C_\mathbf{S}(i_1, i_2, j-1, j))$ is a fresh event variable e such that $p(e) = 1$.

of $V_{\mathbf{S}}(i_1)$ and $V_{\mathbf{S}}(i_2)$; that is, J' must include $E_{\mathbf{T}}(j-1, j)$ if τ satisfies the conjunct c_j. Thus, J' contains (at least) all the facts $E_{\mathbf{T}}(j-1, j)$ such that c_j is satisfied by τ. The dependency d_E^{ttgd} says that $E_{\mathbf{T}}$ is transitively closed. Thus, since J' is a solution, it must contain the fact $E_{\mathbf{T}}(0, m)$ if τ satisfies φ (i.e., every c_j). If, on the other hand, τ violates φ, then J' does not include $E_{\mathbf{T}}(0, m)$, since J' is universal and there is a solution for I_τ^α that does not include $E_{\mathbf{T}}(0, m)$ (which is easy to come up with).

We conclude that a random pair (I_τ^α, J') in the support of $\tilde{\mathcal{P}}$ is such that $E_{\mathbf{T}}(0, m) \in J'$ if and only if τ satisfies φ. Since the marginals of $\tilde{\mathcal{P}}$ are p-space(I_p^α) and $\tilde{\mathcal{J}}$, we conclude that the probability that $\tilde{\mathcal{J}}$ contains $E_{\mathbf{T}}(0, m)$ is equal to the probability that a randomly chosen truth assignment for $\mathsf{EVar}(\alpha)$ satisfies φ. Since every truth assignment has the same probability, 2^{-n}, the probability that a randomly chosen truth assignment satisfies φ is exactly $\#\varphi/2^n$, where $\#\varphi$ is the number of satisfying assignments for φ. Thus, we have the following.

$$\#\varphi = 2^n \cdot \Pr(E_{\mathbf{T}}(0, m) \in \mathcal{J}) \tag{24}$$

Now, let J_q^β be DNF instance that forms a universal p-solution for p-space(I_p^α), and let $\tilde{\mathcal{J}} = \text{p-space}(J_q^\beta)$. We will show that there is a polynomial-time θ-approximation for every positive θ (actually, an FPRAS) for determining the probability that $\mathcal{J}$ contains the fact $E_{\mathbf{T}}(0, m)$. If this fact is not in J, then the probability is 0. Otherwise, we need to approximate the probability that the DNF formula $\beta(E_{\mathbf{T}}(0, m))$ is satisfied by a truth assignment randomly chosen according to q. This can be done by a rather straightforward adaptation (as done in, e.g., Dalvi and Suciu [2007b] and Kimelfeld et al. [2008]) of the Monte-Carlo technique of Karp et al. [1989] for approximating the number of satisfying assignments for a DNF formula. In particular, if we could obtain J_q^β efficiently, we would obtain an efficient randomized θ-approximation for $\Pr(E_{\mathbf{T}}(0, m) \in \mathcal{J})$. But then, multiplying this probability by 2^n would obtain an efficient randomized θ-approximation for $\#\varphi$, since by (24) we have

$$(1+\theta)\#\varphi = 2^n \cdot (1+\theta) \cdot \Pr(E_{\mathbf{T}}(0, m) \in \mathcal{J})$$

and, similarly,

$$\frac{\#\varphi}{1+\theta} = \frac{2^n \cdot \Pr(E_{\mathbf{T}}(0, m) \in \mathcal{J})}{1+\theta}.$$

This contradicts the assumption that RP $\neq$ NP. □

5.3.4. Answering Target UCQs. The fourth problem is that of evaluating unions of conjunctive queries, and it corresponds to the rows of Table I entitled "Target UCQ: Exact." Formally, for a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ and a UCQ Q over $\mathbf{T}$, the problem is the following. Given a source DNF instance I_p^α and a tuple $\mathbf{a}$ of constants, compute $conf_Q(\mathbf{a})$. As shown in the table, in every studied case (even when there are only full st-tgds and source instances are tuple-independent), there is a schema mapping such that this problem is FP$^{\#\text{P}}$-complete. Actually, we can show even more: in the most restricted case (source tuple-independent instances and only full st-tgds), for every *nontrivial* target UCQ Q there exists a schema mapping such that evaluating Q is FP$^{\#\text{P}}$-hard, where a *trivial* UCQ is a Boolean UCQ that is equivalent to **true**.

PROPOSITION 5.8. *Assume that* $\mathbf{T}$ *is a schema, and that* Q *is a nontrivial UCQ over* $\mathbf{T}$. *There exists a schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *where* Σ *has only full st-tgds, such that computing* $conf_Q(\mathbf{a})$*, given a source tuple-independent instance* I_p^α *and a tuple* $\mathbf{a}$*, is* FP$^{\#\text{P}}$*-complete.*

PROOF. Membership in $\mathrm{FP}^{\#\mathrm{P}}$, for all UCQs and standard schema mappings, is shown in Proposition A.13 in the Appendix. Here, we will construct a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$, comprising only full st-tgds, such that computing $conf_Q(\mathbf{a})$ is $\mathrm{FP}^{\#\mathrm{P}}$-hard.

Let $\mathbf{S}_h$ be a schema, and let Q_h be a Boolean CQ over $\mathbf{S}_h$ such that the following hold. First, Q_h has no constants. Second, evaluating Q_h (i.e., computing the probability that it is true) is $\mathrm{FP}^{\#\mathrm{P}}$-hard over tuple-independent instances. Existence of such queries is shown in Dalvi and Suciu [2007a, 2007b]. We will show a reduction from evaluating Q_h over tuple-independent instances K_r^γ to evaluating the UCQ Q over source tuple-independent instances (in a schema mapping).

We denote by $\mathbf{S_T}$ the schema that is obtained from $\mathbf{T}$ by replacing each relation symbol R with a unique relation symbol, which we denote by R^s, that has the same arity as R. We assume that $\mathbf{S}_h$ and $\mathbf{S_T}$ do not have any common relation symbol. The schema $\mathbf{S}$ is $\langle \mathbf{S}_h, \mathbf{S_T} \rangle$. The set Σ of dependencies consists of the following st-tgd for each relation symbol R of $\mathbf{T}$.

$$\forall \mathbf{x}, \mathbf{y}(Q_h(\mathbf{x}) \wedge R^s(\mathbf{y}) \rightarrow R(\mathbf{y})).$$

Let $\mathbf{a}$ and $J_\mathbf{a}$ be a tuple of constants and an instance over $\mathbf{T}$, respectively, such that $\mathbf{a} \in Q(J_\mathbf{a})$. There is such a choice of $\mathbf{a}$ and $J_\mathbf{a}$, because Q, like every UCQ, is not identically **false**. We define $I_\mathbf{a}$ to be the instance over $\mathbf{S_T}$ that is obtained from $J_\mathbf{a}$ by replacing each fact $R(\mathbf{t})$ with $R^s(\mathbf{t})$.

Given a tuple-independent instance K_r^γ over $\mathbf{S}_h$, the source tuple-independent instance I_p^α over $\mathbf{S}$ is defined as follows. The instance I is the union $K \cup I_\mathbf{a}$. The set $\mathsf{EVar}(\alpha)$ and the function p are equal to $\mathsf{EVar}(\gamma)$ and r, respectively. Finally, the function α is the same as γ over the facts of K, and is **true** over the facts of $I_\mathbf{a}$.

We have defined the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ and the source tuple-independent instance I_p^α. Now, consider a feasible truth assignment τ for $\mathsf{EVar}(\gamma)$. Note that I_τ^α contains all the facts of $I_\mathbf{a}$, and that $Q_h(I_\tau^\gamma) = Q_h(K_\tau^\gamma)$. If $Q_h(K_\tau^\gamma) = \mathbf{true}$, then the st-tgds in Σ imply that every solution J for I_τ^α must contain $J_\mathbf{a}$ (since (a) each fact of $I_\mathbf{a}$ is in I_τ^α, (b) each R^s is copied to R because of the st-tgds, and (c) the R^s relation of $I_\mathbf{a}$ has the same tuples as the R relation of $J_\mathbf{a}$). Hence, $\mathbf{a} \in Q(J)$. Therefore, if $Q_h(K_\tau^\gamma) = \mathbf{true}$, then $\mathbf{a} \in \mathrm{certain}(Q, I_\tau^\alpha, \Sigma)$. On the other hand, if $Q_h(K_\tau^\gamma) = \mathbf{false}$, then the empty relation $J_\emptyset$ is a solution for I_τ^α. But then, $\mathbf{a} \notin Q(J_\emptyset)$ because Q is nontrivial, hence $\mathbf{a}$ is not a certain answer. From Proposition 4.12, we conclude that

$$conf_Q(\mathbf{a}) = \mathrm{Pr}_{\tilde{\mathcal{K}}}(Q_h(\mathcal{K}) = \mathbf{true}),$$

where $\tilde{\mathcal{K}}$ is p-space(K_r^γ). Thus, $\mathrm{Pr}_{\tilde{\mathcal{K}}}\big(Q_h(\mathcal{K}) = \mathbf{true}\big)$ can be efficiently computed by an oracle that evaluates Q over I_p^α under the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$, as required. □

Given this intractability, the best that one can hope for when looking for tractable classes of schema mappings (in terms of target-query evaluation) is an evaluation in an approximate manner; in practice, such an evaluation is often good enough. So, the fifth problem is that of approximately evaluating target UCQs, and it is considered in the rows of Table I entitled "Target UCQ: Approx." Formally, let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping, and let Q be a UCQ over $\mathbf{T}$. For a fixed number $\theta > 0$, a *randomized θ-approximation for Q* is a randomized algorithm A that gets as input a DNF instance I_p^α over $\mathbf{S}$ and a tuple $\mathbf{a}$, and returns a (random) value $A(I_p^\alpha, \mathbf{a})$ such that

$$\mathrm{Pr}_A\left(\frac{p'}{1+\theta} \leq A(I_p^\alpha, \mathbf{a}) \leq (1+\theta)p'\right) \geq \frac{2}{3}, \tag{25}$$

where $p' = conf_Q(\mathbf{a})$.[14] A *fully polynomial randomized approximation scheme* (abbreviated *FPRAS*) for Q is defined similarly, except that in addition to I_p^α and $\mathbf{a}$, the algorithm gets θ as input (hence, θ is not assumed to be fixed), and the algorithm is required to run in polynomial time in the size of I_p^α and in $1/\theta$. An even stronger notion is that of *FPAS*, where the approximation algorithm is deterministic (i.e., the reliability factor $2/3$ is replaced with 1).

Table I shows that for source DNF instances, there is an FPRAS for a UCQ when Σ contains only st-tgds, or full st-tgds with t-egds. For such Σ, there is even an FPAS if source instances are tuple-independent. For the rest of the studied cases, there is always a schema mapping Σ and a UCQ Q such that for all $\theta > 0$, no randomized θ-approximation for Q exists unless RP = NP. Moreover, this holds for *all* nontrivial UCQs Q, except for the cell of the column entitled "st-tgds, t-egds" in the bottom row of the table (in the "tuple-independent" part), where this result holds for all UCQs Q except for "near-trivial" ones. A UCQ Q over a schema $\mathbf{T}$ is *near-trivial* if it is a statement about non-emptiness of the relations; more precisely, it is a Boolean UCQ such that $Q(J_1) = Q(J_2)$ whenever J_1 and J_2 are instances such that for each relation symbol R of $\mathbf{T}$, we have that $R^{J_1} = \emptyset$ if and only if $R^{J_2} = \emptyset$. For example, it is easy to see that the UCQ $R(x, y) \vee S(x)$ is near-trivial. Note that this notion is weaker (less restrictive) than UCQ triviality; this weakening is necessary, since it can be shown that over tuple-independent source instances, there is an FPAS for every near-trivial UCQ if Σ contains only st-tgds and t-egds. We will next prove hardness for the case the where the schema mapping has only full st-tgds and full t-tgds.

THEOREM 5.9. *Assume* RP $\neq$ NP. *Let Q be a nontrivial UCQ over a schema $\mathbf{T}$. There is a schema mapping $(\mathbf{S}, \mathbf{T}', \Sigma)$, where $\mathbf{T} \subseteq \mathbf{T}'$, such that Σ contains only full st-tgds and full t-tgds, and for all $\theta > 0$, there is no polynomial-time randomized θ-approximation for Q over source tuple-independent instances.*

PROOF. In the proof of Theorem 5.7, we showed a reduction from #monotone-2SAT. We will show how to combine this proof with that of Proposition 5.8 in order to construct the schema mapping $(\mathbf{S}, \mathbf{T}', \Sigma)$ of the theorem.

