CDMTCS
Research
Report
Series

Normal Forms and Normalization for Probabilistic Databases under Sharp Constraints

Sven Hartmann
Clausthal University of Technology
Clausthal-Zellerfeld, Germany

Sebastian Link
University of Auckland,
Auckland, New Zealand

CDMTCS-454
February 2014

Centre for Discrete Mathematics and
Theoretical Computer Science

Normal Forms and Normalization for Probabilistic Databases under Sharp Constraints

SVEN HARTMANN
Clausthal University of Technology, Germany
sven.hartmann@tu-clausthal.de

SEBASTIAN LINK
The University of Auckland, Private Bag 92019, New Zealand
s.link@auckland.ac.nz

February 13, 2014

Abstract

The data deluge is defined by increasing amounts of large data with increasing degree of uncertainty. In a recent response, probabilistic databases are receiving a great deal of interest from research and industry. One popular approach to probabilistic databases is to extend traditional relational database technology to handle uncertainty. In this approach probabilistic databases are probability distributions over a collection of possible worlds of relational databases. On the one hand, research has seen various efforts to extend query evaluation from relational to probabilistic databases. On the other hand, updates have not received much attention at all. In this paper we show that well-known syntactic normal form conditions capture probabilistic databases with desirable update behavior. Such behavior includes the absence of data redundancy, insertion, deletion, and modification anomalies. We further show that standard normalization procedures can be applied to standard representations of probabilistic databases to obtain database schemata that satisfy the normal form condition, and can thus be updated efficiently.

Keywords: Data redundancy, Normal form, Probabilistic data, Schema design, Uncertainty, Update anomaly

1 Introduction

Major challenges such as climate change, finding new sources of energy, curing diseases and overcoming poverty engage thousands of people globally. In addressing these challenges we have created a data deluge that makes it necessary to manage increasingly large sets of increasingly uncertain data [57]. There are different roots of uncertainty ranging from measurement errors in sensor data, ambiguity in natural language processing for information extraction, tolerating uncertainty to reduce the cost of data cleaning in business intelligence, or even introducing uncertainty to guarantee a higher level of data privacy [17]. Traditional database technology was designed for applications with certainty in data, including payroll and inventory. However, the desire of users to get more out of large volumes of uncertain data highlights the need for generic tools to manage uncertain data.

Probabilistic databases aim to extend standard relational database technology to handle uncertain data. In contrast to relational databases where a tuple is either present or not, tuples in probabilistic databases are present with some probability. One popular approach is to define probabilistic databases as probability spaces over a collection of possible worlds, each of which is a relational database. That is, the exact state of the database is given by a probability distribution. The following example illustrates these concepts on the running example we will utilize in this paper.

Example 1 *As a simple running example we consider probabilistic database over a single relation schema where suppliers deliver articles from a location at a cost.*

World w_1 *with* $P(w_1) = 0.3$

article	*supplier*	*location*	*cost*
mypad	*pear*	*wuhan*	*780*
myphone	*pear*	*phuket*	*350*

World w_2 *with* $P(w_2) = 0.7$

article	*supplier*	*location*	*cost*
urpad	*mango*	*busan*	*760*
urpad	*mango*	*tokyo*	*760*
urphone	*mango*	*tokyo*	*375*
urphone	*mango*	*busan*	*375*

Each possible world in the probabilistic database satisfies the integrity constraints that capture important business rules that hold in this domain. For example, each world satisfies the functional dependencies that a supplier is uniquely determined by an article; and that the cost is uniquely determined by an article and the location it is supplied from. Furthermore, the worlds satisfy the multivalued dependency that the set of locations a supplier supplies from is uniquely determined by the supplier, independently of the article and the cost of its supply.

As probabilistic databases are probability spaces over worlds of relational databases their schema designs can be affected by poor performance behavior under queries and updates. While the processing of queries has been the subject of recent research endeavors [3, 18, 55, 57], updates have not yet received much attention at all. One classic lesson learned from relational databases is that the query and update load of a database must be considered together. In particular, efficient query evaluation benefits from high levels of data redundancy while efficient update processing benefits from low levels of data redundancy [1]. It is no surprise that probabilistic databases follow the same trade-off pattern. In Example 1, for instance, world w_2 suffers from several occurrences of redundant data values, e.g., any occurrence of *banana* or *375* can already be inferred from the remaining data values and the business rules. Data value redundancy is commonly an indicator for inefficiencies with updates.

