Schema Design for Uncertain Databases[⋆]

Anish Das Sarma, Jeffrey Ullman, Jennifer Widom
{anish,ullman,widom}@cs.stanford.edu

Stanford University

Abstract.
We address schema design in uncertain databases. Since uncertain data is relational in nature, decomposition becomes a key issue in design. Decomposition relies on dependency theory, and primarily on functional dependencies.
We study the theory of functional dependencies (FDs) for uncertain relations. We define several kinds of *horizontal* FDs and *vertical* FDs, each of which is consistent with conventional FDs when an uncertain relation doesn't contain any uncertainty. In addition to standard forms of decompositions allowed by ordinary relations, our FDs allow more complex decompositions specific to uncertain data. We show how our theory of FDs can be used for lossless decomposition of uncertain relations. We then present algorithms and complexity results for three fundamental problems with respect to FDs over ordinary and uncertain relations: (1) *Testing* whether a relation instance satisfies an FD; (2) *Finding* all FDs satisfied by a relation instance; and (3) *Inferring* all FDs that hold in the result of a query over uncertain relations with FDs. We also give a sound and complete axiomatization of horizontal and vertical FDs. We look at keys as a special case of FDs. Finally, we briefly consider uncertain data that contains *confidence* values.

1 Introduction

With the recent increase in applications such as scientific and sensor databases, data cleaning and integration, information extraction, and approximate query processing, the field of *uncertain databases* is attracting considerable interest [5, 16, 19, 44]. While a large body of previous and recent work addresses issues like modeling (e.g., [3, 5, 13, 15, 25, 31, 28]), querying (e.g., [15, 17, 20, 24, 31, 41]), and designing systems (e.g., [6, 16, 19, 32, 44]) for uncertain data, there is little past work relevant to *dependency theory* for uncertain databases. Obviously there has been a significant amount of previous work in dependency theory for ordinary relations (refer to [36, 42]), and some work for XML data [8, 11], but none of this past work can be applied directly to uncertain databases.

Dependency theory for uncertain databases introduces a number of new and interesting problems. Moreover, these dependencies give rise to better database designs and useful decompositions of uncertain relations, which are different in nature from decompositions of ordinary relations.

[⋆] This work was supported by a Microsoft Graduate Fellowship, by the National Science Foundation under grants IIS-0324431 and IIS-0414762, and by grants from the Boeing and Hewlett-Packard Corporations.

In this paper, we revisit the theory of functional dependencies (FDs) in the context of uncertain data. As in ordinary databases, FDs in uncertain databases can be useful in several ways: (1) FDs can be used to *decompose* uncertain relations, resulting in compression, faster querying, and a better overall design of the database; (2) The detection of keys (and their violations) can be useful in understanding and verifying properties of the data; (3) Knowledge of FDs can aid in efficient storage and indexing.

At first, we might think that when data is uncertain, there is nothing nontrivial we can say about FDs. But we shall see that the mechanisms we know and love, both from FDs and from multivalued dependencies, appear in uncertain data! They even lead to decompositions with lossless joins or with other means to reconstruct the originals. While some decompositions are like the usual BCNF decompositions, some others are more complex in nature.

We study FD theory for data models whose basic construct for uncertainty is *alternatives*. Several previously proposed data models for uncertainty (for example, [4, 13, 15, 21, 25, 33, 38]) use similar constructs. The semantics of uncertain relations represented in any data model (including those not based on alternatives) are defined through a set *possible worlds*, and we study functional dependencies directly over sets of possible worlds as well. Alternatives in a tuple specify a nonempty finite set of possible values for the tuple. For example:

$$\boxed{\text{(Thomas, Main St.)} \;||\; \text{(Tom, Maine St.)}}$$

contains a tuple with two alternatives giving the two possible values for the tuple. Our data model is defined formally in Section 2.

Our contributions are introduced briefly in the next few subsections, with details and results in Sections 4–9. We review past work and relate it to our contributions in Section 3, and we conclude with future work in Section 10.

1.1 FD Definitions and Axiomatization

In Section 4 we define two kinds of FDs: *horizontal* and *vertical*. While horizontal FDs primarily capture dependencies across the alternatives within tuples, vertical FDs apply to a relation as a whole and capture dependencies across tuples. Over ordinary relations, our definition of horizontal and vertical FDs become conventional FDs. We then study definitions of functional dependency over sets of possible worlds and relate these definitions to horizontal and vertical FDs.

In Section 5 we prove that for both horizontal and vertical FDs, the axioms of *reflexivity*, *transitivity*, and *augmentation* are sound and complete. Interestingly, our complete axiomatization for uncertain data is based on the same three axioms used for ordinary relations, even though the FD definitions appear quite different.

1.2 Decomposition

In Section 6 we show that both horizontal and vertical FDs can be used for the lossless decomposition of an uncertain relation R. Given the FDs that hold in R, we give algorithms for the decomposition of R into a set of *constituents*, $\mathcal{C}(\mathcal{R})$. We then show how to obtain a SQL query Q such that executing Q on the constituents of R gives back R; i.e., $Q(\mathcal{C}(\mathcal{R})) \equiv \mathcal{R}$.[1]

1.3 Test, Find, and Infer

In Section 7 we look at the following three problems, analyzing their complexity and giving algorithms for the tractable cases:

- *FD Testing*: Given a relation R and a horizontal or vertical FD f, *test* whether R satisfies f.
- *FD Finding*: Given a relation R, *find* all horizontal and vertical FDs that R satisfies.
- *FD Inference:* Given a set of input relations $\mathcal{R}$, the set of FDs $\mathcal{F}_{R_i}$ that hold in each $R_i \in \mathcal{R}$ respectively, and a query Q over $\mathcal{R}$, determine all FDs that are guaranteed to hold in the result $Q(\mathcal{R})$.

Since our FDs degenerate to conventional FDs when there is no uncertainty, we also study the three problems above for the special case of ordinary relations, and point out interesting differences. For example, on performing a join query over ordinary relations, the set of FDs satisfied by the result can only increase. However, we can never infer any new FDs for a join query over uncertain relations!

1.4 Keys

In Section 8 we study keys as a special case of FDs in greater detail. In ordinary relations, keys indicate uniqueness of values. In uncertain relations, keys are more complicated. They can indicate the nonduplication of values horizontally, vertically, or in the entire relation. (Informally, a set of attributes X in R is *nonduplicated* if no X value appears twice.) First we revisit the issues of Section 1.3 for keys. We then study the relationship between horizontal and vertical keys in a relation R and the presence of *nonduplicated* attribute values in R.

1.5 Confidence Values

Sometimes models for uncertain data include *confidence values* [13, 15, 16, 18, 25]. In Section 9 we briefly revisit the definitions and results from the paper when confidence values are present on alternatives. (Models with confidence values but no alternatives are *probabilistic databases*, not relevant to this paper.) We provide a stronger definition of FDs that allows for decomposition even in the presence of confidence values.

[1] We use $\equiv$ and not $=$ because $Q(\mathcal{C}(\mathcal{R}))$ may return a different representation of the uncertainty in R, depending on the query-execution strategy.

2 Preliminaries

Recall the definition of alternatives from the previous section. We define an *uncertain relation* R to be a bag of tuples, where each tuple is a nonempty, finite set of alternatives. Semantically, an uncertain relation R represents a set of *possible worlds* (or *possible instances*), each of which is an ordinary relation. The possible worlds for an uncertain relation are a bag obtained by choosing one alternative value for each tuple, in all possible ways. Note that an uncertain relation has at least one possible world. An ordinary relation by definition has exactly one possible world. An uncertain relation is equivalent to an ordinary relation if and only if every tuple contains exactly one alternative.

Given an uncertain relation R, define the *H-relation* (for "horizontal relation") of a tuple $t \in R$, denoted H-relation(t), as the ordinary relation consisting of all the alternatives in t.

We say that two relations are *equivalent* if they have the same set of possible worlds. An uncertain relation with each tuple having a single alternative is equivalent to an ordinary relation. A model for uncertain data is *unique* if two different relations in the model can never be equivalent. The following theorem was proved in [22].

Theorem 1 (Uniqueness). The uncertain-relations model is unique. □

We can compose a family of models using constructs similar to alternatives, e.g., with alternative values for one or a set of attributes (known as *or-sets*), or a combination of or-sets and tuple alternatives. None of the models composed of these constructs is more expressive than the uncertain relations we consider, and hence are not considered in the rest of the paper.

3 Related Work

While the study of uncertainty in databases and the study of functional dependencies for ordinary relations have separately received considerable attention for several decades now, there is little history combining the two bodies of work. Dependency theory has been widely studied from the 70s and we refer the reader to [36, 42]. Uncertainty in databases also have been studied extensively in the last twenty years [2, 3, 24, 25, 31, 32], as well as more recently [5, 15–17, 19, 28, 41, 44]. However, this past work focuses on data modeling, query processing, and system design.