Let $(\mathbf{S}_{\text{sat}}, \mathbf{T}_{\text{sat}}, \Sigma_{\text{sat}})$ be the schema mapping constructed in the proof of Theorem 5.7. For a given monotone 2-CNF formula φ, let I_p^α be the source tuple-independent instance (over $\mathbf{S}_{\text{sat}}$) constructed there. We assume (without loss of generality) that the schemas $\mathbf{T}_{\text{sat}}$ (defined previously) and $\mathbf{T}$ (specified in the theorem currently being proved) are disjoint.

Now, consider the proof of Proposition 5.8. For the query Q over $\mathbf{T}$, let $\mathbf{S_T}$, $\mathbf{a}$, $J_\mathbf{a}$ and $I_\mathbf{a}$ be defined as in that proof. We also use the definition of R^s of $\mathbf{S_T}$ for every relation symbol R of $\mathbf{T}$. We assume that $\mathbf{S}_{\text{sat}}$ and $\mathbf{S_T}$ are disjoint.

To construct the source schema $\mathbf{S}$, we take the concatenation $\langle \mathbf{S}_{\text{sat}}, \mathbf{S_T} \rangle$ and add a unique binary relation symbol $\hat{E}_\mathbf{S}$. The source tuple-independent instance K_r^γ over $\mathbf{S}$ is defined as follows: The instance K consists of all the facts of I and all those of $I_\mathbf{a}$. In addition, K includes the fact $\hat{E}_\mathbf{S}(0, m)$, where m is as in the proof of Theorem 5.7 (the number of conjuncts of the formula φ). The function γ is the same as α for the facts of I, and is **true** for the rest of the facts. Finally, the function r is equal to p (i.e., $r(e) = p(e)$ for all $e \in \mathsf{EVar}(\gamma)$).

To construct the target schema $\mathbf{T}'$, we first take the concatenation $\langle \mathbf{T}, \mathbf{T}_{\text{sat}} \rangle$. We then add a unique relation symbol $\hat{R}$ for each relation R of $\mathbf{T}$, where $\hat{R}$ has the same arity as R. Finally, we add a unique binary relation symbol $\hat{E}_\mathbf{T}$.

[14]Note that the choice of the reliability factor $2/3$ is arbitrary, since one can improve it to $(1 - \delta)$ by taking the median of $O(\log \delta)$ trials [Jerrum et al. 1986].

Next, we construct Σ. We begin with $\Sigma = \Sigma_{\mathsf{sat}}$. We then add to Σ the following st-tgd d_R^{st} for each relation symbol R of $\mathbf{T}$; this dependency copies the relation R^s of $\mathbf{S_T}$ to the relation $\hat{R}$ of $\mathbf{T}'$.

$$d_R^{\mathrm{st}} : \forall \mathbf{x}(R^s(\mathbf{x}) \rightarrow \hat{R}(\mathbf{x})).$$

The st-tgd $d_{\hat{E}}^{\mathrm{st}}$ copies $\hat{E}_{\mathbf{S}}$ to $\hat{E}_{\mathbf{T}}$:

$$d_{\hat{E}}^{\mathrm{st}} : \forall x, y(\hat{E}_{\mathbf{S}}(x, y) \rightarrow \hat{E}_{\mathbf{T}}(x, y)).$$

Finally, for each relation symbol R of $\mathbf{T}$, the t-tgd d_R^{ttgd} copies $\hat{R}$ to R if $\hat{E}_{\mathbf{T}}$ and $E_{\mathbf{T}}$ have a common tuple.

$$d_R^{\mathrm{ttgd}} : \forall \mathbf{x}, \mathbf{y}(\hat{E}_{\mathbf{T}}(\mathbf{y}) \wedge E_{\mathbf{T}}(\mathbf{y}) \wedge \hat{R}(\mathbf{x}) \rightarrow R(\mathbf{x})).$$

We will show that $\mathbf{a}$ is a certain answer for a random instance K_τ^γ, where τ is a truth assignment for φ, if and only if τ satisfies φ. Note that a solution always exists, since Σ contains only st-tgds and full t-tgds [Fagin et al. 2005a]. Then, by Proposition 4.12, the probability that $\mathbf{a}$ is a certain answer is exactly $conf_Q(\mathbf{a})$. Since every truth assignment has the probability 2^{-n} (where n is the number of variables in φ), we have that $\#\varphi = 2^n \cdot conf_Q(\mathbf{a})$. As was shown in the proof of Theorem 5.7, this suffices for showing that no approximation is efficient (assuming RP $\neq$ NP).

Let τ be a truth assignment for φ. Then the random instance K_τ^γ contains the fact $\hat{E}_{\mathbf{S}}(0, m)$ and every fact $R^s(\mathbf{t})$ where $R(\mathbf{t}) \in J_{\mathbf{a}}$. Therefore, from the dependencies $d_{\hat{E}}^{\mathrm{st}}$ and d_R^{st}, it follows that every solution J for a random K_τ^γ must contain the fact $\hat{E}_{\mathbf{T}}(0, m)$ and all the facts $\hat{R}(\mathbf{t})$ where $R(\mathbf{t}) \in J_{\mathbf{a}}$. Also, recall from the proof of Theorem 5.7 that a universal solution for I_τ^α contains the fact $E_{\mathbf{T}}(0, m)$ if and only if τ satisfies φ.

Suppose first that τ satisfies φ, and let J be a solution for K_τ^γ. Then, J contains $E_{\mathbf{T}}(0, m)$ (since J contains a solution for I_τ^α), and hence $E_{\mathbf{T}}(\mathbf{y}) \wedge \hat{E}_{\mathbf{T}}(\mathbf{y})$ is satisfied in each d_R^{ttgd}, implying that J contains $J_{\mathbf{a}}$. Therefore, $\mathbf{a}$ is an answer for Q over J. As J is an arbitrary solution for K_τ^γ, we get that $\mathbf{a}$ is a certain answer.

Now, suppose that τ violates φ. In this case, we construct a solution J as follows. Let J_{sat} be a universal solution for I with respect to Σ_{sat}. As mentioned previously, J_{sat} does not contain $E_{\mathbf{T}}(0, m)$. Then, the relation J is obtained from J_{sat} by adding the facts $\hat{R}(\mathbf{t})$ for each $R(\mathbf{t}) \in J_{\mathbf{a}}$, and the fact $\hat{E}_{\mathbf{T}}(0, m)$. It is easy to see that J is, indeed, a solution for K_τ^γ with respect to Σ. In particular, we do not need to add any fact of $J_{\mathbf{a}}$ to J since the premise of each d_R^{ttgd} is violated. Therefore, $\mathbf{a} \notin Q(J)$ (since Q is nontrivial) and hence $\mathbf{a}$ is not a certain answer, as claimed. □

6. PROBABILISTIC MAPPINGS

In this section, we generalize the framework and results of the previous sections to accommodate uncertainty in the schema mapping. More formally, in this generalization not only is the source data probabilistic, but the set of dependencies specifying the schema mapping is probabilistic as well. Moreover, we will allow the source p-instance and the probabilistic mapping to be arbitrarily correlated. Next, we give the basic definitions. Later, we discuss the generalization of the results of the previous sections to this new setting.

Let $\mathbf{S}$ and $\mathbf{T}$ be two schemas with no relation symbols in common. We assume that there is a fixed countably infinite set $\mathsf{Dep}_{\mathbf{ST}}$ of formulas over $\langle \mathbf{S}, \mathbf{T} \rangle$, such that every set Σ of dependencies specifying a schema mapping is a finite subset of $\mathsf{Dep}_{\mathbf{ST}}$. We denote by $\mathsf{Dep}_{\mathbf{ST}}^*$ the (countable) set of all finite subsets Σ of $\mathsf{Dep}_{\mathbf{ST}}$.

Up until now, we considered schema mappings that are specified by triples $(\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma \in \mathsf{Dep}^*_{\mathbf{ST}}$. Here, as a starting point, we are interested in replacing the fixed Σ with a p-space $\tilde{\Sigma}$ over $\mathsf{Dep}^*_{\mathbf{ST}}$. Thus, both the source instance $\tilde{\mathcal{I}}$ and the schema mapping $(\mathbf{S}, \mathbf{T}, \tilde{\Sigma})$ are probabilistic. However, separating the probabilistic schema mapping from the source p-instance necessitates the assumption of probabilistic independence (or some other specific correlation) between the two. In practice, such an assumption is often a limitation. Therefore, in this section, we do not separate the probabilistic instance from the probabilistic schema mapping; instead, we use a generalized definition that is based on the notion of a probabilistic problem (abbreviated p-problem). The formal definition is the following.

Definition 6.1 (*p-Problem*). Let $\mathbf{S}$ and $\mathbf{T}$ be schemas without common relation symbols. A *p-problem* (*from* $\mathbf{S}$ *to* $\mathbf{T}$) is a p-space $\tilde{\mathcal{P}}$ over $\mathsf{Dep}^*_{\mathbf{ST}} \times \mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$.

Observe that the marginals of a p-problem $\tilde{\mathcal{P}}$ define a unique probabilistic schema mapping $(\mathbf{S}, \mathbf{T}, \tilde{\Sigma})$ and a unique source p-instance $\tilde{\mathcal{I}}$; however, $\tilde{\mathcal{P}}$ is not necessarily the product space of $\tilde{\Sigma}$ and $\tilde{\mathcal{I}}$.

A p-solution $\tilde{\mathcal{J}}$ for a p-problem $\tilde{\mathcal{P}}$ is defined similarly to the case of a fixed Σ, except that now the probabilistic match is from $\tilde{\mathcal{P}}$ to $\tilde{\mathcal{J}}$ (rather than from the source p-instance $\tilde{\mathcal{I}}$ to $\tilde{\mathcal{J}}$). Formally, given a p-problem $\tilde{\mathcal{P}}$ from $\mathbf{S}$ to $\mathbf{T}$, a target p-instance $\tilde{\mathcal{J}}$ is a *p-solution* (for $\tilde{\mathcal{P}}$) if there is a $\mathsf{dSOL}_{\mathbf{ST}}$-match of $\tilde{\mathcal{P}}$ in $\tilde{\mathcal{J}}$, where $\mathsf{dSOL}_{\mathbf{ST}}$ is the binary relation between pairs (Σ, I) and instances J, such that $I \in \mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$, $J \in \mathsf{Inst}(\mathbf{T})$, $\Sigma \in \mathsf{Dep}^*_{\mathbf{ST}}$, and $\langle I, J\rangle \models \Sigma$. (The letter d in $\mathsf{dSOL}_{\mathbf{ST}}$ denotes that dependencies are involved in the relation.) Similarly, $\tilde{\mathcal{J}}$ is a *universal p-solution* (for $\tilde{\mathcal{P}}$) if there is a $\mathsf{dUSOL}_{\mathbf{ST}}$-match of $\tilde{\mathcal{P}}$ in $\tilde{\mathcal{J}}$, where $\mathsf{dUSOL}_{\mathbf{ST}}$ is the relation between pairs (Σ, I) and instances J such that J is a universal solution for I with respect to Σ.

6.1. Generalization of the Results

We now discuss the generalization of our results to the notion of a p-problem. Basically, *all* the results generalize to p-problems. For Sections 3 and 4, this generalization is via a rather mechanical replacement of the p-space $\tilde{\mathcal{I}}$ with the p-space $\tilde{\mathcal{P}}$. Generalizing the results of Section 5 is a little more involved.

We start with the results of Sections 3 and 4. In Theorem 3.6, we need to replace every occurrence of $\tilde{\mathcal{I}}$ with $\tilde{\mathcal{P}}$ and, in addition, the event E of Part 2 is a subset of $\mathsf{Dep}^*_{\mathbf{ST}} \times \mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$ (rather than $\mathsf{Inst}^{\mathsf{c}}(\mathbf{S})$). In Proposition 4.3, the probability space $\tilde{\mathcal{I}}$ is replaced with a p-problem $\tilde{\mathcal{P}}$, and $I \in \Omega_+(\tilde{\mathcal{I}})$ is replaced with $(\Sigma, I) \in \Omega_+(\tilde{\mathcal{P}})$. In Theorem 4.10, we replace the source instance $\tilde{\mathcal{I}}$ with a p-problem $\tilde{\mathcal{P}}$; moreover, the sets $\mathsf{SOL}_{\mathcal{M}}$ and $\mathsf{USOL}_{\mathcal{M}}$ are replaced with $\mathsf{dSOL}_{\mathbf{ST}}$ and $\mathsf{dUSOL}_{\mathbf{ST}}$, respectively. In Proposition 4.12 the probability space $\tilde{\mathcal{I}}$ is replaced with $\tilde{\mathcal{P}}$; that is, $conf_Q(\mathbf{a})$ is equal to the probability that a random pair (Σ, I) of $\tilde{\mathcal{P}}$ is such that $\mathbf{a}$ is a certain answer (i.e., $\mathbf{a} \in \mathrm{certain}(Q, I, \Sigma)$). Finally, Proposition 4.14 generalizes, again, by simply replacing $\tilde{\mathcal{I}}$ with $\tilde{\mathcal{P}}$.