Contribution. For these reasons it is desirable to have available a normalization theory for probabilistic databases, similar to that for relational databases [1, 12, 22, 63]. There are different approaches to normalize probabilistic databases, but in this paper we will focus on the popular approach of defining probabilistic databases as probability spaces over worlds of relational databases. Our contributions are as follows:

- We provide a simple, general and mathematically precise semantics for including integrity constraints in the definition of probabilistic databases. The definition can be used in future research to investigate several classes of integrity constraints, and therefore incorporate more semantics into probabilistic databases.
- We propose several semantic normal form conditions for probabilistic databases. These include a normal form that eliminates local data redundancy in terms of the expressive combined class of functional and multivalued dependencies from any worlds in a probabilistic database. It further includes normal forms that eliminate update anomalies in terms of insertions, deletions, and modifications in probabilistic databases.
- We show that the Fourth Normal Form, well-known from relational databases [22, 63], is equivalent to most of these semantic normal forms for probabilistic databases. Fourth Normal Form reduces to Boyce-Codd-Heath normal form [16, 39] when the given sets of constraints are functional dependencies only. Hence, these normal forms guarantee the same efficient update behavior for probabilistic databases as they do for relational databases [63]; when probabilistic databases are naturally defined as probability spaces over worlds of relational databases.
- Finally, we propose to normalize database schemata for probabilistic databases by applying normalization techniques to standard representations of the probabilistic databases. Recent research has shown that some of the representations of probabilistic databases guarantee that uncertain information can be queried efficiently by standard relational technology. Our approach to normalization further shows that uncertain information can also be updated efficiently by standard relational technology. This is achieved by making available standard relational normalization theory to standard representations of probabilistic databases.

Organization. We summarize related work in Section 2. We define the model of probabilistic databases in Section 3, including integrity constraints used to capture the semantics of application domains. Here, we also summarize previous relevant findings on the class of functional and multivalued dependencies. In Section 4 we define a semantic normal form that guarantees the absence of any redundant data value occurrences in any world of any probabilistic database. We show that the Fourth Normal Form characterizes this semantic normal form syntactically, for the case of functional and multivalued dependencies. For functional dependencies alone, the Boyce-Codd-Heath Normal Form achieves this. We propose several semantic normal forms regarding the absence of any insertion, deletion, and different types of modification anomalies from updates on any worlds of any probabilistic database in Section 5. A syntactic characterization is established for each of these semantic normal forms, which in most cases equates to the Fourth Normal Form in the general case, and to Boyce-Codd-Heath Normal Form in the case of just functional dependencies. In Section 6 we propose to normalize probabilistic databases by normalizing their standard representations, for example, in the form of BID databases. We conclude in Section 7 where we also comment on future work.

2 Related Work

Probabilistic database have become a hot topic due to the need to handle uncertain data in many applications. Any probabilistic database can be represented in form of either PC-tables, tuple-independent databases together with relational algebra views, block-independent-disjoint databases together with conjunctive query views, or U-databases [57]. Querying techniques range from extensional to intensional techniques. In extensional query evaluation, the probability of a tuple to belong to a query answer can be processed efficiently by an SQL engine; but not all queries can be processed correctly this way. In intensional query evaluation, any query can be processed, but the data complexity of a query can be $\sharp P$-hard [57]. Different prototypes of probabilistic databases exist, including Mystiq [18], Trio [55], and MayBMS [3].

Handling uncertain data by extending relational technology has several advantages, in particular the mature technology and the trust from its user base. The relational database industry is worth an estimated 32 billion US dollars [52]. After almost 40 years in use, relational database systems still dominate the market today and influence new paradigms [2]. Web models are applied primarily to roll-out, exchange and integrate data that are commonly relational [53]. Many websites, e.g. Facebook, and distributed applications, e.g. e-commerce, require high scalability, but their core data stores and services remain relational [53].

Relational normalization theory is rich and deep. The present paper shows how probabilistic databases can apply this theory. Functional dependencies (FDs) were already proposed by Codd [14], and Delobel, Fagin, and Zaniolo independently introduced multivalued dependencies (MVDs) [22]. The implication problem of FDs and MVDs is finitely axiomatisable [10], can be decided in almost linear time [25] and enjoys a strong correspondence to logic [54]. These results have recently been extended to SQL [35]. Third Normal Form (3NF) [12, 45], Boyce-Codd-Heath Normal Form (BCHNF) [16, 39], and Fourth Normal Form (4NF) [22, 66] are standard teaching material. Vincent demonstrated what these syntactically defined normal forms actually achieve on the semantic level [63], in terms of the absence of data redundancy and update anomalies. Note that data redundancy may still occur, e.g. in terms of other data dependencies such as join, embedded multivalued or inter-relational dependencies [46].