The problem of *dependency satisfaction* for an uncertain relation with incomplete information was studied in [26, 30, 35]. Conceptually, given a dependency, the possible worlds of an uncertain relation are restricted to only those that satisfy the dependency. Similarly, [43] considers *completions* of relations with null values such that the completed relation satisfies the constraint. More recently, in the *information-source-tracking* method [39] of modeling uncertainty, [40] addressed similar issues of adjusting the model and query answering in the presence of key constraints. Therefore, the focus in all these papers is to transform the uncertain relation into one that satisfies certain dependencies, and on query answering thereof. In comparison, we are interested in preserving the uncertain relation (and possible worlds) and defining when a dependency is satisfied by it. Note that an individual possible world need not satisfy the dependency in our scenario. Moreover, we address the problems of decomposition and inference, not considered in any of these previous papers.

Reference [34] studies two kinds of functional dependencies for incomplete relations: *strong FDs*, which hold in every possible world, and *weak FDs*, which hold in at least one possible world of the incomplete relation. Our work defines functional dependencies directly on the representation of uncertain relations. We exhibit benefits in decomposition when defining FDs directly over uncertain relations, instead of through possible worlds.

Reference [7] briefly considers *factorizing* uncertain relations represented in the *WSD* data model. These factorizations can be thought of as decomposing a relation such that the cross product of the various components results in the original relation. While the work in [7] is restricted to cross-product factorizations, we are interested in a much wider class of decompositions. Specifically, we look at the traditional BCNF-like decompositions, as well as newer kinds of decompositions specific to uncertain data. Moreover, as in traditional database literature, our decompositions are based on functional dependencies, while functional dependencies or the associated problems of testing, finding, and inference are not addressed in [7].

Finally, our work is not to be confused with the area of *inconsistent databases* (e.g., [9, 10, 27, 45]), in which a database that does not satisfy a set of constraints is "repaired." Possible repairs to the database result in a set of possible worlds, i.e., an uncertain database. Hence, an uncertain database is created because of violation of constraints, but constraints are not defined on an uncertain database.

4 Functional Dependencies

In this section we define functional dependencies over uncertain relations and prove some basic results about them. Throughout the paper we use $R, S, T, \dots$ to refer to relations, $A, B, \dots$ to denote single attributes, $X, Y, \dots$ to denote sets of attributes, and when clear from the context, the relation name (e.g., R) to denote the set of all attributes in the relation.

Section 4.1 presents horizontal FDs and Section 4.2 presents vertical FDs. As will be clear shortly, horizontal FDs are a straightforward adaptation of the conventional definition of FDs. Hence several results on horizontal FDs throughout the paper borrow from past literature on regular FDs. Vertical FDs are significantly more interesting and challenging. In Section 4.3 we relate both horizontal and vertical FDs to what they imply on possible worlds.

4.1 Horizontal FDs

We define three kinds of horizontal FDs for an uncertain relation R.

Definition 1 (Horizontal FD).

1. R satisfies FD $X \to Y$ of type H_1 (denoted $X \to_{H_1} Y$) if the conventional FD $X \to Y$ holds in the union of the H-relations of all tuples in R.

2. R satisfies FD $X \rightarrow Y$ of type H_2 (denoted $X \rightarrow_{H_2} Y$) if the conventional FD $X \rightarrow Y$ holds separately in the H-relations of all tuples in R.
3. A tuple t in R satisfies FD $X \rightarrow Y$ of type H_3 (denoted $X \rightarrow_{H_3(t)} Y$) if the conventional FD $X \rightarrow Y$ holds in the H-relation of tuple t. □

A horizontal FD of type H_1 is equivalent to the corresponding conventional FD if the relation R is an ordinary relation. All three FD types permit some lossless decomposition of R, to be discussed in Section 6. When $X \rightarrow_{H_2} Y$ holds in R, all H-relations in R satisfy $X \rightarrow Y$, but the mapping between X and Y values could be different in each H-relation. In contrast when $X \rightarrow_{H_1} Y$ holds in R, all H-relations in R satisfy $X \rightarrow Y$ with the same mapping between X and Y. $X \rightarrow_{H_3(t)} Y$ allows some tuples (specifically, t) to satisfy $X \rightarrow Y$ while others don't. The following straightforward theorem relates the three types of horizontal FDs.

Theorem 2. For any uncertain relation R, $(X \rightarrow_{H_1} Y) \Rightarrow (X \rightarrow_{H_2} Y) \Rightarrow (X \rightarrow_{H_3(t)} Y)$, for any t in R. □

Example 1. Consider the uncertain relations R, S, and T shown below.

ID	**R(SSN, Name, Address)**
r_1	(1,Thomas,Main St.) || (1,Thomas,Maine St.)
r_2	(2,Alice,Poplar Ave)

ID	**S(SSN, Name, Address)**
s_1	(1,Thomas,Main St.) || (1,Thomas,Maine St.)
s_2	(1,Tom,Main St.) || (2,Alice,Poplar Ave)

ID	**T(SSN, Name, Address)**
t_1	(1,Thomas,Main St.) || (1,Tom,Maine St.)
t_2	(2,Alice,Poplar Ave)

R satisfies `SSN→Name` for all horizontal FD types, S satisfies `SSN→Name` for horizontal FD types H_2 and $H_3(s_i)$ for all tuples $s_i \in S$ but not H_1, and T satisfies `SSN→Name` only for FD type $H_3(t_2)$ and not for FD types H_1, H_2, or $H_3(t_1)$. □

4.2 Vertical FDs

Definition 2 (Vertical FD). An uncertain relation R satisfies a vertical FD $X \rightarrow Y$ (denoted $X \rightarrow_V Y$) if and only if the following conditions hold:

1. $\forall A \in (Y - X)$, the H-relation of each tuple in R satisfies $X \rightarrow\!\!\!\rightarrow A$. ($\rightarrow\!\!\!\rightarrow$ denotes the conventional multivalued dependency.)
2. For any two tuples t_1, t_2 in R, let T_1^x (T_2^x respectively) be the set of all tuples in H-relation(t_1) (H-relation(t_2) respectively) that have the value x for attribute X. Let $S_1^{x,A}$ ($S_2^{x,A}$ respectively) be the set of all distinct A values that appear in tuples of T_1^x (T_2^x respectively). Then $\forall A \in (Y - X)$, $\forall x$, at least one of the following holds: (1) $S_1^{x,A} = \emptyset$, (2) $S_2^{x,A} = \emptyset$, (3) $S_1^{x,A} = S_2^{x,A}$. □

Intuitively, for $X \to_V Y$ to hold, a given X value should functionally determine the set of possible values for each attribute in $(Y - X)$. The set of values for each attribute of $(Y - X)$ only depends on the value of X and all these sets are independent of one another.

In our definition we impose the condition $X \twoheadrightarrow A, \forall A \in (Y - X)$ and not the weaker $X \twoheadrightarrow (Y - X)$ because under the weaker condition our definition would not allow certain kinds of decompositions allowed by our current definition (discussed in Section 6). More importantly, it would also not satisfy some desirable axioms such as the *splitting rule*[2], satisfied by our current definition (Section 5), as illustrated by the following example.

Example 2. Consider the following relation R(SSN, Name, Address).

ID	**R(SSN, Name, Address)**
r_1	(1,Thomas,Main St.) || (1,Thomas,Maine St.)
r_2	(2,Bill,Poplar Ave) || (2,William,Poplar Ave)
r_3	(1,Thomas,Main St.) || (1,Thomas,Maine St.) || (2,Bill,Poplar Ave) || (2,William,Poplar Ave)

R satisfies `SSN`$\to_V$`NameAddress`: It can be seen that `SSN`$\twoheadrightarrow$`Name` and `SSN`$\twoheadrightarrow$`Address` in the H-relations of each tuple above (Condition 1 of Definition 2), and the set of `Name` and `Address` values for every distinct `SSN` value is the same for every H-relation (Condition 2 of Definition 2). R also satisfies `SSN`$\to_V$`Name` and `SSN`$\to_V$`Address`. Note that R does not satisfy any horizontal FD with only `SSN` on the left side. Now consider the following simple relation S(SSN, Name, Address).

ID	**S(SSN, Name, Address)**
s_1	(1,Tom,Main St.) || (1,Thomas,Maine St.)

S does not satisfy `SSN`$\to_V$`NameAddress`. Since `SSN`$\twoheadrightarrow$`NameAddress` is satisfied in the only tuple's H-relation in S, if we changed Condition 1 of the vertical FD definition to $X \twoheadrightarrow (Y - X)$, then `SSN`$\to_V$`NameAddress` would be satisfied above. However, `SSN`$\to_V$`Name` or `SSN`$\to_V$`Address` would still not be satisfied, violating the splitting rule. □

The following theorem shows that vertical FDs degenerate to regular FDs in the special case of an uncertain relation being equivalent to an ordinary relation.

Theorem 3. If uncertain relation R is equivalent to an ordinary relation S, then the conventional FD $X \to Y$ holds for S if and only if $X \to_V Y$ holds for R. □

Proof. Let $X \to_V Y$ hold in R. Then $\forall A \in (Y - X)$, $X \twoheadrightarrow A$ (Condition 1 of vertical FD definition) is satisfied in the H-relation of each tuple in R. Since R is equivalent to

[2] The splitting rule allows us to infer $X \to Z$ for any $Z \subseteq Y$ when $X \to Y$ holds.

an ordinary relation, every tuple has exactly one alternative. Therefore, the set of all A values corresponding to a given X value in Condition 2 is a singleton set. Hence by Condition 2, if two tuples in R have the same X value, they also have the same A value for every $A \in (Y - X)$. Hence $X \rightarrow Y$ holds in S.