We now show how the results of Section 5 are generalized. For that, we need to explain how a p-problem is encoded. Recall that a source p-instance is encoded as a DNF instance I^{α}_{p}. We use a similar encoding for a probabilistic mapping. That is, every dependency σ is assigned a DNF formula over EVar (namely, a condition) and each variable is given a probability in $[0, 1]$. Formally, a *DNF schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma^{\gamma}_{r})$ comprises source and target schemas $\mathbf{S}$ and $\mathbf{T}$ (without common relation symbols), a set $\Sigma \in \mathsf{Dep}^*_{\mathbf{ST}}$ of dependencies, a function γ that assigns to each $\sigma \in \Sigma$ a DNF formula $\gamma(\sigma)$ over EVar, and a function $r : \mathsf{EVar}(\gamma) \to [0, 1]$ (where, as usual, $\mathsf{EVar}(\gamma)$ is the set of all the event variables that appear in the image of γ). Now, we allow the DNF schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma^{\gamma}_{r})$ and the source DNF instance I^{α}_{p} to share events, that is, $\mathsf{EVar}(\gamma)$ and $\mathsf{EVar}(\alpha)$ are not necessarily disjoint. In this case, we require r (the probability function

of Σ_r^γ) and p (the probability function of I_p^α) to agree on the common variables (i.e., $r(e) = p(e)$ for all $e \in \mathsf{EVar}(\gamma) \cap \mathsf{EVar}(\alpha)$).

A DNF schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma_r^\gamma)$ and a source DNF instance I_p^α naturally encode a finite p-problem from $\mathbf{S}$ to $\mathbf{T}$, where a sample (Σ', I') is obtained as follows. First, a random truth assignment $\tau : \mathsf{EVar}(\gamma) \cup \mathsf{EVar}(\alpha) \to \{\mathbf{true}, \mathbf{false}\}$ is chosen for all the event variables e of $(\mathbf{S}, \mathbf{T}, \Sigma_r^\gamma)$ and I_p^α, by independently picking a random Boolean value $\tau(e)$, such that the probability for **true** is $r(e)$ or $p(e)$ (depending on whether $e \in \mathsf{EVar}(\gamma)$ or $e \in \mathsf{EVar}(\alpha)$). Second, all the members of Σ and I having their condition satisfied by τ are selected as members of Σ' and I', respectively. We denote this probability space by p-space$(\Sigma_r^\gamma, I_p^\alpha)$. Observe that, since γ and α are allowed to have variables in common, the marginal source p-instance and probabilistic schema mapping are not necessarily probabilistically independent. Moreover, it is easy to show (e.g., by using the encoding of finite probabilistic databases given in Abiteboul and Senellart [2006] and Green and Tannen [2006]) that every finite p-problem can be represented as a combination of a DNF schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma_r^\gamma)$ and a source DNF instance I_p^α.

When analyzing the complexity of the problems considered in Section 5, we make the assumption that the DNF schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma_r^\gamma)$ is fixed (i.e., as in Section 5, data complexity is actually analyzed) and, moreover, that every sample $(\mathbf{S}, \mathbf{T}, \Sigma')$ (obtained by choosing a random truth assignment for $\mathsf{EVar}(\gamma)$) is a standard schema mapping, as defined in Section 5.3.1. Thus, as in Section 5, the input for a computational problem consists of a source DNF (or tuple-independent) instance I_p^α. Then, all the results of Table I remain correct. For example, if Σ contains only st-tgds and t-tgds (and our assumption about the samples $(\mathbf{S}, \mathbf{T}, \Sigma')$ of $(\mathbf{S}, \mathbf{T}, \Sigma_r^\gamma)$ holds), then testing whether a p-solution for the p-problem p-space$(\Sigma_r^\gamma, I_p^\alpha)$ exists, given a source DNF instance I_p^α, is coNP-complete, but it is solvable in polynomial time if I_p^α is a tuple-independent instance.

6.2. Probabilistic Mappings in the Literature

Data integration under uncertainty is studied in Dong et al. [2007, 2009], and Sarma et al. [2008a, 2009], where the schema mapping (which is called there a p-mapping) is probabilistic and finite, and the source data are deterministic. We can compare our basic notions to those of Dong et al. [2007, 2009], and Sarma et al. [2008a, 2009], by restricting our p-problem to a finite one where the source instance is deterministic.

Given a source instance I and a p-mapping $\tilde{\mathcal{M}}$, a *by-table solution*, as defined in Dong et al. [2009] and Sarma et al. [2008a], is a special case of what we call a p-solution, namely, a finite p-solution $\tilde{\mathcal{J}}$ such that there is a left-trivial $\mathsf{dSOL}_{\mathbf{ST}}$-match of the corresponding p-problem $\tilde{\mathcal{P}}$ (namely, the product space $\tilde{\mathcal{M}} \times I$) in $\tilde{\mathcal{J}}$. (See Section 3.1 for the definition of a left-trivial probabilistic match.) Thus, the notion of a by-table solution is more restrictive than our notion of a p-solution (even if we restrict to deterministic source instances); namely, a by-table solution is a p-solution but not necessarily vice-versa. In particular, the characterization of Theorem 3.6 for a p-solution does not hold for a by-table solution. As an example, for a by-table solution $\tilde{\mathcal{J}}$, the set $\Omega_+(\tilde{\mathcal{J}})$ must be at most as large as $\Omega_+(\tilde{\mathcal{M}})$ whereas no restriction of this nature exists for our p-solution.

A different type of solution studied (explicitly or implicitly) in Dong et al. [2007, 2009], and Sarma et al. [2009] is the *by-tuple solution*.[15] A by-tuple solution differs from a by-table solution in that a by-tuple solution allows different tuples to be transformed by different possible worlds of the p-mapping; that is, for each tuple there is a probabilistic choice of the mapping to apply thereon. This notion is restrictive, since it is makes sense only when the mappings are transformations of individual facts. In particular,

[15]In Sarma et al. [2008a], only the by-table type is considered.

the mappings of Dong et al. [2007, 2009] and Sarma et al. [2009] for the by-tuple semantics are essentially inclusion dependencies [Casanova et al. 1984].

7. CONCLUSIONS

In this article, we developed a broad and flexible framework for data exchange over probabilistic data. For that, we had to consider the fundamental notions of traditional data exchange, such as solution, universal solution, and target query evaluation, and generalize them appropriately. We did so by introducing the notion of a probabilistic match. In particular, to accommodate source and target p-instances we defined the notion of a p-solution in terms of a probabilistic match (namely, the $\mathsf{SOL}_{\mathcal{M}}$-match). We explored the coherence of our basic definitions by scrutinizing them and providing several different characterizations for each of them. We explored the application of the framework to a concrete setting where p-instances are compactly encoded by annotations. Finally, we generalized the framework to allow for probabilistic schema mappings, by introducing the p-problem as a construct that represents a joint probability distribution over the data and mappings.

The notion of a probabilistic match allows us to systematically extend other concepts of data exchange into the probabilistic setting. An example is the core solution [Fagin et al. 2005b; Gottlob and Nash 2008]; in fact, it turns out that this extension of the core has various desired properties, which we are currently exploring.

APPENDIX A: PROOFS FOR THE COMPLEXITY RESULTS OF TABLE I

In Section 5.3, we analyzed the complexity of basic operations in data exchange, where the source p-instance is compactly encoded in the form of I_p^α. The results are summarized in Table I, and here we give the proofs that do not appear in the body of the article.

A.1. Existence of a (Universal) p-Solution

We now consider the existence-of-solution problem under different types of dependencies (that correspond to the header row of Table I). If the schema mapping contains only st-tgds and t-tgds, then a solution always exists for an ordinary (deterministic) source instance [Fagin et al. 2005a]; therefore, by Proposition 4.3, a p-solution always exists. So next, we consider schema mappings that include t-egds.

We first show if that if the schema mapping contains only full st-tgds and t-egds (rather than general st-tgds and t-egds as in Theorem 5.6), then one can efficiently test whether a p-solution exists. The proof is based on the following two lemmas, where the second one is proved by using the first one.

LEMMA A.1. *For all schema mappings* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *where* Σ *contains only full st-tgds and t-egds, there is a natural number* K_Σ *with the following property. For all source instances* I*, if there is a solution for all* $I' \subseteq I$ *such that* $|I'| \leq K_\Sigma$*, then there is a solution for* I*.*

PROOF. We choose as K_Σ the number L^2, where L is the maximal number of conjuncts on the premise of a dependency of Σ. Now, let I be a source instance and suppose that no solution for I exists. We will construct a set I' of K_Σ or fewer facts of I, such that some *chase* [Beeri and Vardi 1984; Maier et al. 1979] of I' with Σ fails (hence, no solution exists for I' [Fagin et al. 2005a]). Consider a failing chase c for I with Σ (it is shown in Fagin et al. [2005a] that one necessarily exists). Since the tgds of Σ are full, c constructs only ground facts. Failure of c means that there exists some t-egd d that is violated by some (ground) facts $g_1, \ldots, g_l$ constructed by c, where l is the number of conjuncts on the premise of d. Now, since Σ has no t-tgds, each g_i is constructed by an

st-tgd. More particularly, for each g_i ($1 \leq i \leq l$), there exists an st-tgd d_i and j_i facts $f_1^i, \ldots, f_{j_i}^i$ of I (where j_i is the number of conjuncts on the premise of d_i), such that g_i is obtained by applying d_i to $f_1^i, \ldots, f_{j_i}^i$. So, we define $I' = \cup_{i=1}^{l} \{f_1^i, \ldots, f_{j_i}^i\}$. Observe that I' has at most K_Σ facts and, from its construction, a failing chase of I' with Σ exists, as claimed. □

In the next lemma, we use the number K_Σ of the previous lemma (Lemma A.1). This next lemma uses the notion of a *feasible* instance, which is defined in Section 5.3.1.

LEMMA A.2. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping, such that* Σ *contains only full st-tgds and t-egds. Let* I_p^α *be a DNF source instance. A p-solution exists if and only if for all* $I' \subseteq I$ *where* $|I'| \leq K_\Sigma$*, if* I' *is feasible then a solution for* I' *exists.*

PROOF. We start with the "only if" direction. Suppose that I' is a feasible set of K_Σ or fewer facts of I, such that no solution for I' exists. We will show that there exists a feasible truth assignment τ for $\mathsf{EVar}(\alpha)$, such that no solution for I_τ^α exists; it then follows from Proposition 5.5 that no p-solution exists. Since I' is feasible, it follows by definition that there exists a feasible truth assignment τ such that $I' \subseteq I_\tau^\alpha$. And since $I' \subseteq I_\tau^\alpha$, it is easy to see that every solution for I_τ^α is also a solution for I'. Thus, our assumption implies that no solution for I_τ^α exists, as claimed.

We now prove the "if" direction. Suppose that no p-solution exists. Then, by Proposition 5.5, there exists a feasible truth assignment τ for $\mathsf{EVar}(\alpha)$, such that no solution for I_τ^α exists. From Lemma A.1 we conclude that there exists an instance $I' \subseteq I_\tau^\alpha$ of K_Σ or fewer facts, such that no solution for I' exists. From the choice of I', it follows that I' is feasible. □

We can now prove the tractability result for the case of full st-tgds and t-tgds.

THEOREM A.3. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping, such that* Σ *contains only full st-tgds and t-egds. Testing whether a p-solution exists, given a source DNF instance, is in polynomial time.*

PROOF. Let I_p^α be a source DNF instance. Lemma A.2 implies the correctness of following algorithm for testing whether a p-solution exists.

1: **for all** $I' \subseteq I$, such that $|I'| \leq K_\Sigma$ **do**
2: **if** I' is feasible **then**
3: **if** there is no solution for I' **then**
4: return **false**
5: return **true**

Note that K_Σ is fixed, so the iteration of Line 1 is efficient. It follows from Fagin et al. [2005a] that the condition of Line 3 can be efficiently tested. It remains to show how to efficiently apply the test of Line 2. Here, we use the fact that the condition α is in disjunctive normal form. Let $I' = \{f_1, \ldots, f_m\}$. We first eliminate all the variables of $\mathsf{EVar}(\alpha)$ that are assigned by π the probability 0 or 1 (e.g., if the variable e is deterministically **true** then we remove the atomic formulas e from each conjunction, and we remove all the conjunctions that contain the atomic formula $\neg e$). So, we assume that no variable has a deterministic value. Then, testing feasibility of I' reduces to testing satisfiability of the formula $\wedge_{i=1}^{m} \alpha(f_i)$. This formula is a conjunction of K DNF formulas, where K is upper-bounded by a fixed number (recall that $m \leq K_\Sigma$). Satisfiability of such a formula can be tested in polynomial time by, for example, transforming the formula into DNF using distributivity of conjunction over disjunction. The number of operations is at most S^K, where S is the size of the formula. □

The next theorem shows that one cannot generalize Theorem A.3 by adding t-tgds to the set Σ (that contains full st-tgds and e-tgds), even if these t-tgds are restricted to be full.