Work on normalization in probabilistic databases is rather limited. Dalvi, Ré and Suciu note that Şto date there exists no comprehensive theory of normalization for probabilistic databases" [17]. Noticeably, two other papers have studied normalization in the context of probabilistic databases. Dey and Sarkar [20] introduce *stochastic dependencies* as generalizations of FDs to model the dependency between the probability distribution of attributes. The work is not founded on the possible world semantics. Finally, Das Sarma, Ullman and Widom study various classes of FDs over uncertain relations [56]. While they do found their work on a possible world semantics, the possible worlds originate from alternatives of tuples, which is their main construct for uncertainty. Probabilistic databases, normal forms and their semantic justification are out of that work's scope. This, however, is the focus of this paper.

3 Probabilistic Databases and Data Dependencies

We give the main definitions for the data model, and introduce integrity constraints as first-class citizens of probabilistic databases. Results on FDs and MVDs are summarized.

3.1 Relational databases

First we fix standard relational database terminology [1].

As usual, we assume that there is a countably infinite set $\mathfrak{A}$ of symbols, whose elements we call *attributes*. Each attribute $A \in \mathfrak{A}$ has an at most countable set $dom(A)$ as its *domain*, i.e., the set of possible values associated with an attribute.

Let R denote some finite, non-empty set of attributes from $\mathfrak{A}$. A tuple over R is a function $t : R \to \cup_{A \in R} dom(A)$ such that for all $A \in R$, $t(A) \in dom(A)$ holds. For some $X \subseteq R$ we write $t(X)$ to denote the *projection* of t onto X. An *R-local integrity constraint* over R is a function i that maps a finite set r of tuples over R to $\{0, 1\}$. If $i(r) = 1$, we say that r satisfies i. Popular classes of local integrity constraints are keys, functional dependencies, and multivalued dependencies; which we define later on.

A *relation schema* is a pair (R, Σ_R) where R is a finite, non-empty set of attributes from $\mathfrak{A}$, and Σ_R is a set of R-local integrity constraints. A *relation* over (R, Σ_R) is a finite set of tuples over R that satisfies all elements of Σ_R.

Let $D = \{(R_1, \Sigma_1), \ldots, (R_k, \Sigma_k)\}$ denote a finite set of relation schemata. A *D-global integrity constraint* over D is a function i that maps a finite set $d = \{r_1, \ldots, r_k\}$ of relations r_i over relation schema (R_i, Σ_i), $i = 1, \ldots, k$, to $\{0, 1\}$. If $i(d) = 1$, we say that d satisfies i. Popular classes of global integrity constraints are foreign keys, inclusion dependencies, and cardinality constraints. They are outside the scope of this paper, but model important application semantics.

A *database schema* is a pair (D, Σ) where $D = \{(R_1, \Sigma_1), \ldots, (R_k, \Sigma_k)\}$ is a finite, non-empty set of relation schemata, and Σ is a set of D-global integrity constraints. A *relational database* over (D, Σ) is a set $d = \{r_1, \ldots, r_k\}$ that satisfies every $\sigma \in \Sigma$, and where for $i = 1, \ldots, k$, r_i is a relation over (R_i, Σ_i).

3.2 Probabilistic databases

In recent popular approaches to probabilistic databases, uncertainty is modeled by allowing different relational databases to co-exist [57]. Each of them represents a possible world, comes associated with a weight between 0 and 1, and the weights sum up to 1. In a subjectivist Bayesian interpretation, one of the possible worlds represents the ŞtrueŤ relational database. However, we are uncertain about the true world, and the probabilities represent degrees of belief in the various possible worlds. This model can be formalized by the following definition.