Conversely, if $X \rightarrow Y$ holds in S, then both conditions of the vertical FD definition are satisfied for $X \rightarrow_V Y$ in R, since each tuple in R corresponding to a tuple in S has one alternative.

4.3 Possible Worlds

In this section we consider defining FDs in terms of possible worlds. Since the semantics of uncertain relations are based on a set of possible worlds, one may wonder why we can't define FDs in terms of possible worlds, as in [34]. Next we consider the most natural definition of FDs in terms of possible worlds. We then relate this definition in terms of possible worlds to our definitions of horizontal and vertical FDs. Finally, we describe why horizontal and vertical FDs are more appropriate than the definition in terms of possible worlds, for schema design in uncertain databases.

Definition 3 (Possible-World FD). *An uncertain relation R, with possible worlds $\{P_1, \ldots, P_m\}$, satisfies a possible-world FD $X \rightarrow Y$ (denoted $X \rightarrow_{PW} Y$) if and only if each possible world P_i satisfies the conventional FD $X \rightarrow Y$.*

Example 3. Consider the following single-tuple uncertain relation R(SSN, Name, Address).

ID	R(SSN, Name, Address)
r_1	(1,Tom,Main St.) || (1,Thomas,Maine St.)

R has two possible worlds: $P_1(SSN, Name, Address)$ containing the single tuple (1,Tom,Main St.) and $P_2(SSN, Name, Address)$ containing the single tuple (1,Thomas,Maine St.). Since both P_1 and P_2 satisfy SSN$\rightarrow$NameAddress, R satisfies SSN$\rightarrow_{PW}$NameAddress. However, R does not satisfy the vertical FD SSN$\rightarrow_V$NameAddress, nor does it satisfy any of the horizontal FDs SSN$\rightarrow_{H_1}$NameAddress, SSN$\rightarrow_{H_2}$NameAddress, or SSN$\rightarrow_{H_3(r_1)}$NameAddress. □

Next let us try to understand how possible-world FDs relate to horizontal and vertical FDs. The example above showed that even if a possible-world FD $X \rightarrow_{PW} Y$ holds, none of the corresponding horizontal or vertical FDs is guaranteed to hold. Conversely. our next example shows that the possible-world FD cannot be inferred from horizontal FDs of type H_2 or $H_3(t)$, or from vertical FDs.

Example 4. Consider relation S from Example 1:

ID	S(SSN, Name, Address)
s_1	(1,Thomas,Main St.) || (1,Thomas,Maine St.)
s_2	(1,Tom,Main St.) || (2,Alice,Poplar Ave)

S satisfies SSN$\rightarrow$Name for horizontal FD types H_2 and $H_3(s_i)$ for all tuples $s_i \in S$. However, S does not satisfy SSN$\rightarrow_{PW}$Name: The possible world obtained by selecting the first alternative from each of s_1 and s_2 does not satisfy the conventional FD SSN$\rightarrow$Name.

To prove that even vertical FDs don't imply possible-world FDs, consider the relation R from Example 2. R satisfies SSN$\rightarrow_{V}$Name. However R does not satisfy SSN$\rightarrow_{PW}$Name since R has the following possible world, which does not satisfy SSN$\rightarrow$Name.

R(SSN, Name, Address)
(1,Thomas,Main St.)
(2,Bill,Poplar Ave)
(2,William,Poplar Ave)

□

Finally, the following theorem shows that the only remaining implication between a horizontal or a vertical FD, and the corresponding possible-world FD holds.

Theorem 4. *For any uncertain relation R,* $(X \rightarrow_{H_1} Y) \Rightarrow (X \rightarrow_{PW} Y)$. □

Proof. Let the union of the H-relations of all tuples in R be $\mathcal{R}$. Since R satisfies $(X \rightarrow_{H_1} Y)$, $\mathcal{R}$ satisfies the conventional FD $(X \rightarrow Y)$. Therefore, any subset of $\mathcal{R}$ also satisfies $(X \rightarrow Y)$. Since each possible world of R is a subset of $\mathcal{R}$, each possible world of R also satisfies $(X \rightarrow Y)$; hence, R satisfies $(X \rightarrow_{PW} Y)$.

While the definition of possible-world FDs described above is simple and intuitive, it isn't suitable for schema design. In particular, unlike the definition of horizontal and vertical FDs, possible-world FDs in uncertain relations don't give rise to useful decompositions. Consider relation R from Example 3. While R satisfies SSN$\rightarrow_{PW}$Name, there is no lossless decomposition of R based on this FD. The standard BCNF decomposition of R based on SSN$\rightarrow$Name is lossy. However, as we shall see in Section 6, both horizontal and vertical FDs always give rise to useful (BCNF-like or other) decompositions. Hence, we focus on horizontal and vertical FDs for the rest of the paper.

5 Sound and Complete Axioms

In this section we prove that the traditional Armstrong's axioms for conventional FDs are also sound and complete for horizontal and vertical FDs over uncertain data. First we review the definitions of soundness and completeness, then Section 5.1 presents results for horizontal FDs and Section 5.2 for vertical FDs.

Definition 4 (Soundness). A set of axioms $\mathcal{A}$ is *sound* if for any relation R and a set of FDs $\mathcal{F}$ satisfied by R, any FD f derived from $\mathcal{F}$ using $\mathcal{A}$ holds in R. □

Definition 5 (Completeness). A set of axioms $\mathcal{A}$ is *complete* if for any set of FDs $\mathcal{F}$, if FD f is true in every relation R satisfying $\mathcal{F}$, then f can be derived from $\mathcal{F}$ using $\mathcal{A}$. □

Definition 6 (Strong Completeness). A set of axioms $\mathcal{A}$ is *strongly complete* if for any set of FDs $\mathcal{F}$ on attributes U, there exists a relation $R(U)$ satisfying all and only FDs that can be derived from $\mathcal{F}$ and $\mathcal{A}$. □

Intuitively strong completeness proves the existence of a relation which shows that any FD that cannot be derived from $\mathcal{F}$ and $\mathcal{A}$ is not implied by $\mathcal{F}$.

5.1 Horizontal FDs

Theorem 5. The following Armstrong's axioms (where $\rightarrow$ stands for one of $\rightarrow_{H_1}$, $\rightarrow_{H_2}$, and $\rightarrow_{H_3(t)}$) are *sound* and *complete* with respect to each type of horizontal FD:

1. **Reflexivity:** If $Y \subseteq X$, then $X \rightarrow Y$.
2. **Transitivity:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.
3. **Augmentation:** If $X \rightarrow Y$ and $Z \subseteq W$, then $XW \rightarrow YZ$. □

Proof. For an uncertain relation R, consider the union of the H-relations S of all tuples in R. An FD $X \rightarrow_{H_1} Y$ holds in S if and only if $X \rightarrow Y$ holds in S. Hence if any FD can be derived using a set of axioms, the corresponding H_1 FD is sound and can be derived using the H_1 axioms. Hence the Armstrong's axioms are sound and complete for H_1 FDs. Similarly considering H-relations for either a specific tuple t or independently for all tuples in R we argue that the Armstrong's axioms are sound and complete for H_3(t) and H_2 FDs respectively.

Note that the above theorem applies to only one of H_1, H_2, and H_3(t) FDs at a time. However, if we have a combination of the different kinds of horizontal FDs, we can use Theorem 2 and Armstrong's axioms to derive further dependencies that hold in the weakest form of horizontal FDs used to derive the result.

Theorem 6. Armstrong's axioms from Theorem 5 are strongly complete for H_1, H_2, and H_3(t) FDs with respect to uncertain relations. □

Proof. Given a set $\mathcal{F}$ of horizontal FDs over a set of attributes U, we use the strong completeness of regular FDs [14] to construct an ordinary relation $R(U)$ which satisfies all and only regular functional dependencies derivable from $\mathcal{F}$ using the Armstrong's axioms. Now we construct an uncertain relation $S(U)$ with a single tuple whose alternatives are exactly the set of tuples in R. $S(U)$ satisfies all and only horizontal FDs derivable from $\mathcal{F}$ and the Armstrong's axioms for horizontal FDs.

5.2 Vertical FDs

The next three lemmas prove the soundness of Armstrong's axioms for vertical FDs. We then prove the completeness and strong completeness of Armstrong's axioms.

Lemma 1 (Reflexivity). Given an uncertain relation R with sets of attributes $X, Y \subseteq R$, if $Y \subseteq X$, then $X \rightarrow_V Y$ holds in R. □

Proof. Since $(Y - X) = \emptyset$, it can be seen easily that both the conditions of Definition 2 are satisfied for $X \rightarrow_V Y$.

Lemma 2 (Transitivity). Given an uncertain relation R with sets of attributes $X, Y, Z \subseteq R$, if $X \rightarrow_V Y$ and $Y \rightarrow_V Z$, then $X \rightarrow_V Z$. □

Proof. Consider a tuple t in R. By Definition 2, we have:

(1) $\forall T_i \in (Y - X)$, we have $X \twoheadrightarrow T_i$ in H-relation(t).