THEOREM A.4. *There exists a schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma)$*, such that* Σ *contains only full st-tgds, full t-tgds, and t-egds, and testing whether a p-solution exists, given a source DNF instance, is* coNP*-complete.*

PROOF. Recall that the proof of Theorem 5.6 constructed a schema mapping that contains exactly one non-full st-tgd. The proof of this theorem is similar to the proof of Theorem 5.6, except that we replace the non-full st-tgd with a full t-tgd. This is done as follows:

The schemas $\mathbf{S}$ and $\mathbf{T}$ are defined exactly as in the proof of Theorem 5.6. Similarly, for a satisfying assignment φ, the DNF instance I_p^α is defined exactly as in the proof of Theorem 5.6. The set Σ' is obtained from the set Σ in the proof of Theorem 5.6 by replacing the non-full st-tgd d_{EL}^{st} with the following full t-tgd:

$$d_{EL}^{\text{ttgd}} : \forall x, y, u(E_{\mathbf{T}}(x, y) \wedge L_{\mathbf{T}}(x, u) \rightarrow L_{\mathbf{T}}(y, u)).$$

Now, consider a truth assignment τ for φ, and let G_τ and G' be the graphs defined in the proof of Theorem 5.6. The dependency d_{EL}^{ttgd} means that if a node v_1 of G' is labeled and it is connected by an edge to the node v_2, then v_2 should also be labeled and, moreover, its label should be that of v_1. Recall that, by the dependency d_L^{st}, node 0 and m must have the labels a and b, respectively. Consequently, due to d_{EL}^{st}, if G_τ contains a path from 0 to m, then we get a contradiction to d_e^{egd}. On the other hand, if G_τ does not have a path from 0 to m, then it is easy to come up with a solution (e.g., the construction mentioned in the proof of Theorem 5.6). It follows that there is a p-solution if and only if φ is not satisfiable, as required. □

This concludes the proofs of our complexity results that are stated in the top row of each of the two parts (i.e., the top part and the bottom part) of Table I, namely, the complexity of testing for the existence of a p-solution and for that of a universal p-solution.

A.2. Materializing a (Universal) p-Solution

We now consider the problem of materializing a candidate p-solution (i.e., a target p-instance $\tilde{\mathcal{J}}$ that forms a p-solution if one exists) and a candidate universal p-solution, where each is represented as a DNF instance J_q^β (i.e., $\tilde{\mathcal{J}} = \text{p-space}(J_q^\beta)$).

We start with the case where the schema mapping contains only st-tgds. In this case, a universal p-solution can be constructed by combining the chase algorithm [Beeri and Vardi 1984; Fagin et al. 2005a; Maier et al. 1979] with the known concept of maintaining conditions (or provenance) in relational operators, which is used in Green and Tannen [2006], Green et al. [2007b], and Imielinski and Lipski [1984] for showing closure of annotated databases under relational algebra. A similar construction is used in Green et al. [2007a] for the task of propagating trust conditions through data exchange between peers in a network. For completeness of presentation, we describe this construction.

Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ contains only st-tgds, and let I_p^α be a source DNF instance. We incrementally construct a target DNF instance J_q^β with the following properties. First, J_q^β uses only event variables from I_p^α, that is, $\mathsf{EVar}(\beta) \subseteq \mathsf{EVar}(\alpha)$. Second, q is the restriction of p to $\mathsf{EVar}(\beta)$. This is done as follows. We start with $J = \emptyset$. We then do the following for each st-tgd d of Σ and homomorphism h mapping the premises of d into I. Let I' be the tuples of I that

are the image of the premises of d under the homomorphism h, and let J' be the image of the conclusions of d under a homomorphism that extends h by mapping each existentially quantified variable into a fresh labeled null (as done in the tgd "chase step" in Fagin et al. [2005a]). Next, we construct a DNF formula ξ that is equivalent to the conjunction $\wedge_{f \in I'} \alpha(f)$. Note that ξ can be obtained efficiently, since Σ is fixed and, therefore, the conjunction involves a bounded number of conjuncts, each of which in disjunctive normal form. Then, the following is done for each fact $g \in J'$.

—If g is not in J, then we add g to J and set $\beta(g) = \xi$.
—If g is already in J, then replace $\beta(g)$ with $\beta(g) \vee \xi$.

The following proposition shows that this construction is, indeed, a universal p-solution.

PROPOSITION A.5. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping where* Σ *contains only st-tgds. Given a source DNF instance* I_p^α*, the DNF instance* J_q^β *produced by the construction is a universal p-solution.*

PROOF. Consider a truth assignment τ for $\mathsf{EVar}(\alpha)$. We will show that J_τ^β consists precisely of the facts that are produced when restricting the algorithm to I_τ^α. Observe that this algorithm slightly differs from the chase of Fagin et al. [2005a], since their chase does not apply a step if the tgd is already satisfied (for a given set of source facts that match the premise), while our algorithm may do so. Nevertheless, the proof of Fagin et al. [2005a] that the result is a universal solution ignores this issue (which is only needed for showing termination—a property that is irrelevant in our t-tgd-free case). Consequently, it follows from that proof of Fagin et al. [2005a] that J_τ^β is a universal solution for I_τ^α. Thus, we conclude that by randomly choosing a truth assignment τ for $\mathsf{EVar}(\alpha)$ and then producing I_τ^α and J_τ^β, we obtain a p-space over $\mathsf{Inst}(\mathbf{S}) \times \mathsf{Inst}(\mathbf{T})$ that is a $\mathsf{USOL}_\mathcal{M}$-match of p-space$(I_p^\alpha)$ in p-space(J_q^β), hence, p-space(J_q^β) is a universal p-solution.

First, we will show that J_τ^β contains all the facts that are produced when restricting the algorithm to I_τ^α. Let $d \in \Sigma$ be a dependency and let I' be a set of facts of I_τ^α that matches the premise of d. When I' and d are considered in the algorithm, we add to J the set J' of tuples (that matches the conclusion of d), and ξ becomes a disjunct in the condition of each fact of J'. Since τ satisfies each of the conditions of the facts of I', it follows from the definition of ξ that τ satisfies ξ and, hence, it also satisfies the condition of each of the facts of J'. Therefore, $J' \subseteq J_\tau^\beta$, as claimed.

Next, we will show that every fact g of J_τ^β is produced when restricting the algorithm to I_τ^α. Let g be a fact of J_τ^β. Then, since $\beta(g)$ is satisfied by τ, some ξ that was added as a disjunct of $\beta(g)$ is satisfied by τ. Consider the set I' of facts (that matches the premise of d) that was used to create ξ when it was added to $\beta(g)$ as a disjunct. Then τ satisfies each fact of I'; hence, $I' \subseteq I_\tau^\alpha$. Therefore, g is produced when restricting the algorithm to I_τ^α, as claimed. □

From Proposition A.5, we conclude the following theorem.

THEOREM A.6. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *contains only st-tgds. Given a source DNF instance* I_p^α*, one can materialize a universal p-solution as a DNF instance* J_q^β*, in polynomial time.*

Next, we show that Theorem A.6 generalizes to the case where the schema mapping includes t-egds in addition to st-tgds, but where the st-tgds are full. For that, we need the following lemma.

LEMMA A.7. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *contains only full st-tgds and t-egds, and let* I *be a source instance such that a solution for* I *with respect to* Σ *exists. Let* Σ' *be obtained from* Σ *by removing all the t-egds. Every ground universal solution for* I *with respect to* Σ' *is a universal solution for* I *with respect to* Σ.

PROOF. Let J be a solution for I with respect to Σ and let J' be a ground universal solution for I with respect to Σ'. Note that J exists, by assumption. Since J is a solution with respect to Σ, it is also a solution with respect to Σ'; hence, we have $J' \rightarrow J$. This means that J contains all the tuples of J', since the latter is ground. Therefore, J' violates none of the t-egds of Σ, or otherwise J violates such a t-egd as well (and, hence, J is not a solution with respect to Σ). Thus, J' is a solution with respect to Σ. Moreover, since J' is universal with respect to Σ', it is also universal with respect to Σ (since $\Sigma' \subseteq \Sigma$). □

We can now prove the following theorem.

THEOREM A.8. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *contains only full st-tgds and t-egds. Given a source DNF instance* I_p^α*, one can materialize a candidate universal p-solution as a DNF instance* J_q^β*, in polynomial time.*

PROOF. Let I_p^α be a given source DNF instance. We will show how to efficiently construct a candidate universal p-solution as a DNF instance J_q^β. We first test whether a universal p-solution exists. We can do so efficiently, due to Theorem A.3 and Proposition 5.3. If no universal p-solution exists, then we generate an arbitrary target DNF-instance J_q^β. So, suppose that a universal p-solution exists. Let Σ' be obtained from Σ by removing all the t-egds. Then, we execute the previously described chase procedure to I_p^α and the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma')$, and obtain a DNF instance J_q^β, which by Proposition A.5 is a universal p-solution with respect to Σ'.

To complete the proof, we will show that J_q^β is also a universal p-solution for I_p^α with respect to Σ (and not just with respect to Σ'). Denote by $\mathcal{M}$ the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ and by $\mathcal{M}'$ the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma')$. Let $\tilde{\mathcal{P}}$ be a $\mathsf{USOL}_{\mathcal{M}'}$-match of p-space$(I_p^\alpha)$ in p-space(J_q^β), and let $(\hat{I}, \hat{J})$ be a pair in the support of $\tilde{\mathcal{P}}$. Note that the construction of J_q^β is such that J is a ground instance, since Σ' contains only full st-tgds. Therefore, every random instance in the support of p-space(J_q^β) is a ground instance. In particular, $\hat{J}$ is ground (recall that p-space(J_q^β) is a marginal distribution of $\tilde{\mathcal{P}}$), hence, it is a ground universal solution for $\hat{I}$ with respect to Σ'. Note that a solution for $\hat{I}$ with respect to Σ exists, due to Proposition 4.3 and the assumption that a universal p-solution for I_p^α with respect to Σ exists. Therefore, by Lemma A.7, it holds that $\hat{J}$ is a universal solution for $\hat{I}$ with respect to Σ. We conclude that $\tilde{\mathcal{P}}$ is a $\mathsf{USOL}_{\mathcal{M}}$-match of p-space$(I_p^\alpha)$ in p-space(J_q^β), which means that J_q^β is a universal p-solution with respect to Σ, as claimed. □

The last tractability results that we study in this section are about materializing a p-solution that is not necessarily universal. Let $(\mathbf{S}, \mathbf{T}, \Sigma)$ be a standard schema mapping, and let I_p^α be a source p-instance. If Σ contains only st-tgds and t-tgds, then we construct a solution J for I with respect to Σ. This deterministic J can be viewed as a special case of a p-instance, and as such, J is actually a p-solution for I_p^α. This is correct, because of the following two facts.

(1) A solution for I necessarily exists (due to the content of Σ).
(2) A solution for I is also a solution for every subset I' of I.

Recall that constructing a solution for I is tractable [Fagin et al. 2005a]. Now, consider the case of a general standard schema mapping (where Σ may contain t-egds in addition to st-tgds and t-tgds). In this case, Item 2 is still correct, but Item 1 is not; that is, a solution for I does not necessarily exist, even if a p-solution for I_p^α exists. Actually, we will later show an example where materializing a (not necessarily universal) p-solution as a DNF instance is intractable. But, if I_p^α is a tuple-independent instance, then by the proof of Proposition 5.4, a solution for I is a (deterministic) p-solution for I_p^α. Hence, we get the following:

PROPOSITION A.9. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a standard schema mapping. The following holds:*

(1) *Given a source tuple-independent instance, one can materialize a candidate p-solution as a deterministic instance, in polynomial time.*
(2) *If* Σ *contains only st-tgds and t-tgds, then given a source DNF instance, one can materialize a candidate p-solution as a deterministic instance, in polynomial time.*

We now prove hardness results. The following theorem shows that there are schema mappings $\mathcal{M}_1$ and $\mathcal{M}_2$, such that $\mathcal{M}_1$ contains only st-tgds and e-tgds, $\mathcal{M}_2$ contains only full st-tgds, full t-tgds, and t-egds, and in both schema mappings materializing a p-solution (which is not necessarily universal) as a DNF instance, given a source DNF instance, is intractable.