Definition 1 *A probabilistic database over a database schema (D, Σ) is a probability space (W, P) over the finite set W of relational databases over (D, Σ). That is, $P : W \to (0, 1]$ such that $\sum_{w \in W} P(w) = 1$. Each element of W is called a possible world of W.* ▮

Example 2 *Recall Example 1 where (D, Σ) consists of the single relation schema*

$$R = \{article, supplier, location, cost\}$$

and the set Σ_R of consists of the three business rules mentioned. The set Σ of D-global integrity constraints is empty. Example 1 also shows the set W of two possible worlds w_1 and w_2, each of which satisfies all of the business rules, and are thus relations over (R, Σ_R). The probabilities

$P(w_1) = 0.3$ *and* $P(w_2) = 0.7$ *sum up to 1, and* (W, P) *is therefore a probability space over* W. *That is,* (W, P) *is a probabilistic database.*

Definition 1 is purposefully more general than we require here. For the remainder of this article the set Σ of global constraints will be empty, as is the case in the running example. That means we will not be concerned with constraints between different relation schemata, including referential integrity constraints. Our more general definition has the purpose to encourage further research into this subject, and promote integrity constraints as first-class citizens of probabilistic databases, similar to their role in relational databases [1].

Definition 1 is a simple, intuitive and natural definition of a probabilistic database. It meets the requirements of the discussion from the beginning of this section. As part of this definition of a probabilistic database (W, P), every relation r over some relation schema (R, Σ_R) that occurs in some possible world $w \in W$ satisfies all constraints in Σ_R. This is a natural way to model integrity constraints, whose purpose is to constrain instances to those considered meaningful for the application at hand. For probabilistic databases, specifically, this means that a world is considered possible only if it satisfies the integrity constraints. For the remainder of this article our attention will focus on the expressive combined class of FDs and MVDs. In Section 7 we briefly discuss other approaches towards integrity constraints in probabilistic databases.

3.3 Keys, Functional and Multivalued Dependencies

Keys, functional and multivalued dependencies play a fundamental role in database design and facilitate many data processing tasks. Literature on these dependencies in the relational model include [6, 8, 9, 10, 11, 12, 15, 19, 21, 23, 25, 31, 41, 47, 48, 49, 50, 51, 54, 59, 63], in conceptual models [44, 61, 62, 65], in models that incorporate incomplete information [7, 28, 29, 35, 42, 43, 58], in nested data models [24, 26, 36, 30, 37, 32, 60], and more recently in XML [4, 5, 13, 27, 33, 34, 38, 40, 64].

Let R denote a finite set of attributes. A *functional dependency* (FD) over R is a statement $X \to Y$ where $X, Y \subseteq R$. The FD $X \to Y$ over R is satisfied by a finite set r of tuples over R if and only if for all $t_1, t_2 \in r$ the following holds: if $t_1(X) = t_2(X)$, then $t_1(Y) = t_2(Y)$. We call $X \to Y$ *trivial* whenever $Y \subseteq X$, and non-trivial otherwise.

A *multivalued dependency* (MVD) over R is a statement $X \twoheadrightarrow Y$ where $X, Y \subseteq R$. The MVD $X \twoheadrightarrow Y$ over R is satisfied by a finite set r of tuples over R if and only if for all $t_1, t_2 \in r$ the following holds: if $t_1(X) = t_2(X)$, then there is some $t \in r$ such that $t(XY) = t_1(XY)$ and $t(X(R - Y)) = t_2(X(R - Y))$. We call $X \twoheadrightarrow Y$ *trivial* whenever $Y \subseteq X$ or $XY = R$, and non-trivial otherwise.

For a set Σ of R-local integrity constraints, we say that a finite set r of tuples over R *satisfies* Σ if r satisfies every $\sigma \in \Sigma$.

Constraints interact with one another. Let $\Sigma \cup \{\varphi\}$ be a set of FDs and MVDs over R. We say that Σ *implies* φ if every finite set t of tuples over R that satisfies Σ also satisfies φ. For Σ we let $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ be the *semantic closure* of Σ, i.e., the set of all FDs and MVDs implied by Σ. In order to determine the logical consequences we use a syntactic approach by applying inference rules, e.g. those in Table 1. We let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by the set $\mathfrak{R}$ of inference rules [1]. We let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be the *syntactic closure* under