(2) $\forall V_j \in (Z - Y)$, $Y \twoheadrightarrow V_j$ in H-relation(t).

From (1), applying the Union rule for MVDs [14] on H-relation(t), we have $X \twoheadrightarrow (Y - X)$. Using Augmentation [14], we get:

(3) $X \twoheadrightarrow Y$ in H-relation(t).

Combining (2) and (3) above using the transitivity rule for MVDs [14], we have:

(4) $\forall V_j \in (Z - Y)$, $X \twoheadrightarrow (V_j - Y)$ in H-relation(t).

Hence,

(5) $\forall V_j \in (Z - Y)$, $X \twoheadrightarrow V_j$ in H-relation(t).

Finally, since all attributes in $(Z - X)$ are in at least one of $(Z - Y)$ and $(Y - X)$, combining Equations (1) and (5), we get Condition 1 of Definition 2 for $X \rightarrow_V Z$:

(6) $\forall A_k \in (Z - X)$, $X \twoheadrightarrow A_k$ in H-relation(t).

Next we prove Condition 2 of Definition 2 for $X \rightarrow_V Z$. Now consider two tuples t_1 and t_2 in R, and a particular X value x. If $X \neq x$ in all tuples of H-relation(t_1) or all tuples of H-relation(t_2), we have Condition 2 satisfied for $X \rightarrow_V Z$ for $X = x$. If not, by the vertical FD definition for $X \rightarrow_V Y$, the set of all $Y - X$ that appear in tuples having value x for attribute X is the same in H-relation(t_1) and H-relation(t_2). Call this set S_1. Consider a particular value $s \in S_1$. Since $(Y - X) = s$ in at least one tuple each of H-relation(t_1) and H-relation(t_2), by the vertical FD definition for $Y \rightarrow_V Z$, the set of all A values for $A \in (Z - Y)$ values associated with s is the same in H-relation(t_1) and H-relation(t_2). Let this set be $S_2(s)$. For any $B \in (Z - X)$, the set of all B values that appears in tuples with value x for attribute X is the same in both H-relation(t_1) and H-relation(t_2): It is equal to the B attributes in $\cup_{s \in S_1} S_2(s)$ if $B \in (Z - Y)$ and the B attributes of S_1 if $B \in (Y - X)$. Hence, Condition 2 of Definition 2 for $X \rightarrow_V Z$ holds.

Lemma 3 (Augmentation). Given an uncertain relation R with sets of attributes $X, Y, Z, W \subseteq R$, if $X \rightarrow_V Y$ and $Z \subseteq W$, then $XW \rightarrow_V YZ$. □

Proof. Since $X \rightarrow_V Y$, for every tuple t in R we have $X \twoheadrightarrow A$ in H-relation(t) for every $A \in (Y - X)$. By the augmentation rule for MVDs over ordinary relations, we have $XW \twoheadrightarrow A$ for all $A \in (Y - X)$ and hence for all $A \in (YZ - XW)$. Hence Condition 1 of Definition 2 for $XW \rightarrow_V YZ$ is satisfied.

Now consider Condition 2 for $XW = xw$. Let t_1 and t_2 be two tuples in R. If either of H-relation(t_1) and H-relation(t_2) does not have any tuple with $XW = xw$, Condition 2 is satisfied. Consider an attribute $A \in (YZ - XW) = (Y - XW)$. Using Condition 2 for $X \rightarrow_V Y$, the set of all A values that appears with $X = x$ is the same in H-relation(t_1) and H-relation(t_2). Let this set of $Y - X$ values be S. For both H-relation(t_1) and H-relation(t_2) since $X \twoheadrightarrow A$, $A \notin XW$, and since there exists a tuple with $W = w$, the set of all A values in tuples having $XW = xw$ is also S. Hence Condition 2 is satisfied for all $A \in (YZ - XW)$.

Example 5. Recall Example 2 giving an uncertain relation in which the splitting rule is not satisfied by a weaker definition of vertical FDs. Our current definition satisfies the splitting rule as it follows from Armstrong's axioms. □

The following theorem proves the strong completeness, and hence also completeness, of Armstrong's axioms for vertical FDs.

Theorem 7 (Completeness). The Armstrong's axioms are strongly complete for vertical FDs with respect to uncertain relations. □

Proof. We use the completeness of Armstrong's axioms for conventional FDs over ordinary relations [14]. Consider a set of vertical FDs $\mathcal{F}$ over a set of attributes U. The strong completeness of regular FDs allows us to construct an ordinary relation $R(U)$ which satisfies all and only regular functional dependencies derivable from regular FD counterpart of $\mathcal{F}$ using the Armstrong's axioms. Since $R(U)$ is also an uncertain relation, using Theorem 3, we know that $R(U)$ satisfies all only those vertical FDs that can be derived from $\mathcal{F}$ and Armstrong's axioms for vertical FDs.

Theorem 8. The Armstrong's axioms are a sound and complete axiomatization for vertical FDs with respect to uncertain relations. □

Proof. The soundness of Armstrong's axioms follows from Lemmas 1, 2, and 3 and the completeness follows from Theorem 7

6 Decomposition

We address the problem of decomposition of an uncertain relation when it satisfies horizontal or vertical functional dependencies. We shall see that we can use techniques similar to those for conventional functional and multivalued dependencies. Some of our decompositions are like the usual BCNF decompositions for ordinary relations, but there are also new decomposition forms that can be used in some cases. We first briefly describe the standard possible-worlds semantics [3] of relational queries over uncertain relations, necessary for our discussion of decomposition, and then show how uncertain relations can be decomposed based on horizontal (Section 6.1) and vertical (Section 6.2) FDs.

Consider uncertain database D (consisting of one or more uncertain relations) with possible worlds $D_1, \ldots, D_n$. Each D_i is a set of ordinary relations, one corresponding to each relation in the database. (When D contains multiple relations, the set of possible worlds of D is obtained by taking the cross-product of the set of possible worlds for each relation.) Let $Q(D_i)$ be the ordinary relation obtained by evaluating Q on D_i. The result of performing a relational query Q over D is an uncertain relation whose possible worlds are $\{Q(D_1), \ldots, Q(D_n)\}$, assuming such an uncertain relation exists.

6.1 Horizontal FDs

Consider an uncertain relation $R(XYZ)$ that satisfies $X \rightarrow_{H_1} Y$. We can decompose R into two relations R_1 and R_2 as follows. $R_1(XZ) = \mathit{\Pi}_{XZ}(R)$, where $\mathit{\Pi}$ stands for the project relational operator. $R_2(XY)$ is an ordinary relation consisting of all distinct XY values appearing in R.[3] The decomposition of R into R_1 and R_2 is lossless. Informally, R_1 retains all the alternatives of R but projects them onto XZ, and R_2 stores the mappings between X and Y. Since $X \rightarrow_{H_1} Y$, each X value appears exactly once in R_2, resulting in a compressed representation of R. The following theorem proves that R can be obtained by joining R_1 and R_2.

Theorem 9. For any uncertain relation $R(XYZ)$ satisfying $X \rightarrow_{H_1} Y$, if $R_1(XZ) = \mathit{\Pi}_{XZ}(R)$ and $R_2(XY)$ is an ordinary relation containing all distinct XY values that appear in some alternative of R, then $R \equiv (R_1 \bowtie_X R_2)$.[4] □

Proof. Let R have n possible worlds $P_1(XYZ), \ldots, P_n(XYZ)$. By the definition of R_1, the possible worlds of R_1 are $\mathit{\Pi}_{XZ}(P_1), \ldots, \mathit{\Pi}_{XZ}(P_n)$. R_2 has exactly one possible world $P(XY)$ containing all distinct XY values in R. Hence, the possible worlds of $(R_1 \bowtie_X R_2)$ are $(P \bowtie_X (\mathit{\Pi}_{XZ}(P_1))), \ldots, (P \bowtie_X (\mathit{\Pi}_{XZ}(P_n)))$, which are equal to $P_1, \ldots, P_n$ since every XY value appearing in any P_i also appears in P.

Example 6. Consider relation R from Example 1 satisfying `SSN`$\rightarrow_{H_1}$`Name`. After decomposing R as described above, we have the following two relations:

R_1**(SSN, Address)**	R_2**(SSN, Name)**
(1,Main St.) || (1,Maine St.)	(1,Thomas)
(2,Poplar Ave)	(2,Alice)

□

Next let us consider horizontal FD type H_2. Suppose relation $R(XYZ)$ satisfies $X \rightarrow_{H_2} Y$. We now know that every H-relation in R satisfies $X \rightarrow Y$ but the same X value could be mapped to different Y values in the H-relations of different tuples. We give two schemes for decomposing R in this case.

The first decomposition scheme introduces a new attribute $\mathcal{I}$ in R, denoting a unique tuple identifier for each tuple. All alternatives of a given tuple have the same value in R. Once we have unique identifiers in each tuple, we have the H_1 horizontal FD $\mathcal{I}X \rightarrow_{H_1} Y$ satisfied in R. We can now decompose R into $R_1(\mathcal{I}XZ)$ and $R_2(\mathcal{I}XY)$ as in the H_1 FD case discussed above. By Theorem 9, the decomposition is lossless and $R \equiv \mathit{\Pi}_{XYZ}(R_1 \bowtie_{\mathcal{I}X} R_2)$.