THEOREM A.10. *Assume* P $\neq$ NP. *There are two schema mappings* $(\mathbf{S}_1, \mathbf{T}_1, \Sigma_1)$ *and* $(\mathbf{S}_2, \mathbf{T}_2, \Sigma_2)$*, such that no polynomial-time algorithm constructs a candidate p-solution as a DNF instance* J_q^β*, given a source DNF instance* I_p^α*. Moreover,*

(1) Σ_1 *contains only st-tgds and t-egds, and*
(2) Σ_2 *contains only full st-tgds, full t-tgds, and t-egds.*

PROOF. We start with Part 1. The proof is an adaptation of the proof of Theorem 5.6. Recall that the reduction (from the complement of 3-SAT) constructed a schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$ (where Σ contains only st-tgds and t-egds) and a source DNF instance I_p^α such that a p-solution exists if and only if the given 3-CNF formula φ is not satisfiable. The idea there is as follows: Consider a truth assignment τ for φ. If τ satisfies φ, then the existence of the fact $L_{\mathbf{T}}(m, \mathtt{a})$ in a solution for I_τ^α is dictated; otherwise, it is not dictated. In addition, the source fact $L_{\mathbf{S}}(m, \mathtt{b})$ and the st-tgd d_L^{st} dictate the existence of the fact $L_{\mathbf{T}}(m, \mathtt{b})$ in a solution I_τ^α (regardless of whether or not τ satisfies φ). Since the t-egd d_l^{egd} requires uniqueness on the left side of $L_{\mathbf{T}}$, it follows that a p-solution exists if and only if φ is not satisfiable.

To adapt the proof of Theorem 5.6 to Part 1, let $(\mathbf{S}_1, \mathbf{T}_1, \Sigma_1)$ be the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$. Given a 3-CNF formula φ, we construct the source DNF instance I_p^α exactly as in the proof of Theorem 5.6, except for the following change. The condition $\alpha(L_{\mathbf{S}}(m, \mathtt{b}))$ is $\neg\varphi$ instead of **true**. Observe that $\neg\varphi$ can be efficiently transformed into DNF (hence, the reduction is efficient), since φ is in CNF. The effect of this change is that a p-solution necessarily exists due to Proposition 5.5 and the fact that a truth assignment τ cannot dictate the simultaneous existence of $L_{\mathbf{T}}(m, \mathtt{a})$ and $L_{\mathbf{T}}(m, \mathtt{b})$ in a solution for I_τ^α. This is because, as noted above, for τ to dictate the existence of $L_{\mathbf{T}}(m, \mathtt{a})$, necessarily τ must satisfy φ, but then τ cannot dictate the existence of $L_{\mathbf{T}}(m, \mathtt{b})$, because the condition $\alpha(L_{\mathbf{S}}(m, \mathtt{b}))$ is $\neg\varphi$. Thus, a candidate p-solution is, in fact, a p-solution. Now, let $\tilde{\mathcal{J}}$ be a p-solution, and let $\tilde{\mathcal{P}}$ be a $\mathsf{SOL}_{\mathcal{M}}$-match of p-space$(I_p^\alpha)$ in $\tilde{\mathcal{J}}$. Consider a pair (I_τ^α, J') in the support of $\tilde{\mathcal{P}}$. If τ satisfies φ, then J' must contain $L_{\mathbf{T}}(m, \mathtt{a})$. If τ violates φ, then J' must contain $L_{\mathbf{T}}(m, \mathtt{b})$ (since I_τ^α contains $L_{\mathbf{S}}(m, \mathtt{b})$) and, hence, J' cannot contain $L_{\mathbf{T}}(m, \mathtt{a})$. Therefore, the support of $\tilde{\mathcal{P}}$ contains a pair (I', J') such that $L_{\mathbf{T}}(m, \mathtt{a}) \in J'$ if

and only if φ is satisfiable. Since $\tilde{\mathcal{J}}$ is the right marginal of $\tilde{\mathcal{P}}$, the fact $L_{\mathbf{T}}(m, \mathtt{b})$ occurs in $\tilde{\mathcal{J}}$ with a nonzero probability if and only if φ is satisfiable.

So, to test whether φ is satisfiable, we construct a (candidate) p-solution for I_p^α as a DNF instance J_q^β, and test whether $L_{\mathbf{T}}(m, \mathtt{a})$ occurs in p-space(J_q^β) with a nonzero probability. This test is tractable, since it is true if and only if J contains $L_{\mathbf{T}}(m, \mathtt{a})$ and, moreover, $\beta(L_{\mathbf{T}}(m, \mathtt{a}))$, which is in DNF, is satisfiable by a feasible assignment.

For Part 2, we use the exact same adaption as the one for proving Part 1, except that the new adaptation is of the proof of Theorem A.4 (rather than Theorem 5.6). Since the two adaptations are essentially identical, we do not give the one for Part 2. □

Recall from Part 2 of Theorem A.9 that when the schema mapping is standard and it contains only st-tgds and t-tgds, then one can efficiently materialize a p-solution (which is not necessarily universal) as a deterministic target instance. Also, recall from Theorem A.6 that, if the schema mapping contains only st-tgds, then one can efficiently materialize a universal p-solution. The following theorem shows that materializing a universal p-solution is intractable for some schema mapping that contains only full st-tgds and full t-tgds. Note that, in this case, a universal p-solution is the same as a candidate universal p-solution, since a universal p-solution always exists (Proposition 4.3).

THEOREM A.11. *Assume* P $\neq$ NP. *There is a schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma)$*, such that* Σ *has only full st-tgds and full t-tgds, and no polynomial-time algorithm constructs a universal p-solution as a DNF instance* J_q^β*, given a source DNF instance* I_p^α*.*

PROOF. For the proof, we use essentially a simplified version of the reduction we used for proving Theorem A.4. In particular, we show a reduction from 3-SAT.

The source schema $\mathbf{S}$ and the target schema $\mathbf{T}$ are exactly those used in the proofs of Theorems 5.6 and A.4. That is, $\mathbf{S}$ has two binary relation symbols $E_{\mathbf{S}}$ and $L_{\mathbf{S}}$, and $\mathbf{T}$ contains two binary relation symbols $E_{\mathbf{T}}$ and $L_{\mathbf{T}}$. The set Σ contains the following dependencies.

—d_E^{st}: $\forall x, y(E_{\mathbf{S}}(x, y) \rightarrow E_{\mathbf{T}}(x, y))$
—d_L^{st}: $\forall x, u(L_{\mathbf{S}}(x, u) \rightarrow L_{\mathbf{T}}(x, u))$
—d_{EL}^{ttgd} : $\forall x, y, u(E_{\mathbf{T}}(x, y) \wedge L_{\mathbf{T}}(x, u) \rightarrow L_{\mathbf{T}}(y, u))$

Let $\varphi = c_1 \wedge \cdots \wedge c_m$ be an instance of 3-SAT, and assume, without loss of generality, that the variables of φ belong to EVar. For all integers i where $1 \leq i \leq m$, the source instance I contains the fact $f_i = E_{\mathbf{S}}(i-1, i)$, and the condition $\alpha(f_i)$ is the conjunct c_i. In addition, I contains the fact $L_{\mathbf{S}}(0, \mathtt{a})$ with the condition **true**. The function p assigns to each variable of φ the probability 0.5. As in the proof of Theorem A.10, we will show how to efficiently decide the satisfiability of φ from a p-solution in the form of a DNF instance.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, let $\tilde{\mathcal{J}}$ be a p-solution for p-space(I_p^α), and let $\tilde{\mathcal{P}}$ be a $\mathsf{USOL}_{\mathcal{M}}$-match of p-space($I_p^\alpha$) in $\tilde{\mathcal{J}}$. Consider a pair (I_τ^α, J') in the support of $\tilde{\mathcal{P}}$. Similarly to the proof of Theorem A.4 it is shown that τ satisfies φ if and only if the existence of the fact $L_{\mathbf{T}}(m, \mathtt{a})$ is dictated in a solution for I_τ^α. Now, since J' is a universal solution for I_τ^α, it contains the fact $L_{\mathbf{T}}(m, \mathtt{a})$ if and only if this containment is dictated. Thus, the support of $\tilde{\mathcal{P}}$ contains a pair (I', J') such that $L_{\mathbf{T}}(m, \mathtt{a}) \in J'$ if and only if φ is satisfiable. From here, we continue exactly as in the proof of Theorem A.4; that is, to test whether φ is satisfiable, we construct a universal p-solution for I_p^α as a DNF instance J_q^β, and test whether $L_{\mathbf{T}}(m, \mathtt{a})$ occurs in p-space(J_q^β) with a nonzero probability. □

Note that the proofs given thus far in this section cover all the entries of Table I regarding the problems of materializing a candidate p-solution and materializing a candidate universal p-solution, for the case of source DNF instances (the top part of the table). Moreover, the proofs given thus far cover all the positive (tractability) results for these problems in the case of tuple-independent instances.

It is left to show the existence of a schema mapping that contains only st-tgds and t-egds, such that it is intractable to materialize a candidate universal p-solution as a DNF instance, given a source tuple-independent instance. (Recall from Theorem A.8 that if the schema mapping contains only full st-tgds and t-egds, then the problem is tractable.)

THEOREM A.12. *Assume* RP $\neq$ NP. *There is a schema mapping* $(\mathbf{S}, \mathbf{T}, \Sigma)$*, such that* Σ *has only st-tgds and t-egds, and no polynomial-time algorithm constructs a universal p-solution as a DNF instance* J_q^β*, given a tuple-independent instance* I_p^α.

PROOF. The proof is essentially an adaptation of the proof of Theorem 5.7, where we use reduction from #monotone-2-SAT. Let $\varphi = c_1 \wedge \cdots \wedge c_m$ be a monotone 2-CNF formula over the event variables $e_1, \ldots, e_n$. The source schema $\mathbf{S}$ is similar to one in the proof of Theorem 5.7, except that we add a ternary relation symbol $E_\mathbf{S}$. The source tuple-independent instance I_p^α is, again, similar to that of the proof of Theorem 5.7, except that we add to it the fact $E_\mathbf{S}(0, 0, \mathtt{a})$. The target schema $\mathbf{T}$ consists of a ternary relation symbol $E_\mathbf{T}$. The set Σ contains the following dependencies.

—d_E^{st}: $\forall x, y, l(E_\mathbf{S}(x, y, l) \rightarrow E_\mathbf{T}(x, y, l))$
—d_l^{st}: $\forall v, u, x, y(V_\mathbf{S}(v) \wedge C_\mathbf{S}(v, u, x, y) \rightarrow \exists l E_\mathbf{T}(x, y, l))$
—d_r^{st}: $\forall v, u, x, y(V_\mathbf{S}(u) \wedge C_\mathbf{S}(v, u, x, y) \rightarrow \exists l E_\mathbf{T}(x, y, l))$
—d_E^{egd}: $\forall x, y, z, l_1, l_2(E_\mathbf{T}(x, y, l_1) \wedge E_\mathbf{T}(y, z, l_2) \rightarrow l_1 = l_2)$.

Let $\mathcal{M}$ be the schema mapping $(\mathbf{S}, \mathbf{T}, \Sigma)$. A p-solution for p-space(I_p^α) exists (hence, a universal one also exists by Proposition 5.3), because such a p-solution can be the deterministic instance that consists of all the facts $E_\mathbf{T}(i-1, i, \mathtt{a})$, where $1 \leq i \leq m$.