Table 1: Axiomatization $\mathfrak{S}$ of FDs and MVDs

$\dfrac{}{XY \to Y}$ (reflexivity, $\mathcal{R}_{\mathrm{F}}$)	$\dfrac{X \to Y}{X \to XY}$ (extension, $\mathcal{E}_{\mathrm{F}}$)	$\dfrac{X \to Y \quad Y \to Z}{X \to Z}$ (transitivity, $\mathcal{T}_{\mathrm{F}}$)
$\dfrac{}{\emptyset \twoheadrightarrow R}$ (R-axiom, $\mathcal{C}^{R}_{\mathrm{M}}$)	$\dfrac{X \twoheadrightarrow Y \quad X \twoheadrightarrow Z}{X \twoheadrightarrow YZ}$ (MVD union, $\mathcal{U}_{\mathrm{M}}$)	$\dfrac{X \twoheadrightarrow Y \quad Y \twoheadrightarrow Z}{X \twoheadrightarrow Z - Y}$ (pseudo-transitivity, $\mathcal{T}_{\mathrm{M}}$)
	$\dfrac{X \to Y}{X \twoheadrightarrow Y}$ (MVD implication, $\mathcal{I}_{\mathrm{FM}}$)	$\dfrac{X \twoheadrightarrow Y \quad Y \to Z}{X \to Z - Y}$ (mixed pseudo-transitivity, $\mathcal{T}_{\mathrm{FM}}$)

inferences by $\mathfrak{R}$. A set $\mathfrak{R}$ of inference rules is said to be *sound* (*complete*) for the implication of FDs and MVDs if for every finite set R of attributes, and for every set Σ of FDs and MVDs over R we have $\Sigma^{+}_{\mathfrak{R}} \subseteq \Sigma^{*}$ ($\Sigma^{*} \subseteq \Sigma^{+}_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* for the implication of FDs and MVDs if $\mathfrak{R}$ is both sound and complete. The *implication problem* for FDs and MVDs is to decide, given some finite set R of attributes and some set $\Sigma \cup \{\varphi\}$ of FDs and MVDs over R, whether Σ implies φ. Several finite axiomatizations exist for the implication of FDs and MVDs [10], Table 1 shows that from [31]. When we define the syntactic normal forms later we will make reference to the set $\mathfrak{S}$ of inference rules from Table 1, but, in fact, any finite axiomatization will have the same effect.

For a set Σ of FDs and MVDs over R, let Σ_k denote the set of keys of R with respect to Σ. An FD $X \to R \in \Sigma^{+}_{\mathfrak{S}}$ is called a *superkey* of R with respect to Σ. A superkey $X \to R \in \Sigma^{+}_{\mathfrak{S}}$ is called a *key* of R with respect to Σ, if there is no superkey $X' \to R \in \Sigma^{+}_{\mathfrak{S}}$ of R with respect to Σ where $X' \subset X$.

Example 3 *Consider the relation schema $R = \{article, supplier, location, cost\}$ from Example 1. The business rules from that example can be formalized as FDs and MVDs over R. Indeed, the FD article $\to$ supplier says that every article has at most one supplier, the FD article,location $\to$ cost says that the cost of an article is determined by the article and the location the article is supplied from. Finally, the MVD supplier $\twoheadrightarrow$ location says that the set of locations where a supplier supplies from is determined by the supplier, independently of the article and cost. The set Σ_R, consisting of these two FDs and the MVD, imply other FDs and MVDs. For example, the FD article $\to$ cost or the MVD article $\twoheadrightarrow$ location. The only key of R with respect to Σ_R is article,location $\to$ R.*

For the remainder of this article we assume that all local constraints are functional and multivalued dependencies.

4 Data Value Redundancy in Probabilistic Databases

In this section we will propose the Probabilistic Redundancy-Free Normal Form that captures database schemata for which no probabilistic database exists that features any redundant data value occurrence in any possible world. From the point of view of updates and consistency, this is highly desirable since there is no need to ever consistently update any redundant data values - consistency enforcement would be cheap. We will then show that Fagin's 4NF proposal, specifically designed for relational databases, is equivalent to Probabilistic Redundancy-Free Normal Form. Since 4NF is a syntactic normal form and can be checked efficiently, it is a highly desirable normal form for probabilistic databases, too.

4.1 Probabilistic Redundancy-Free Normal Form

Before the definition of this semantic normal form, we need to define explicitly what a redundant data value occurrence constitutes. For this, we follow the proposal by Vincent [63]. Let (R, Σ) denote a relation schema, A an attribute of R, and t a tuple over R. A *replacement* of $t(A)$ is a tuple $\bar{t}$ over R that satisfies the following conditions: i) for all $\bar{A} \in R - \{A\}$ we have $\bar{t}(\bar{A}) = t(\bar{A})$, and ii) $\bar{t}(A) \neq t(A)$. Intuitively, a data value occurrence in some possible world is redundant if the occurrence cannot be replaced by any other data value without violating some constraint in Σ.