The second decomposition scheme first horizontally partitions R such that $X \rightarrow_{H_1} Y$ holds in every horizontal partition and then applies the decomposition described above. We partition the set of tuples in R, say $\{t_1, \ldots, t_n\}$, into m groups to form m

[3] R_2 can be obtained from R using a query containing the *flatten* operator [1] designed to create an ordinary relation from an uncertain relation. First project R onto XY, then flatten the result into an ordinary relation.

[4] Recall "$\equiv$" represents equivalence of uncertain relations, i.e., same set of possible worlds.

uncertain relations $R^1(XYZ), \ldots, R^m(XYZ)$. We describe the process of partition shortly. The key point is our partition ensures that $X \rightarrow_{H_1} Y$ holds in every R^i. Each R^i is now decomposed into $R_1^i(XZ)$ and $R_2^i(XY)$ based on the H_1 horizontal FD. The following theorem, which follows from Theorem 9 and the horizontal partition of R into $R^1, \ldots, R^m$, shows that our decomposition is lossless and that R can be obtained by a relational query over the decomposed components.

Theorem 10. For any uncertain relation $R(XYZ)$ satisfying $X \rightarrow_{H_2} Y$, the decomposition of R into $R_1^i(XZ)$ and $R_2^i(XY)$, $1 \leq i \leq m$, described above is lossless: $R \equiv \cup_i (R_1^i \bowtie_X R_2^i)$. □

Let us now turn to the horizontal partitioning of R into $R^1, \ldots, R^m$. Intuitively, we would like to have the fewest partitions (i.e., smallest m) allowing us to decompose each R^i based on $X \rightarrow_{H_1} Y$. Two tuples t_1 and t_2 can appear together in any R^i if only if $X \rightarrow_{H_1} Y$ holds in an uncertain relation composed of just t_1 and t_2. In other words, t_1 and t_2 should not constitute a violation of the H_1 FD. For every pair of tuples in R we check whether all X values are mapped to the same Y value in all alternatives of both the tuples. Let us suppose S stores all pairs of tuples that do constitute a violation. Finding the fewest number of partitions is equivalent to solving the following NP-hard graph coloring problem, which can be solved approximately using well-known techniques [37].

> Given a set $U = \{1, \ldots, n\}$ and a set S of pairs (i, j), $1 \leq i < j \leq n$, let $U_1, \ldots, U_m$ be a partition of U. That is, $U = \cup_{k=1}^m U_k$ and if $k \neq l$, $U_k \cap U_l = \emptyset$. Find the smallest partition of U, i.e. minimize m, such that if $(i, j) \in S$, then $\forall k \; i \notin U_k$ or $j \notin U_k$.

Example 7. Consider the relation S from Example 1 satisfying SSN$\rightarrow_{H_2}$Name. Introducing tuple identifiers s_1 and s_2 for the two tuples and then decomposing S according to the first scheme mentioned above, we have the following decomposed components:

S($\mathcal{I}$, SSN, Address)
(s_1,1,Main St.) || (s_1,1,Maine St.)
(s_2,1,Main St.) || (s_2,2,Poplar Ave)

S($\mathcal{I}$, SSN, Name)
(s_1,1,Thomas)
(s_2,1,Tom)
(s_2,2,Alice)

Under the second scheme of decomposition s_1 and s_2 constitute different horizontal partitions, and each of them is decomposed similarly. □

Finally, suppose $R(XYZ)$ satisfies $X \rightarrow_{H_3(t)} Y$ for a tuple t in R, we horizontally partition R into two components: R^1 consisting of tuple t only and R^2 consisting of the rest of the tuples. R^1 is then partitioned based on $X \rightarrow_{H_1} Y$ and R^2 remains as is.

6.2 Vertical FDs

Consider an uncertain relation $R(XYZ)$ that satisfies $X \rightarrow_V Y$ where $Y = \{A_1, \ldots, A_m\}$. Intuitively, the X value in any alternative uniquely determines the set of possible A_i values for every $A_i \in Y$, and these sets of values are independent of one another. We can therefore decompose R by retaining all the alternative values of R in one relation, and for each A_i, creating a relation that gives the mapping between every X value and the set of possible A_i values it determines. We have the following decomposition of R into $R_0(XZ)$, $R_1(XA_1)$, $R_2(XA_2), \ldots, R_m(XA_m)$. $R_0(XZ) = \Pi_{XZ}(R)$. $R_i(XA_i)$ contains n tuples, one for each distinct X value that appears in R. For a given X-value x_0, suppose the set of all A_i values that appear in tuples having value x_0 for attributes X is $\{a_i^1, \ldots, a_i^{k_i}\}$. Then the tuple corresponding to $X = x_0$ has k_i alternatives $(x_0, a_i^1), \ldots, (x_0, a_i^k)$.[5] The following theorem shows that R can be reconstructed from its components using a relational query, thus proving that the decomposition is lossless.

Theorem 11. For any uncertain relation $R(XYZ)$ satisfying $X \rightarrow_V Y$, where $Y = \{A_1, \ldots, A_m\}$, let $R_0(XZ) = \Pi_{XZ}(R)$. For all $A_i \in Y$, let $R_i(XA_i)$ be an uncertain relation containing one tuple for every distinct X-value in R, with alternatives corresponding to all possible A_i values it appears with. Then $R \equiv (R_0 \bowtie_X R_1 \bowtie_X R_2 \bowtie_X \ldots \bowtie_X R_m)$. □

Proof. If R contains K distinct X values in its alternatives, by the definition of $R_i(XA_i)$, each possible world of R_i contains K tuples, one for each distinct X value. The possible worlds of R_i list all possible combinations of mappings between X and A_i. Now consider $S(XY) = (R_1 \bowtie_X R_2 \bowtie_X \ldots \bowtie_X R_m)$. The possible worlds of S are obtained by joining all combinations of possible worlds from R_1 to R_m. Since every possible world of every R_i contains exactly K tuples containing all the distinct X values, every possible world of S also has K tuples containing all distinct K values. The possible worlds of S list all combinations of mappings between X and Y. Let these possible worlds of S be $Q_1(XY), \ldots, Q_l(XY)$.

Let the set of possible worlds of R be $PW(R) = \{P_1(XYZ), \ldots, P_n(XYZ)\}$. Since $R_0(XZ) = \Pi_{XZ}(R)$, the set of possible worlds of R_0 is $\{\Pi_{XZ}(P_1), \ldots, \Pi_{XZ}(P_n)\}$. Hence, the set of possible worlds of $T = R \bowtie_X S$, $PW(T)$, is:

$$\{(Q_1 \bowtie_X (\Pi_{XZ}(P_1))), \ldots, (Q_1 \bowtie_X (\Pi_{XZ}(P_n))), \ldots, (Q_n \bowtie_X (\Pi_{XZ}(P_n)))\}.$$

We claim $PW(R) = PW(T)$: Every $P_i(XYZ) \in PW(T)$ because $\Pi_{XZ}(P_i)$ is joined with every combination of mappings between X and Y, and in particular it is joined with some Q_j containing exactly the mappings between X and Y present in P_i. Conversely, every $(Q_j \bowtie_X (\Pi_{XZ}(P_i)))$ is equal to P_k for some k: Consider the specific mappings between X and Y in Q_j and the tuples in $\Pi_{XZ}(P_i)$. P_i was obtained from R by choosing one alternative from every tuple in R. Applying the definition of vertical FDs on each H-relation of R, if we replace the Y values in the alternatives chosen in P_i with Y values determined by the mappings in Q_j, the resulting alternatives must also be present in the H-relations of R. Hence if we pick the resulting alternatives from each tuple, we get some possible world P_k, which is exactly equal to $(Q_j \bowtie_X (\Pi_{XZ}(P_i)))$.

[5] R_i can be obtained from R using a query containing the *group-alts* operator [1] designed to create an ordinary relation from an uncertain relation. First project R onto XA_i, then group alternatives by X.

Example 8. Consider the following slightly modified relation from Example 2 satisfying `SSN`$\rightarrow_V$`NameAddress`.

R(SSN, Name, Address,Phone)
(1,Tom,Main St.,P1) || (1,Tom,Maine St.,P1)
(2,Bill,Poplar Ave,P2) || (2,William,Poplar Ave,P2)
(1,Tom,Main St.,P1) || (1,Tom,Maine St.,P1) || (2,Bill,Poplar Ave,P3) || (2,William,Poplar Ave,P3)

Decomposing R as described above, we have the following three components:

R_0**(SSN, Phone)**
(1,P1)
(2,P2)
(1,P1) || (2,P3)

R_1**(SSN, Name)**
(1,Tom)
(2,Bill) || (2,William)

R_2**(SSN, Address)**
(1,Main St.) || (1,Maine St.)
(2,Poplar Ave)

□

Let us now revisit the weaker definition of vertical FDs discussed in Section 4.2, which imposes $X \twoheadrightarrow Y$, instead of $X \twoheadrightarrow A_i$, in the H-relation of every tuple. Under this weaker definition the decomposition of R described above would be not lossless. We can however decompose R into $R_0(XZ)$ and $R_1(XY)$, where like $R_i(XA_i)$, R_1 gives mappings between X and Y. The following abstract example shows that the decomposition into R_i's can result in an exponentially more compact representation of R.