Let $\tilde{\mathcal{J}}$ be a p-solution for p-space(I_p^α), and let $\tilde{\mathcal{P}}$ be a $\mathsf{USOL}_\mathcal{M}$-match of p-space$(I_p^\alpha)$ in $\tilde{\mathcal{J}}$. Let (I_τ^α, J') be in the support of $\tilde{\mathcal{P}}$. Since J' is a solution for I_τ^α, the dependencies d_E^{st}, d_l^{st} and d_r^{st} say that J' must include the fact $E_\mathbf{T}(0, 0, \mathtt{a})$, and a fact of the form $E_\mathbf{T}(j-1, j, l)$ (for some l) if I_τ^α has a fact of the form $C_\mathbf{S}(i_1, i_2, j-1, j)$ and at least one of the facts $V_\mathbf{S}(i_1)$ and $V_\mathbf{S}(i_2)$. If we view the relation $E_\mathbf{T}$ as a set of edges $(j-1, j)$ such that each edge has a label l (and multiple edges are allowed to connect two nodes $j-1$ and j), then a self-loop $(0, 0)$ has the label $\mathtt{a}$, and the dependency d_E^{egd} says that two edges that lie on a single path must have the same label. If τ satisfies φ, then J' must label the edge $(m-1, m)$ with $\mathtt{a}$, that is, it must contain the fact $E_\mathbf{T}(m-1, m, \mathtt{a})$. On the other hand, if τ violates φ, then J' does not include $E_\mathbf{T}(m-1, m, \mathtt{a})$, since J' is universal and there is a solution for I_τ^α that does not include $E_\mathbf{T}(m-1, m, \mathtt{a})$. We conclude that a random pair (I_τ^α, J') in the support of $\tilde{\mathcal{P}}$ is such that $E_\mathbf{T}(m-1, m, \mathtt{a}) \in J'$ if and only if τ satisfies φ. From this point, we continue exactly as in the proof of Theorem 5.7, and we omit the repetition of that part of the proof. □

This concludes the proof of the complexity results regarding the problem of materializing a p-solution and a universal p-solution, namely, the second and third rows of the two parts (the top and bottom parts) of Table I.

A.3. Evaluating Target UCQs

In this section, we study the complexity of evaluating target UCQs, and provide proofs for the fourth and fifth rows in each of the two parts of Table I.

A.3.1. Exact Query Answering. We start with exact answering of UCQs over source DNF instances. The following proposition shows that, in general, this problem is in $\mathrm{FP}^{\#\mathrm{P}}$.

PROPOSITION A.13. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a standard schema mapping and let* Q *be a UCQ over* $\mathbf{T}$. *Computing* $conf_Q(\mathbf{a})$*, given a source DNF instance* I_p^α*, is in* $\mathrm{FP}^{\#\mathrm{P}}$.

PROOF. We will show a polynomial-time algorithm that uses an oracle to a #P-complete problem and, given a source DNF instance I_p^α and a tuple $\mathbf{a}$, computes $conf_Q(\mathbf{a})$. First, we use the oracle to test whether a p-solution exists. By Proposition 5.5, this can be done efficiently. If no p-solution exists, then we return 1 and terminate. Next, we assume that a p-solution exists.

Let $\mathcal{A}$ be the set of all the feasible truth assignments τ for $\mathsf{EVar}(\alpha)$, such that $\mathbf{a} \in \mathrm{certain}(Q, I_\tau^\alpha, \Sigma)$. By Proposition 4.12, the confidence of $\mathbf{a}$ is the probability that a random instance I' of p-space(I_p^α) is such that $\mathbf{a} \in \mathrm{certain}(Q, I', \Sigma)$. By the definition of p-space(I_p^α), this is the probability that a random truth assignment chosen according to p belongs to $\mathcal{A}$.

In Fagin et al. [2005a], it is shown that determining whether a tuple $\mathbf{b}$ is a certain answer can be computed in polynomial time. It follows that we can efficiently test whether a given assignment τ for $\mathsf{EVar}(\alpha)$ is in $\mathcal{A}$. So, our goal is to compute the probability that given independent random variables satisfy some polynomial-time testable property. By using the fact that probabilities are represented as rational numbers, we can use the techniques of Grädel et al. [1998] in order to compute this probability using the #P-complete oracle (essentially, this technique constructs an NP machine with redundant branches that accommodate the probabilities). An alternative proof is sketched as follows.

Let $\mathsf{EVar}(\alpha) = \{e_1, \ldots, e_n\}$, and suppose that each $\pi(e_i)$ is represented as n_i/m_i. So, it is rather easy to see that the probability that a random truth assignment belongs to $\mathcal{A}$ is the product of $(\prod_{i=1}^n m_i)^{-1}$ and the number of all functions $F : \{e_1, \ldots, e_n\} \to \{\mathbf{true}, \mathbf{false}\} \times \mathbb{N}$ with the following properties.

—For all $1 \le i \le n$, if $F(e_i) = (\mathbf{true}, k_i)$, then $1 \le k_i \le n_i$, and if $F(e_i) = (\mathbf{false}, k_i)$, then $n_i + 1 \le k_i \le m_i$.
—The projection of F on the first coordinate is a truth assignment of $\mathcal{A}$.

Since there is a polynomial-time algorithm that accepts all pairs $((\mathbf{a}, I_p^\alpha), F)$ such that F satisfies the above properties, counting the number of such functions F is in $\#P$. □

A.3.2. Approximate Query Answering. We now prove the results on approximate answering of target UCQs over source DNF instances. (The definition of approximate query answering in data exchange is in Section 5.3.4.) We first need some notation.

Let J_q^β be a DNF instance and let k be a natural number. We say that J_q^β is a *DNF*$_{\le k}$ *instance* if each of the disjuncts in the image of β have at most k conjuncts; that is, for all facts f of J, the condition $\beta(f)$ is in the form $d_1 \vee \cdots \vee d_t$, where each d_i is a conjunction of k or fewer atomic formulas. (Note that a tuple-independent instance is a special case of an $\mathrm{DNF}_{\le 1}$ instance.)

We first discuss approximate evaluation of UCQs over DNF instances (and in particular tuple-independent instances), regardless of any schema mapping. In Dalvi and Suciu [2007b], it is shown that, over tuple-independent instances, every CQ (over a fixed schema $\mathbf{T}$) has an FPRAS. To prove that, they applied the Monte-Carlo technique

of Karp et al. [1989]. This technique can be applied to generalize the result of Dalvi and Suciu [2007b] to UCQs over DNF instances. The formal result is as follows.

PROPOSITION A.14. *Let* $\mathbf{T}$ *be a schema, let* Q *be a UCQ over* $\mathbf{T}$ *and let* k *be a fixed integer.*

—Over DNF instances, there is an FPRAS for Q.
—Over $\text{DNF}_{\leq k}$ *instances, there is an FPAS for* Q.

PROOF. Let J_q^β be a DNF instance and let $\mathbf{a}$ be a tuple. Let K_Q be the maximal number of conjuncts in any of the disjuncts of Q. The following is an easy observation. For all instances J over $\mathbf{S}$ it holds that $\mathbf{a} \in Q(J)$ if and only if there exists an instance $J' \subseteq J$, such that J' has at most K_Q facts and $\mathbf{a} \in Q(J')$. Consequently, a truth assignment τ for $\mathsf{EVar}(\beta)$ satisfies $\mathbf{a} \in Q(J_\tau^\beta)$ if and only if τ satisfies the following condition. There is a set $J' \subseteq J$ of K_Q or fewer facts, such that $\mathbf{a} \in Q(J')$ and $J' \subseteq J_\tau^\beta$ (i.e., all the conditions in the image of β over J' are satisfied by τ).

We will show that the above condition about truth assignments τ can be phrased as a DNF formula φ over $\mathsf{EVar}(\beta)$; moreover, φ can be constructed efficiently. In addition, if J_q^β is a $\text{DNF}_{\leq k}$ instance, then each disjunct of φ is a conjunction of at most $k \cdot K_Q$ atomic formulas. Consequently, the goal will be to compute the probability that a DNF formula (over probabilistically independent random variables) is satisfied. As mentioned in the proof of Theorem 5.7, an FPRAS for this task can be obtained by a rather straightforward adaptation of the Monte-Carlo technique of Karp et al. [1989] for approximating the number of satisfying assignments for a DNF formula. Moreover, it is shown in Luby and Velickovic [1996] that if there is a fixed upper bound on the size of each disjunct, then an FPAS exists.[16]

It remains to show how φ is constructed. Let $E(Q, J)$ be the set of all the subsets $J' \subseteq J$, such that J' has at most K_Q facts and $\mathbf{a} \in Q(J')$. (Observe that $E(Q, J)$ can be efficiently constructed, since K_Q is fixed.) For all $J' \in E(Q, J)$, the conjunction $\wedge_{f \in J'} \beta(f)$ contains at most K_Q conjuncts, each of which is in disjunctive normal form. Consequently, one can efficiently transform $\wedge_{f \in J'} \beta(f)$ into a DNF formula by distributivity of conjunction over disjunction. Denote this DNF formula by $\varphi_{J'}$. If J_q^β is a $\text{DNF}_{\leq k}$ instance, then each disjunct of $\varphi_{J'}$ has at most $k \cdot K_Q$ conjuncts. Then, we choose as φ the disjunction $\vee_{J' \in E(Q,J)} \varphi_{J'}$. □

Recall that Theorems A.6 and A.8 consider two classes of schema mappings: in the first, there are only st-tgds, and in the second there are only full st-tgds and t-egds. The theorems show that in the two classes, given a source DNF instance I_p^α, one can efficiently test whether a p-solution exists and, if so, materialize a universal p-solution as a DNF instance J_q^β. Moreover, the reader can verify that the algorithm applied for generating J_q^β (the chase procedure) is such that if I_p^α is a $\text{DNF}_{\leq k'}$ instance for some constant k', then J_q^β is a $\text{DNF}_{\leq k}$ instance for some other constant k (more particularly, k' depends only on k and the schema mapping). In particular, if I_p^α is tuple independent, then J_q^β is a $\text{DNF}_{\leq k}$ instance for some constant k. Now, Part 1 of Proposition 4.14 shows that the evaluation of a target UCQ Q over a universal p-solution gives the correct confidence value for every answer. Finally, Proposition A.14 shows that there is an FPRAS for Q over a DNF instances, and an FPAS over tuple-independent instances. By combining all of these results, we get the following.

[16]For the case where all the variables have the probability 1/2, an extremely simple FPAS is given in Trevisan [2002].

THEOREM A.15. *Let* $(\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping where* Σ *contains either* (1) *only st-tgds,* or (2) *only full st-tgds and t-egds. Then every target UCQ has an FPRAS over source DNF instances, and an FPAS over source tuple-independent instances.*

Observe that Theorem A.15 covers all the tractability results in the relevant rows of Table I (i.e., the fifth in each of the two parts). Next, we prove hardness results.

THEOREM A.16. *Assume* RP $\neq$ NP. *Let* Q *be a nontrivial UCQ over a schema* $\mathbf{T}$. *There is a schema mapping* $(\mathbf{S}, \mathbf{T}', \Sigma)$, *where* $\mathbf{T} \subseteq \mathbf{T}'$, *such that* Σ *contains only st-tgds and t-egds, and for all* $\theta > 0$, *there is no polynomial-time randomized* θ*-approximation for* Q *over source DNF instances.*

PROOF. In the proof of Theorem 5.6, we constructed a schema mapping that contains only st-tgds and t-egds, such that deciding whether a p-solution exists, given a source DNF instance I_p^α, is coNP-hard. Let $(\mathbf{S}_{\mathsf{sat}}, \mathbf{T}_{\mathsf{sat}}, \Sigma_{\mathsf{sat}})$ be the schema mapping constructed in the proof of Theorem 5.6. Here, we will construct a new schema mapping $(\mathbf{S}, \mathbf{T}', \Sigma)$, such that Σ contains only st-tgds and t-egds. Then, we will show a reduction from the problem of deciding whether a p-solution exists under $(\mathbf{S}_{\mathsf{sat}}, \mathbf{T}_{\mathsf{sat}}, \Sigma_{\mathsf{sat}})$ to the problem of θ-approximating the evaluation of Q under $(\mathbf{S}, \mathbf{T}', \Sigma)$.

Consider the proof of Proposition 5.8. For the query Q over $\mathbf{T}$, let $\mathbf{S}_\mathbf{T}$, $\mathbf{a}$, $J_\mathbf{a}$ and $I_\mathbf{a}$ be defined as in that proof. We also use the definition of R^s of $\mathbf{S}_\mathbf{T}$ for every relation symbol R of $\mathbf{T}$. We assume that $\mathbf{S}_\mathbf{T}$ and $\mathbf{S}_{\mathsf{sat}}$ are disjoint.

The source schema $\mathbf{S}$ is the concatenation $\langle \mathbf{S}_{\mathsf{sat}}, \mathbf{S}_\mathbf{T} \rangle$, and the target schema $\mathbf{T}'$ is the concatenation $\langle \mathbf{T}, \mathbf{T}_{\mathsf{sat}} \rangle$. The set Σ is obtained from Σ_{sat} by adding, for each relation symbol R of $\mathbf{T}$, the st-tgd d_R^{st} that copies the relation R^s to the relation R.

$$d_R^{\mathrm{st}} : \forall \mathbf{x}(R^s(\mathbf{x}) \rightarrow R(\mathbf{x}))$$

This completes the definition of $(\mathbf{S}, \mathbf{T}', \Sigma)$.