Example 9. Consider the following single tuple relation $R(X, A_1, \ldots, A_n, Z)$ containing alternatives with $X = 1$, $Z = 1$, and all 2^n combinations of 0s and 1s for A_1 to A_n.

R(X,A_1,. . .,A_n,Z)
(1,0,. . .,0,1) || . . . || (1,1,. . .,1,1)

R satisfies $X \rightarrow_V Y$, where $Y = \{A_1, \ldots, A_n\}$. Decomposing R into two relations $R_0(XZ)$ and $R_1(XY)$ gives:

R_0**(X,Z)**
(1,1)

R_1**(X,**A_1**,. . .,**A_n**)**
(1,0,. . .,0) || . . . || (1,1,. . .,1)

However, decomposing into $n + 1$ relations gives an exponentially (in n) more compact representation:

R_0(**X**,**Z**)	R_1(**X**,A_1)	$\cdots$	R_n(**X**,A_n)
(1,1)	(1,0) || (1,1)		(1,0) || (1,1)

□

7 Test, Find, and Infer

In this section we consider the problems of testing (Section 7.1), finding (Section 7.2), and inference (Section 7.3) of FDs. We study the problems for conventional FDs over ordinary relations as well as for horizontal and vertical FDs over uncertain relations. Not surprisingly the solution techniques for all of the problems are similar for conventional and horizontal FDs, so we discuss them together. While the complexity of some problems remains the same for conventional or horizontal FDs over ordinary relations and vertical FDs over uncertain relations, some problems become more challenging for the case of vertical FDs. Even more interestingly, inference of join queries over uncertain relations is actually easier than that for ordinary relations!

7.1 Testing

Recall the testing problem: Given a relation instance R and an FD f, determine whether R satisfies f. The testing problem is simple for conventional FDs over ordinary relations. Even horizontal FD testing can be reduced easily to conventional FD testing: For horizontal FD type H_1 we test whether the union of the H-relation of all tuples in R satisfies f, for FD type H_2 we test whether the H-relation of each tuple in R satisfies f, and for FD type $H_3(t)$ we test whether H-relation(t) satisfies f.

Next consider testing whether uncertain relation R having tuples $t_1, \dots, t_n$ satisfies vertical FD $X \rightarrow_V Y$, where

$$(Y - X) = \{A_1, A_2, \dots, A_m\}, Z = (R - Y - X).$$

This problem also can be solved easily by checking for both the conditions of Definition 2, as shown in Algorithm 1.

7.2 Finding

Recall the finding problem: Given a relation instance R, find all FDs satisfied by R. Finding horizontal FDs in an uncertain relation is similar to finding conventional FDs over ordinary relations. For horizontal FD type H_1 we find all FDs satisfied by the union of the H-relations of all tuples in R, for FD type H_2 we find all FDs satisfied by the H-relation of every tuple in R, and for FD type $H_3(t)$ we find all FDs satisfied by H-relation(t).

We address the problem of finding conventional FDs over ordinary relations in Section 7.2 and turn to finding vertical FDs over uncertain relations in Section 7.2.

```
Condition 1, the H-relation($t_j$) of each tuple $t_j$ in $R$ satisfies $X \twoheadrightarrow A_i$, $1 \leq i \leq m$: For every $X$-value $x$ appearing in H-relation($t_j$):
    1. Find the set $S_Z$ of all $Z$-values that appear in tuples with value $x$ for attribute $X$
    2. $\forall i$, find the set $S^j_{A_i}(x)$ of all $A_i$-values that appear in tuples with value $x$ for attribute $X$
    3. Compute the cross-product of sets $S^j_{A_1}(x), \ldots, S^j_{A_m}(x)$, $S^j_Z(x)$ and for every element $(a_1 \ldots a_m a_Z)$ in the result check whether the tuple $(x a_1 \ldots a_m a_Z)$ is present in H-relation($t_j$). If not, return "$X \rightarrow_V Y$ not satisfied."
Condition 2: For every $X$-value $x$ appearing in any alternative of $R$, for each $A_i$, check whether $S^1_{A_i}(x) = \ldots = S^n_{A_i}(x)$. If not, return "$X \rightarrow_V Y$ not satisfied."
Return "$X \rightarrow_V Y$ satisfied."
```

Algorithm 1: Testing whether R satisfies $X \rightarrow_V Y$.

Conventional FDs Before solving the problem of finding conventional FDs in ordinary relations, we define the notion of *closed sets* for a relation and give some properties. Previous work such as [29] has studied algorithms for finding all FDs in ordinary relations. Here we take an alternative approach: We give algorithms for generating all closed sets of a relation, and show that closed sets completely characterize conventional FDs for ordinary relations.

Consider an ordinary relation R. We define an instance-level notion of closed sets for R. A similar schema-level notion of closed sets was first introduced in [12] as a characterization of FDs. A set of attributes X in R is a *closed set for* R if for any attribute $A \notin X$, R does not satisfy $X \rightarrow A$. Intuitively, no attribute outside of X is functionally determined by X. In this section we also use traditional FD rules (such as the union and transitivity rules) at the instance-level. All rules for FD's hold on each relation instance.

The following theorem shows that closed sets completely characterize the set of FDs satisfied by R.

Theorem 12. Consider an ordinary relation R, and let S be the set of all closed sets for R. R satisfies $X \rightarrow Y$ if and only if $\forall s \in S$, $(X \subseteq s) \Rightarrow (Y \subseteq s)$. □

Proof. If: Suppose $\forall s \in S$, $(X \subseteq s) \Rightarrow (Y \subseteq s)$ but R does not satisfy $X \rightarrow Y$. Let X^+ be the set of all attributes A_i such that $X \rightarrow A_i$. By the union rule of FDs [14] we have $X \rightarrow X^+$. Moreover, if $A \notin X^+$, then R does not satisfy $X^+ \rightarrow A$ (because otherwise by transitivity $X \rightarrow A$). Therefore X^+ is a closed set for R and since $X \rightarrow X$, $X \subseteq X^+$ but $Y \nsubseteq X^+$, contradicting our assumption.

Only if: Suppose R satisfies $X \rightarrow Y$. Consider a closed set $s \in S$. Suppose $X \subseteq s$, then $s \rightarrow X$, and $X \rightarrow Y$. Hence by transitivity $s \rightarrow Y$. Therefore $Y \subseteq s$.

Algorithm 2 shows how to generate all the closed sets for R. We first compare all pairs of tuples in R to generate a *base set* of closed sets as follows. Given a pair of tuples t_1 and t_2 in R, if t_1 and t_2 have the same set of values for attributes in X and different values for all other attributes, then X is a closed set: for any attribute $A \notin X$, tuples t_1 and t_2 express a violation of $X \rightarrow A$. Once we have the base set of closed

```
1: Base Set: Initialize base set of closed sets $B = \emptyset$. For every pair of tuples $t_1$ and $t_2$ in $R$:
    1. Find the set of all attributes $X$ in which $t_1$ and $t_2$ have the same value.
    2. Set $B = B \cup \{X\}$.
2: All Closed Sets: Let $B = \{b_1, \ldots, b_n\}$. Initialize $S = \{b_1\}$. For $i = 2..n$, do:
    1. $\forall s_j \in S$, add $s_j \cap b_i$ to $S$.
    2. Set $S = S \cup \{b_i\}$.
```

Algorithm 2: Finding all closed sets in R.

sets obtained by pairwise tuple comparisons, we successively find intersections of these closed sets to generate all closed sets. The following lemma shows that Algorithm 2 generates only correct closed sets, and the next theorem shows that it generates all the closed sets in R.

Lemma 4. If X and Y are closed sets, then $X \cap Y$ is also a closed set. □

Proof. Let $Z = X \cap Y$. For any attribute $A \notin Z$, either $A \notin X$ or $A \notin Y$. Suppose without loss of generality $A \notin X$. Then $X \not\rightarrow A$. Therefore, $Z \not\rightarrow A$. To see why, if $Z \rightarrow A$, then using $X \rightarrow Z$ and transitivity we get $X \rightarrow A$.

Theorem 13. Algorithm 2 generates all closed sets of R. □

Proof. We show that any closed set of R can be obtained by successive intersections of the basic set of closed sets obtained by pairwise comparisons of tuples in R in Step 1. Suppose Z is a closed set. Let $A_1, \ldots, A_m$ be all attributes of R not in Z. $\forall i$, $Z \not\rightarrow A_i$. Therefore, there exists a pair of tuples t_{i_1} and t_{i_2} in R such that t_1 and t_2 have the same value for all attributes of Z but have different values for attribute A_i. In Algorithm 2 suppose we constructed base closed set S_i when we compared t_{i_1} and t_{i_2}. We have $Z \subseteq S_i$ and $A_i \notin S_i$. Therefore, $Z = (S_1 \cap \ldots \cap S_m)$.