Let $\theta > 0$ be given, and suppose that A_Q^θ is a randomized algorithm that produces a θ-approximation for Q under $(\mathbf{S}, \mathbf{T}', \Sigma)$, where A_Q^θ plays the role of A in (25) (we shall shortly define K_r^γ that will play the role of I_p^α in (25)). Let I_p^α be a given DNF instance over $\mathbf{S}_{\mathsf{sat}}$. Recall that our goal is to decide whether a p-solution for I_p^α with respect to Σ_{sat} exists.

We construct the source DNF instance K_r^γ over $\mathbf{S}$ as follows. The instance K consists of all the facts of I, and all the facts of $I_\mathbf{a}$. The function γ is the same as α for the facts of I, and is a unique event variable $e_\mathbf{a}$ for every fact of $I_\mathbf{a}$ (hence, the facts of $I_\mathbf{a}$ share the same event). Finally, the function r is equal to p over $\mathsf{EVar}(\alpha)$ (i.e., $r(e) = p(e)$ for all $e \in \mathsf{EVar}(\alpha)$), and $r(e_\mathbf{a}) = q_\theta$, where q_θ is a number (that depends on θ) that we will determine later.

If there is no p-solution for I_p^α with respect to Σ_{sat}, then there is no p-solution for K_r^γ with respect to Σ, since a p-solution for K_r^γ with respect to Σ is a p-solution for I_p^α with respect to Σ_{sat} if one restricts the facts to those of $\mathbf{T}_{\mathsf{sat}}$ and $\mathbf{S}_{\mathsf{sat}}$. Hence, if there is no p-solution for I_p^α, then $conf_Q(\mathbf{a}) = 1$ (by definition). In this case, the following holds with a high (2/3) probability.

$$A_Q^\theta(K_r^\gamma, \mathbf{a}) \geq \frac{1}{1+\theta} \tag{26}$$

Now suppose that there is a p-solution for I_p^α with respect to Σ_{sat}. Clearly, in this case there is a p-solution for K_r^γ with respect to Σ, and by Proposition 4.12 it holds that $conf_Q(\mathbf{a})$ is the probability that $\mathbf{a}$ is a certain answer for K_τ^γ, where τ is a random truth assignment for $\mathsf{EVar}(\gamma)$. If $\tau(e_\mathbf{a}) = \mathbf{true}$, then every solution J for K_τ^γ must contain $J_\mathbf{a}$,

hence satisfy $\mathbf{a} \in Q(J)$. Otherwise, if $\tau(e_{\mathbf{a}}) = \mathbf{false}$, there is a solution J for K_τ^γ with no facts over $\mathbf{T}$; in this case, $\mathbf{a} \notin Q(J)$ since Q is nontrivial. Therefore, if $\tau(e_{\mathbf{a}}) = \mathbf{false}$ then $\mathbf{a}$ is not a certain answer. We conclude that the confidence of $\mathbf{a}$ is exactly the probability that $\tau(e_{\mathbf{a}})$ is $\mathbf{true}$, that is, $conf_Q(\mathbf{a}) = q_\theta$. So, we have the following since A_Q^θ is a θ-approximation.

$$A_Q^\theta(K_\tau^\gamma, \mathbf{a}) \leq (1+\theta)q_\theta \tag{27}$$

From (26) and (27), we conclude that if choose q_θ to be such that $(1+\theta)q_\theta < 1/(1+\theta)$, then we could use A_Q^θ to decide (with a high probability) whether there is a p-solution for I_p^α with respect to Σ_{sat}; clearly, $q_\theta < 1/(1+\theta)^2$ suffices. Thus, if A_Q^θ terminates in polynomial time, then we get a polynomial-time randomized solution for an NP-hard problem, which implies that NP $\subseteq$ BPP, or equivalently [Ko 1982], that RP = NP. □

Next, we show that Theorem A.16 holds even for source tuple-independent instances, if we assume that the UCQ is not near-trivial. (Recall from Section 5.3.4 that near-triviality is weaker than triviality, and that this weakening is necessary.) First, we need the following lemma that shows a useful property of a UCQ that is not near-trivial.

LEMMA A.17. *Let Q be a UCQ over a schema $\mathbf{T}$, and suppose that Q is not near-trivial. There exists a tuple $\mathbf{a}$ of constants, a ground instance J_0 over $\mathbf{T}$, a relation symbol R_0 of $\mathbf{T}$, and two tuples $\mathbf{t_a}$ and $\mathbf{t}_0$ of constants, such that the following hold.*

—$\mathbf{a} \in Q(J_0 \cup \{R_0(\mathbf{t_a})\})$
—$\mathbf{a} \notin Q(J_0 \cup \{R_0(\mathbf{t}_0)\})$.

PROOF. Assume first that Q is non-Boolean. Consider a ground instance J over $\mathbf{T}$ and a tuple $\mathbf{a}$ of constants such that $\mathbf{a} \in Q(J)$. We apply the following procedure. We iterate over all the facts of J, and for each fact $R(\mathbf{t})$, where $\mathbf{t} = (t_1, \ldots, t_k)$, we change J as follows. We remove from J the fact $R(\mathbf{t})$ and introduce a tuple $\mathbf{u} = (u_1, \ldots, u_k)$, where each u_i is a fresh constant that does not appear in J. If $\mathbf{a} \in Q(J \cup \{R(\mathbf{u})\})$, then we add $R(\mathbf{u})$ to J and continue (note that from now on $f \notin J$). Otherwise, if $\mathbf{a} \notin Q(J \cup \{R(\mathbf{u})\})$, then we choose $J_0 = J$, $R_0 = R$, $\mathbf{t_a} = \mathbf{t}$, and $\mathbf{t}_0 = \mathbf{u}$; we then report "success" and terminate.

Note that this procedure necessarily succeeds (i.e., reports "success"), since we cannot eliminate the values of $\mathbf{a}$ and still obtain $\mathbf{a}$ as an answer. Clearly, when this procedure succeeds we have that J_0, R_0, $\mathbf{t_a}$ and $\mathbf{t}_0$ are as claimed in the lemma.

Now suppose that Q is Boolean. So, $\mathbf{a}$ is necessarily the empty tuple, and we need to find J_0, R_0, $\mathbf{t_a}$ and $\mathbf{t}_0$ such that $Q(J_0 \cup \{R_0(\mathbf{t_a})\}) = \mathbf{true}$ and $Q(J_0 \cup \{R_0(\mathbf{t}_0)\}) = \mathbf{false}$. Here, we use the assumption that Q is not near-trivial, and do the following. Let J and J' be ground instances, such that $R^J = \emptyset$ if and only if $R^{J'} = \emptyset$ for each relation symbol R of $\mathbf{T}$, and moreover, $Q(J) = \mathbf{true}$ and $Q(J') = \mathbf{false}$. By the definition of near triviality, the instances J and J' exist. Now, we apply this procedure to J and $\mathbf{a} = ()$. Clearly, if the procedure succeeds, then we are done. We next show that the procedure must succeed.

Suppose, by way of contradiction, that the procedure fails. Then, in the end of the procedure, J is a ground instance such that every constant in J has exactly one occurrence in J, and no occurrences in Q. Let $J_\perp$ be obtained from J by replacing each constant with a unique null. Then, $Q(J_\perp) = \mathbf{true}$ since Q does not distinguish between a value of J and its corresponding null in $J_\perp$. Moreover, $J_\perp \to J'$ holds, since all the nonempty relations of $J_\perp$ are nonempty in J'. Thus, $Q(J') = \mathbf{true}$, which contradicts our choice of J'. □

THEOREM A.18. *Assume* RP $\neq$ NP. *Let Q be a UCQ over a schema $\mathbf{T}$, such that Q is not near-trivial. There is a schema mapping $(\mathbf{S}, \mathbf{T}', \Sigma)$, where $\mathbf{T} \subseteq \mathbf{T}'$, such that Σ contains*

only st-tgds and t-egds, and for all $\theta > 0$, there is no polynomial-time randomized θ-approximation for $\mathbf{Q}$ over source tuple-independent instances.

PROOF. Let $(\mathbf{S}_{\mathsf{cnf}}, \mathbf{T}_{\mathsf{cnf}}, \Sigma_{\mathsf{cnf}})$ be the schema mapping constructed in the proof of Theorem A.12, and for a given monotone 2-CNF formula φ, let I_p^α be the source tuple-independent instance (over $\mathbf{S}_{\mathsf{cnf}}$) constructed there. Recall that for a truth assignment τ for $\mathsf{EVar}(\alpha)$ it holds that a universal solution for I_τ^α contains the fact $E_{\mathbf{T}}(m-1, m, \mathtt{a})$ if and only if τ satisfies φ. We assume (without loss of generality) that the schemas $\mathbf{T}_{\mathsf{cnf}}$ (of the proof of Theorem A.12) and $\mathbf{T}$ (specified in the theorem currently being proved) are disjoint.

Let $\mathbf{a}$, J_0, R_0, $\mathbf{t_a}$ and $\mathbf{t}_0$ be those of Lemma A.17. The source schema $\mathbf{S}$ contains all the relation symbols of $\mathbf{S}_{\mathsf{cnf}}$, a unique relation symbol R^s for each relation symbol R of $\mathbf{T}$, a unique ternary relation symbol $\hat{E}_{\mathbf{S}}$, and a unique relation symbol $\hat{R}_0^s$ with the same arity as R_0. (Note that in $\mathbf{S}$, both R_0^s and $\hat{R}_0$ correspond to R_0.)

The source tuple-independent instance K_r^γ over $\mathbf{S}$ is defined as follows. Let J_0^s be obtained from J_0 by replacing each fact $R(\mathbf{t})$ with $R^s(\mathbf{t})$. The instance K consists of all the facts of I and all those of J_0^s. Also, K contains the facts $\hat{R}_0^s(\mathbf{t_a})$ and $\hat{E}_{\mathbf{S}}(m-1, m, \mathtt{a})$. The function γ is the same as α for the facts of I, and is **true** for the rest of the facts. Finally, the function r is equal to p (i.e., $r(e) = p(e)$ for all $e \in \mathsf{EVar}(\gamma)$).

The target schema $\mathbf{T}'$ is obtained from $\langle \mathbf{T}, \mathbf{T}_{\mathsf{cnf}} \rangle$ by adding a unique relation symbol $\hat{R}_0$ with the same arity as R_0, and a unique ternary relation symbol $\hat{E}_{\mathbf{T}}$.

Next, we construct Σ. We begin with $\Sigma = \Sigma_{\mathsf{cnf}}$. We add to Σ the following st-tgd d_R^{st} for each relation symbol R of $\mathbf{T}$; this dependency copies the relation R^s of $\mathbf{S}$ to the relation R of $\mathbf{T}$.

$$d_R^{\mathrm{st}} : \forall \mathbf{x}(R^s(\mathbf{x}) \rightarrow R(\mathbf{x})).$$

The st-tgd $d_{\hat{E}}^{\mathrm{st}}$ copies $\hat{E}_{\mathbf{S}}$ to $\hat{E}_{\mathbf{T}}$:

$$d_{\hat{E}}^{\mathrm{st}} : \forall x, y, z(\hat{E}_{\mathbf{S}}(x, y, z) \rightarrow \hat{E}_{\mathbf{T}}(x, y, z)).$$

The st-tgd $d_{\hat{R}}^{\mathrm{st}}$ copies $\hat{R}_0^s$ to $\hat{R}_0$:

$$d_{\hat{R}}^{\mathrm{st}} : \forall \mathbf{x}(\hat{R}_0^s(\mathbf{x}) \rightarrow \hat{R}_0(\mathbf{x})).$$

The st-tgd $d_{\mathbf{a}}^{\mathrm{st}}$ says that if $\hat{R}_0^s$ is nonempty, then $\hat{R}_0$ and R_0 must have a common tuple.

$$d_0^{\mathrm{st}} : \forall \mathbf{x}(\hat{R}_0^s(\mathbf{x}) \rightarrow \exists \mathbf{y}(\hat{R}_0(\mathbf{y}) \wedge R_0(\mathbf{y}))).$$

Finally, the t-egd d^{egd} says that if $\hat{E}_{\mathbf{T}}$ and $E_{\mathbf{T}}$ have a common tuple, then $\hat{R}_0$ has at most one tuple.

$$d^{\mathrm{egd}} : \forall \mathbf{x}, \mathbf{y}, \mathbf{z}(\hat{E}_{\mathbf{T}}(\mathbf{z}) \wedge E_{\mathbf{T}}(\mathbf{z}) \wedge \hat{R}_0(\mathbf{x}) \wedge \hat{R}_0(\mathbf{y}) \rightarrow \mathbf{x} = \mathbf{y}).$$

We first show that there is a p-solution (hence, a universal p-solution by Proposition 5.3) for K_r^γ, by constructing a (deterministic) such p-solution. Recall from the proof of Theorem A.12 that there exists a deterministic p-solution for I_p^α under the schema mapping $(\mathbf{S}_{\mathsf{cnf}}, \mathbf{T}_{\mathsf{cnf}}, \Sigma_{\mathsf{cnf}})$; let J be such a p-solution. To obtain from J a deterministic p-solution for K_r^γ, we add to J all the facts of J_0, the fact $\hat{E}_{\mathbf{T}}(m-1, m, \mathtt{a})$, and the facts $\hat{R}_0(\mathbf{t_a})$ and $R_0(\mathbf{t_a})$. It is easy to verify that the resulting J is indeed a p-solution, by observing that each dependency is satisfied for every random source instance K_τ^γ.