Vertical FDs We give a PTIME (in the combined size of the input and output) algorithm for finding all vertical FDs that an uncertain relation R satisfies. We first find all vertical FDs of the form $X \rightarrow_V A$, where $A \in R$ and $X \subseteq R$, that R satisfies; we can then use the union rule of FDs to find all vertical FDs satisfied by R. (Since the union rule follows from Armstrong's axioms, vertical FDs satisfy the union rule.)

Let us consider finding all FDs of the form $X \rightarrow_V A$ for one attribute A. (We repeat the procedure for each attribute A in R.) Note that if R satisfies $X \rightarrow_V A$, then $\forall X' \supset X$, R also satisfies $X' \rightarrow_V A$. We can use this idea to start by testing $(R - A) \rightarrow_V A$. If $(R - A) \rightarrow_V A$ is satisfied, for every possible subset Y of $(R - A)$ obtained by removing a single attribute from $(R - A)$, we test whether $Y \rightarrow_V A$ is satisfied by R. So on, we recurse over every set of attributes Y for which $Y \rightarrow_V A$ is satisfied by R, testing subsets of attributes if they haven't been tested before. Our algorithm finds all nontrivial FDs with singleton right sides (i.e., FDs of the form $X \rightarrow_V A$ where $A \notin X$) in polynomial time in the number of such FDs.

7.3 Inference

Recall the inference problem: given a set of input relations $\mathcal{R}$, for each $R_i \in \mathcal{R}$ the set of FDs $\mathcal{F}_{R_i}$ that hold in each R_i, and a query Q over $\mathcal{R}$, determine all FDs that are guaranteed to hold in the result $Q(\mathcal{R})$. We consider two variants of the problem: when we only have the schema of $\mathcal{R}$, and when have the relation instances for $\mathcal{R}$. We consider arbitrary SPJ queries whose results can always be represented using an uncertain relation. Note from Theorem 1, whenever the result is representable using an uncertain relation, there is a unique representation. Hence we are interested in inferring FDs for the unique representation of the result.

Once again, the inference problem for horizontal FDs over uncertain relations is solved as in the case of conventional FDs over ordinary relations, studied in Section 7.3. Inference of vertical FDs over uncertain relations is considered in Section 7.3.

Conventional FDs The intractability of the version of the inference problem where we only have the input relation's schema has been established in previous work [2, 23]. When the input includes the schema and data, then the inference problem can easily be seen to be polynomially-solvable.

Theorem 14. Given a set of input relation instances $\mathcal{R}$ and an SPJ query Q over $\mathcal{R}$, we can infer all nontrivial FDs with singleton right sides that are satisfied by the result in PTIME (in the combined size of the input and output). □

Proof. Since the query can be answered in PTIME, we first obtain the query result. Then, in PTIME, we can find all the FDs satisfied by the result as discussed in Section 7.2.

Vertical FDs By Theorem 3, the hardness result from Section 7.3 carries over to the case of vertical FDs over uncertain relations as well. The subclass of join queries however illustrates an interesting difference between inference in ordinary relations and uncertain relations. The following theorem shows that for join queries over uncertain relations, without looking at the input data, we cannot infer any new vertical FDs involving the join attribute on the right side. Therefore, this version of the inference problem is trivial, and in fact easier than inferring conventional FDs over ordinary relations!

Theorem 15. Given uncertain relations $R(X, B), S(B, Y)$, where X and Y are a set of one or more attributes, set of FDs F_R, F_S that hold in R and S respectively, and join query $Q = (R \bowtie_B S)$, no FD with B on the right side is guaranteed to hold in the result of Q. □

Proof. We construct uncertain relation instances $R(A, B)$ and $S(B, C)$, where R satisfies $A \rightarrow_V B$, S satisfies $B \rightarrow_V C$, but the join of R and S does not satisfy $A \rightarrow_V B$. (Our construction can be extended easily for schemas with X and Y constituting more than one attribute.)

$R(A, B)$ has one tuple with two alternatives: $[(a, b_1) \ || \ (a, b_2)]$ and $S(B, C)$ has two tuples: (b_1, c_1) and (b_2, c_2). Clearly R satisfies $A \rightarrow_V B$ and S satisfies $B \rightarrow_V$

C. The result $T(A, B, C) = R \bowtie S$ has one tuple with two alternatives: $[(a, b_1, c_1) \mid\mid (a, b_2, c_2)]$. Although T still satisfies $B \rightarrow_V C$, it does not satisfy $A \rightarrow_V B$.

Clearly when R does not satisfy $A \rightarrow_V B$, we can similarly construct S such that the join result still does not satisfy $A \rightarrow_V B$.

Note for conventional FDs over ordinary relations, when a join query is performed over a set of input relations to obtain result R, all FDs involving attributes of R satisfied by any input relation are still satisfied by R. In addition, R may satisfy new FDs not satisfied by any input relation. Interestingly, vertical FDs over uncertain relations do not display this behavior: As shown by the example in the proof above, although $A \rightarrow_V B$ was satisfied by R and both the attributes of R were projected onto the result, still the result did not satisfy $A \rightarrow_V B$.

Finally, as in the case of conventional FDs over ordinary relations, for inference of SPJ queries when the input includes data, we first answer the query and then find all vertical FDs in the result.

8 Keys

We study the special case of *keys*. Given an uncertain relation R, a set of attributes X, $X \subseteq R$, is an H_1, H_2, $H_3(t)$, or vertical *key* if $X \rightarrow_{H_1} R$, $X \rightarrow_{H_2} R$, $X \rightarrow_{H_3(t)} R$, or $X \rightarrow_V R$ respectively. It can be seen easily that all the algorithms and results of our paper naturally carry over for the special case of keys.

Theorem 16. All definitions, algorithms, and results from this paper, and in particular the complexity result of Section 7.3 and the result of Theorem 15, carry over for keys as a special case of FDs. □

Next we study the relationship of keys to the following notions of nonduplication of attribute values.

Definition 7. Given an uncertain relation R and a set of attributes X in R, we say that X is:

- *Vertically nonduplicated (VND)* if no two alternatives from different tuples in R have the same value for attributes in X.
- *Horizontally nonduplicated (HND)* if no two alternatives of the same tuple in R have the same value for attributes in X.
- *Totally nonduplicated (TND)* if no two alternatives in R have the same value for attributes in X. (Equivalent to VND and HND.)

The following theorem summarizes all the implication relationships we can draw between the above notions and keys. The theorem lists the strongest forms of the relationships. We can of course infer others using relationships within the definitions above, such as (X is NDT) ⇒ (X is NDH), or relationships of FDs, such as ($X \rightarrow_{H_1} R$) ⇒ ($X \rightarrow_{H_2} R$).

Theorem 17. 1. (X is TND) ⇒ ($X \rightarrow_{H_1} R$)

2. (X is TND) $\Rightarrow$ ($X \rightarrow_V R$)
3. (X is HND) $\Leftrightarrow$ ($X \rightarrow_{H_2} R$) □

Proof. We prove each of the three statements of the theorem below:

1. Since X is TND, X is also nonduplicated in the union of the H-relation of all tuples in R. Hence $X \rightarrow_{H_1} R$.
2. Since X is TND, X is HND. Hence in the H-relation of each tuple in R, $\forall A \in R$, $X \twoheadrightarrow A$. Moreover, no two alternatives from different tuples in R have the same value for attribute X. Hence, $X \rightarrow_V R$.
3. If X is HND, then in the H-relation of each tuple in R, X is unique. Hence, $X \rightarrow_{H_2} R$. Conversely, if $X \rightarrow_{H_2} R$, X is HND as each tuple contains a set of alternatives.

9 Confidence Values

Finally, we briefly look at uncertain relations with confidence values. Confidence values are attached to each alternative. For example:

(Thomas, Main St.):0.6 || (Tom, Maine St.):0.4

The sum of the confidence values of all alternatives in each tuple is equal to 1. (Refer to [15] for more details, including query semantics.)

Even in the presence of confidence values, horizontal FDs are defined as before. Decompositions based on horizontal FDs described in Section 6.1 are slightly modified as follows: Recall the decomposition of uncertain relation R into R_1 containing the various alternatives and R_2 containing the mappings between X and Y. The confidence values of each alternative are now carried over to alternatives in R_1 and the confidence value of every alternative in R_2 is 1.

However, decomposition based on vertical FDs becomes more complicated in the presence of confidence values. Specifically, if we disregard the confidence values and decompose based on the techniques in Section 6.2, in many cases the decomposition is necessarily lossy. Moreover, even if the decomposition can be lossless, it is not clear where to store the confidence values of alternatives in R. To solve this problem, we strengthen the definition of vertical FDs by adding a third condition as follows.

Definition 8 (Vertical FD). An uncertain relation R with confidence values satisfies a vertical FD $X \rightarrow Y$ (denoted $X \rightarrow_{V_C} Y$) if and only if the following conditions hold:

1. $\forall A \in (Y - X)$, the H-relation of each tuple in R satisfies $X \twoheadrightarrow A$. ($\twoheadrightarrow$ denotes the conventional multivalued dependency.)
2. For any two tuples t_1, t_2 in R, let T_1^x (T_2^x respectively) be the set of all tuples in H-relation(t_1) (H-relation(t_2) respectively) that have the value x for attribute X. Let $S_1^{x,A}$ ($S_2^{x,A}$ respectively) be the set of all distinct A values that appear in tuples of T_1^x (T_2^x respectively). Then $\forall A \in (Y - X)$, $\forall x$, at least one of the following holds: (1) $S_1^{x,A} = \emptyset$, (2) $S_2^{x,A} = \emptyset$, (3) $S_1^{x,A} = S_2^{x,A}$.

3. For any $A \in (Y - X)$, for any two tuples t_1, t_2 in R, if there exist alternatives $a_{11}, a_{12} \in t_1$ and $a_{21}, a_{22} \in t_2$, such that $a_{11}.(R-A) = a_{12}.(R-A)$, $a_{21}.(R-A) = a_{22}.(R-A)$, $a_{11}.A = a_{21}.A$, and $a_{12}.A = a_{22}.A$, then $\frac{c(a_{11})}{c(a_{21})} = \frac{c(a_{12})}{c(a_{22})}$. ($a_{ij}.Z$ is the set of values for attributes in Z in a_{ij} and $c(a_{ij})$ is the confidence value for alternative a_{ij}. □

Intuitively, for every attribute A in $(Y - X)$ functionally determined by X, the value of X also determines the confidence values for distinct A values. Now, when we decompose an uncertain relation based on vertical FD $X \rightarrow_{V_C} Y$, the distribution of confidence values for each attribute A in $(Y - X)$ is stored with the partition containing A, and the distribution of confidence values for alternative in R are stored in the partition containing all the alternatives. We illustrate using the following example.

Example 10. Consider relation R from Example 8 but now with confidence values.

R(SSN, Name, Address,Phone)
(1,Tom,Main St.,P1):0.4 || (1,Tom,Maine St.,P1):0.6
(2,Bill,Poplar Ave,P2):0.8 || (2,William,Poplar Ave,P2):0.2
(1,Tom,Main St.,P1):0.2 || (1,Tom,Maine St.,P1):0.3 || (2,Bill,Poplar Ave,P3):0.4 || (2,William,Poplar Ave,P3):0.1

R satisfies $\texttt{SSN}\rightarrow_{V_C}\texttt{NameAddress}$, and decomposing R as described above, we have the following three components:

R_0**(SSN, Phone)**
(1,P1):1.0
(2,P2):1.0
(1,P1):0.5 || (2,P3):0.5

R_1**(SSN, Name)**
(1,Tom):1.0
(2,Bill):0.8 || (2,William):0.2

R_2**(SSN, Address)**
(1,Main St.):0.4 || (1,Maine St.):0.6
(2,Poplar Ave):1.0

□

10 Conclusions and Future Work

As a step toward schema design in uncertain databases, we proposed a theory of functional dependencies for uncertain relations. We defined horizontal and vertical FDs, which give rise to useful lossless decompositions of uncertain relations. We provided a sound and complete axiomatization of both kinds of FDs. We gave algorithms for decomposition (and reconstruction thereafter) of uncertain relations. Then we looked at the problems of testing, finding and inference of FDs for ordinary and uncertain relations. Finally, we studied keys as a special case of FDs and briefly considered uncertain relations with confidence values.

Our paper, obviously, does not solve all problems related to schema design or dependency theory for uncertain relations. Instead, it suggests several directions for future work. Of course, developing a parallel theory for multivalued and other kinds of dependencies for uncertain relations would be theoretically interesting. A more detailed study of uncertain relations with confidence values is yet another avenue for future work. Finally, a notion of "uncertain dependency" for uncertain relations, capturing the fact that a large (but possibly not entire) fraction of a relation satisfies a dependency, might also be practically useful.

References

1. TriQL: The Trio query language. Available at http://i.stanford.edu/widom/triql.html.
2. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
3. S. Abiteboul, P. Kanellakis, and G. Grahne. On the Representation and Querying of Sets of Possible Worlds. *Theoretical Computer Science*, 78(1), 1991.
4. P. Andritsos, A. Fuxman, and R. Miller. Clean answers over dirty databases: A probabilistic approach. In *Proc. of ICDE*, 2006.
5. L. Antova, C. Koch, and D. Olteanu. MayBMS: Managing Incomplete Information with Probabilistic World-Set Decompositions. In *Proc. of ICDE*, 2007.
6. L. Antova, C. Koch, and D. Olteanu. MayBMS: Managing Incomplete Information with Probabilistic World-Set Decompositions (Demonstration). In *Proc. of ICDE*, 2007.
7. L. Antova, C. Koch, and D. Olteanu. World-set decompositions: Expressiveness and efficient algorithms. In *Proc. of ICDT*, 2007.
8. M. Arenas. Normalization theory for XML. *SIGMOD Record*, 35(4):57–64, 2006.
9. M. Arenas, L. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. *TPLP*, 3(4), 2003.
10. M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent Query Answers in Inconsistent Databases. In *Proc. of ACM PODS*, 1999.
11. M. Arenas and L. Libkin. A normal form for XML documents. *TODS*, 29(1):195–232, 2004.
12. W. W. Armstrong. Dependency structures of database relationships. In *Proc. of IFIP*, pages 580–583, 1974.
13. D. Barbará, H. Garcia-Molina, and D. Porter. The Management of Probabilistic Data. *TKDE*, 4(5), 1992.
14. C. Beeri, R. Fagin, and J. H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *Proc. of ACM SIGMOD*, 1977.
15. O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *Proc. of VLDB*, 2006.
16. J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. MYSTIQ: a system for finding more answers by using probabilities. In *Proc. of ACM SIGMOD*, 2005.
17. D. Burdick, P. M. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. *J. VLDB*, 16(1):123–144, 2007.
18. R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In *Proc. of VLDB*, 1987.
19. R. Cheng, S. Singh, and S. Prabhakar. U-DBMS: A database system for managing constantly-evolving data. In *Proc. of VLDB*, 2005.
20. N. Dalvi and D. Suciu. Efficient Query Evaluation on Probabilistic Databases. In *Proc. of VLDB*, 2004.
21. A. Das Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working Models for Uncertain Data. In *Proc. of ICDE*, 2006.

22. A. Das Sarma, S. Nabar, and J. Widom. Representing uncertain data: Uniqueness, equivalence, minimization, and approximation. Technical report, Stanford InfoLab, 2005. Available at http://dbpubs.stanford.edu/pub/2005-38.
23. P. C. Fischer, J. H. Jou, and D. Tsou. Succinctness in dependency systems. *TCS*, 24, 1983.
24. N. Fuhr. A Probabilistic Framework for Vague Queries and Imprecise Information in Databases. In *Proc. of VLDB*, 1990.
25. N. Fuhr and T. Rölleke. A Probabilistic NF2 Relational Algebra for Imprecision in Databases. *Unpublished Manuscript*, 1997.
26. G. Grahne. Dependency Satisfaction in Databases with Incomplete Information. In *Proc. of VLDB*, 1984.
27. G. Greco, S. Greco, and E. Zumpano. A logical framework for querying and repairing inconsistent databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(6).
28. T. J. Green and V. Tannen. Models for incomplete and probabilistic information. In *Proc. of IIDB Workshop*, 2006.
29. Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: an efficient algorithm for discovering functional and approximate dependencies. *Comp. Journal*, 42(2), 1999.
30. T. Imielinski and W. Lipski. Incomplete information and dependencies in relational databases. In *Proc. of ACM SIGMOD*, 1983.
31. T. Imielinski and W. Lipski. Incomplete Information in Relational Databases. *Journal of the ACM*, 31(4), 1984.
32. L. V. S. Lakshmanan, N. Leone, R. Ross, and V.S. Subrahmanian. ProbView: A Flexible Probabilistic Database System. *ACM TODS*, 22(3), 1997.
33. S. K. Lee. An extended Relational Database Model for Uncertain and Imprecise Information. In *Proc. of VLDB*, 1992.
34. M. Levene and G. Loizou. Axiomatisation of functional dependencies in incomplete relations. *Theoretical Computer Science*, 206, 1998.
35. E. Lien. Multivalued dependencies with null values in relational databases. In *Proc. of VLDB*, 1979.
36. D. Maier. *Theory of Relational Databases*. Computer Science Pr, 1983.
37. V. Th. Paschos. Polynomial approximation and graph-coloring. *Computing*, 70(1), 2003.
38. C. Re and D. Suciu. Materialized views in probabilistic databases for information exchange and query optimization. In *Proc. of VLDB*, 2007.
39. F. Sadri. Reliability of answers to queries in relational databases. *TKDE*, 3(2):245–251, 1991.
40. F. Sadri. Integrity constraints in the information source tracking method. *TKDE*, 7(1):106–119, 1995.
41. P. Sen and A. Deshpande. Representing and Querying Correlated Tuples in Probabilistic Databases. In *Proc. of ICDE*, 2007.
42. J. D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press, 1988.
43. Y. Vassiliou. Functional dependencies and incomplete information. In *Proc. of VLDB*, 1981.
44. J. Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *Proc. of CIDR*, 2005.
45. J. Wijsen. Condensed representation of database repairs for consistent query answering. In *Proc. of ICDT*, 2003.