Since a p-solution exists, we will complete the proof by showing that $\mathbf{a}$ is a certain answer for a random instance K_τ^γ if and only if τ satisfies φ; then, we will apply Proposition 4.12 as in the proof of Theorem 5.9.

Suppose first that τ satisfies φ. Let J be a solution for K_τ^γ. By the construction of K_τ^γ and by the dependency d_R^{st}, the solution J must contain all the facts of J_0; hence, from Lemma A.17 we get that in order to show that $\mathbf{a} \in Q(J)$ it suffices to prove that $R_0(\mathbf{t_a})$ is also in J. By the construction of K_τ^γ and by the dependencies $d_{\hat{R}}^{\text{st}}$, and $d_{\hat{E}}^{\text{st}}$, the solution J must also contain the fact $\hat{R}_0(\mathbf{t_a})$, and the fact $\hat{E}_{\mathbf{T}}(m-1, m, \mathsf{a})$. By the t-tgd d_0^{st}, the relations $\hat{R}_0$ and R_0 have a tuple in common, since K_τ^γ contains $\hat{R}_0^{\text{s}}(\mathbf{t_a})$. Since τ satisfies φ, the proof of Theorem A.12 shows that J contains the fact $E_{\mathbf{T}}(m-1, m, \mathsf{a})$. Thus, the premise of d^{egd} is satisfied, and then the conclusion of d^{egd} implies that $\hat{R}_0$ has exactly one tuple in J, which must be $\mathbf{t_a}$ (we already now that $\hat{R}_0(\mathbf{t_a})$ is in J). We conclude that $\mathbf{t_a}$ is the common tuple of $\hat{R}_0$ and R_0, and hence, $R_0(\mathbf{t_a})$ is also in J, as claimed.

Now, suppose that τ violates φ. We construct a solution J as follows. Let J_{cnf} be a universal solution for I_τ^α with respect to Σ_{cnf}. The proof of Theorem A.12 shows that the instance J_{cnf} does not contain $E_{\mathbf{T}}(m-1, m, \mathsf{a})$, since τ violates φ. The relation J is obtained from $\langle J_{\mathsf{cnf}}, J_0\rangle$ by adding the facts $\hat{E}_{\mathbf{T}}(m-1, m, \mathsf{a})$, $\hat{R}_0(\mathbf{t_a})$, $\hat{R}_0(\mathbf{t}_0)$ and $R_0(\mathbf{t}_0)$. It is easy to see that J is a solution for K_τ^γ. Note that the premise of d^{egd} is violated, so there is no problem with J having both $\hat{R}_0(\mathbf{t_a})$ and $\hat{R}_0(\mathbf{t}_0)$. Now, the only facts of J with relation symbols from $\mathbf{T}$ are those of $J_0 \cup \{R_0(\mathbf{t}_0)\}$. By the choice of $\mathbf{t}_0$ we have that $\mathbf{a} \notin Q(J_0 \cup \{R_0(\mathbf{t}_0)\})$, and hence $\mathbf{a} \notin Q(J)$. Thus, $\mathbf{a}$ is not a certain answer, as claimed. □

ACKNOWLEDGMENTS

We thank Laura Haas, Elad Hazan, Renée J. Miller and C. Seshadhri for fruitful discussions. We especially thank Peter Haas for providing valuable comments on this work.

REFERENCES

ABITEBOUL, S. AND SENELLART, P. 2006. Querying and updating probabilistic information in XML. In *Proceedings of the Extending Data Base Technology Conference (EDBT)*. ACM, 1059–1068.

AGRAWAL, P., BENJELLOUN, O., SARMA, A. D., HAYWORTH, C., NABAR, S. U., SUGIHARA, T., AND WIDOM, J. 2006. Trio: A system for data, uncertainty, and lineage. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. ACM, 1151–1154.

AHARONI, R., BERGER, E., GEORGAKOPOULOS, A., PERLSTEIN, A., AND SPRÜSSEL, P. 2011. The max-flow min-cut theorem for countable networks. *J. Comb. Theory, Ser. B 101,* 1, 1–17.

ARENAS, M., FAGIN, R., AND NASH, A. 2010. Composition with target constraints. In *Proceedings of the International Conference on Database Theory (ICDT)*. ACM, 129–142.

BARBARÁ, D., GARCIA-MOLINA, H., AND PORTER, D. 1992. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng. 4,* 5, 487–502.

BEERI, C. AND VARDI, M. Y. 1984. A proof procedure for data dependencies. *J. ACM 31,* 4, 718–741.

BOULOS, J., DALVI, N. N., MANDHANI, B., MATHUR, S., RÉ, C., AND SUCIU, D. 2005. MYSTIQ: A system for finding more answers by using probabilities. In *Proceedings of the SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 891–893.

CASANOVA, M. A., FAGIN, R., AND PAPADIMITRIOU, C. H. 1984. Inclusion dependencies and their interaction with functional dependencies. *J. Comput. Syst. Sci. 28,* 1, 29–59.

CHANDRA, A. K. AND MERLIN, P. M. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the Symposium on Theory of Computing (STOC)*. ACM, 77–90.

COHEN, S., KIMELFELD, B., AND SAGIV, Y. 2008. Incorporating constraints in probabilistic XML. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 109–118.

DALVI, N. N. AND SUCIU, D. 2004. Efficient query evaluation on probabilistic databases. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. Morgan Kaufmann, 864–875.

DALVI, N. N. AND SUCIU, D. 2007a. The dichotomy of conjunctive queries on probabilistic structures. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 293–302.

DALVI, N. N. AND SUCIU, D. 2007b. Efficient query evaluation on probabilistic databases. *VLDB J. 16,* 4, 523–544.

DONG, X., HALEVY, A., AND YU, C. 2009. Data integration with uncertainty. *VLDB J. 18,* 2, 469–500.

DONG, X., HALEVY, A. Y., AND YU, C. 2007. Data integration with uncertainty. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. ACM, 687–698.

FAGIN, R., KIMELFELD, B., AND KOLAITIS, P. G. 2010. Probabilistic data exchange. In *Proceedings of the International Conference on Database Theory (ICDT)*. ACM, 76–88.

FAGIN, R., KOLAITIS, P. G., MILLER, R. J., AND POPA, L. 2005a. Data exchange: Semantics and query answering. *Theoret. Comput. Sci. 336,* 1, 89–124.

FAGIN, R., KOLAITIS, P. G., AND POPA, L. 2005b. Data exchange: Getting to the core. *ACM Trans. Datab. Syst. 30,* 1, 174–210.

FAGIN, R., KOLAITIS, P. G., POPA, L., AND TAN, W.-C. 2005c. Composing schema mappings: Second-order dependencies to the rescue. *ACM Trans. Datab. Syst. 30,* 4, 994–1055.

FRÉCHET, M. 1951. Sur les tableaux de corrélation dont les marges sont donnés. *Annales de l'Université de Lyon 4*, 53–57.

FUHR, N. AND RÖLLEKE, T. 1997. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst. 15,* 1, 32–66.

GALIL, Z. AND TARDOS, É. 1986. An $o(n^2(m + n \log n) \log n)$ min-cost flow algorithm. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1–9.

GOTTLOB, G. AND NASH, A. 2008. Efficient core computation in data exchange. *J. ACM 55,* 2, 1–49.

GRÄDEL, E., GUREVICH, Y., AND HIRSCH, C. 1998. The complexity of query reliability. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 227–234.

GREEN, T. J., KARVOUNARAKIS, G., IVES, Z. G., AND TANNEN, V. 2007a. Update exchange with mappings and provenance. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. ACM, 675–686.

GREEN, T. J., KARVOUNARAKIS, G., AND TANNEN, V. 2007b. Provenance semirings. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 31–40.

GREEN, T. J. AND TANNEN, V. 2006. Models for incomplete and probabilistic information. *IEEE Data Eng. Bull. 29,* 1, 17–24.

GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. 1988. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin.

HALL, P. 1935. On representatives of subsets. *J. London Math. Soci. 10*, 26–30.

HELL, P. AND NEŠETŘIL, J. 2004. *Graphs and Homomorphisms*. Oxford Lecture Series in Mathematics and Its Applications, 28. Oxford University Press.

IMIELINSKI, T. AND LIPSKI, W. 1984. Incomplete information in relational databases. *J. ACM 31,* 4, 761–791.

JERRUM, M., VALIANT, L. G., AND VAZIRANI, V. V. 1986. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci. 43*, 169–188.

KARP, R. M., LUBY, M., AND MADRAS, N. 1989. Monte-Carlo approximation algorithms for enumeration problems. *J. Algorithms 10,* 3, 429–448.

KIMELFELD, B., KOSHAROVSKY, Y., AND SAGIV, Y. 2008. Query efficiency in probabilistic XML models. In *Proceedings of the SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 701–714.

KIMELFELD, B. AND SAGIV, Y. 2007. Maximally joining probabilistic data. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 303–312.

KO, K.-I. 1982. Some observations on the probabilistic algorithms and NP-hard problems. *Inf. Process. Lett. 14,* 1, 39–43.

KOCH, C. 2008. Approximating predicates and expressive queries on probabilistic databases. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 99–108.

LUBY, M. AND VELICKOVIC, B. 1996. On deterministic approximation of DNF. *Algorithmica 16,* 4/5, 415–433.

MAIER, D., MENDELZON, A. O., AND SAGIV, Y. 1979. Testing implications of data dependencies. *ACM Trans. Datab. Syst. 4,* 4, 455–469.

MORGENSTERN, D. 1956. Einfache beispiele zweidimensionaler verteilungen. *Mitteilingsblatt für Mathematische Statistik 8*, 234–235.

PAPADIMITRIOU, C. H. AND YANNAKAKIS, M. 1984. The complexity of facets (and some facets of complexity). *J. Comput. Syst. Sci. 28,* 2, 244–259.

RE, C., DALVI, N. N., AND SUCIU, D. 2007. Efficient top-k query evaluation on probabilistic data. In *Proceedings of the International Conference on Data Engineering (ICDE)*. IEEE, 886–895.

RE, C. AND SUCIU, D. 2007a. Efficient evaluation of HAVING queries on a probabilistic database. In *Proceedings of the International Workshop on Database and Programming Languages (DBPL)*. Lecture Notes in Computer Science Series, vol. 4797, Springer, 186–200.

RE, C. AND SUCIU, D. 2007b. Materialized views in probabilistic databases for information exchange and query optimization. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. ACM, 51–62.

SARMA, A. D., DONG, X., AND HALEVY, A. 2009. *Uncertainty In Data Integration*. Springer, New York, Chapter 7, 185–222.

SARMA, A. D., DONG, X., AND HALEVY, A. Y. 2008a. Bootstrapping pay-as-you-go data integration systems. In *Proceedings of the SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 861–874.

SARMA, A. D., THEOBALD, M., AND WIDOM, J. 2008b. Exploiting lineage for confidence computation in uncertain and probabilistic databases. In *Proceedings of the International Conference on Data Engineering (ICDE)*. IEEE, 1023–1032.

SENELLART, P. AND ABITEBOUL, S. 2007. On the complexity of managing probabilistic XML data. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*. ACM, 283–292.

SHAKED, M. AND SHANTHIKUMAR, J. 1994. *Stochastic Orders and Their Applications*. Academic Press, San Diego, CA.

TODA, S. AND OGIWARA, M. 1992. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM J. Comput. 21,* 2, 316–328.

TREVISAN, L. 2002. A note on deterministic approximate counting for k-DNF. In *Proceedings of the Electronic Colloquium on Computational Complexity (ECCC). 9,* 069.

VALIANT, L. G. 1979. The complexity of computing the permanent. *Theor. Comput. Sci. 8*, 189–201.

ZACHOS, S. 1988. Probabilistic quantifiers and games. *J. Comput. Syst. Sci. 36,* 3, 433–451.

ZUCKERMAN, D. 1996. On unapproximable versions of NP-complete problems. *SIAM J. Comput. 25,* 6, 1293–1304.

Received August 2010; revised May 2011; accepted May 2011