Definition 2 *Let D be a database schema, (R, Σ) a relation schema over D, $A \in R$ an attribute, r a relation over R, and t a tuple in r. We say that the data value occurrence $t(A)$ is* redundant *if and only if for* every *replacement $\bar{t}$ of $t(A)$, $\bar{r} := (r - \{t\}) \cup \{\bar{t}\}$ is not a relation over (R, Σ).*

Example 4 *Consider the worlds w_1 and w_2 from Example 1. Both satisfy the set Σ_R of FDs and MVDs from Example 3. However, in w_1 no data value occurrence is redundant. That is, each data value can be replaced by some other data value without violating Σ_R. In w_2 there are several redundant data value occurrences. For example, any occurrence of* mango *cannot be replaced by a different value since it would result in a set of tuples that does not satisfy Σ_R and would thus not be a relation over (R, Σ_R).*

Given this definition of redundant data value occurrence, a database schema is now said to be in Probabilistic Redundancy-Free Normal Form if there cannot be any probabilistic database over D that features a possible world with some redundant data value occurrence in it.

Definition 3 *We say that D is in* Probabilistic Redundancy-Free Normal Form *(PRFNF) if and only if there is no probabilistic database (W, P) over D such that there is some possible world $w \in W$ with some relation r in w over some relation schema (R, Σ) in D, some attribute $A \in R$, and some tuple $t \in r$ where the data value occurrence $t(A)$ is redundant.* ▮

Example 5 *Clearly, the database schema D consisting of (R, Σ_R) from Example 3 is not in PRFNF. The probabilistic database (W, P) from Example 1 is a probabilistic database over D which features the possible world w_2 over (R, Σ_R) in which a redundant data value occurs.*

The next question is how to recognize database schemata in PRFNF, without having to look at any probabilistic databases over this schema. In other words, we would like to have a syntactic characterization of PRFNF.

4.2 Syntactic Characterization of PRFNF

Fagin [22] introduced the Fourth Normal Form condition on relation schemata that characterizes the absence of redundant data value occurrences caused by FDs and MVDs [63]. Let D denote a database schema. Then D is said to be in *Fourth Normal Form* (4NF) if and only if for all relation schemata (R, Σ) in D and for all non-trivial multivalued dependencies $X \twoheadrightarrow Y \in \Sigma^+_{\mathfrak{S}}$ we have $X \to R \in \Sigma^+_{\mathfrak{S}}$. Using the axiomatization $\mathfrak{S}$ this normal form is purely syntactic, since it does not make any reference to any (probabilistic) database. The 4NF proposal is also cover-insensitive, i.e., for any relation schema (R, Σ) we can replace Σ by an equivalent set of FDs and MVDs, without affecting the property of D to be in 4NF. At first sight it appears that it might require time exponential in the size of the given constraints to check whether a given database schema is in 4NF. However, it suffices to check the constraints given in Σ instead of checking all the constraints in the syntactic closure $\Sigma^+_{\mathfrak{S}}$. Indeed, D is in 4NF if and only if for all relation schemata (R, Σ) in D, for all non-trivial FDs $X \to Y \in \Sigma$ and for all non-trivial MVDs $X \twoheadrightarrow Y \in \Sigma$, $X \to R \in \Sigma^+_{\mathfrak{S}}$. Indeed, 4NF characterizes database schemata in Probabilistic Redundancy-Free Normal Form.

Theorem 1 *D is in Fourth Normal Form if and only if D is in PRFNF.*

Example 6 *Continuing our running example, D from Example 3 is not in 4NF. For example, the FD article $\to$ supplier $\in \Sigma$, but article $\to$ location $\notin \Sigma^+_{\mathfrak{S}}$. Indeed, we had already confirmed that D is also not in PRFNF.*

For the special case where all constraints are functional dependencies, Boyce-Codd-Heath Normal Form (BCHNF) characterizes PRFNF. Recall that D is in BCHNF [16, 39] if and only if for all relation schemata (R, Σ) in D, for all non-trivial FDs $X \to Y \in \Sigma$, $X \to R \in \Sigma^+_{\mathfrak{S}}$.

Corollary 1 *Suppose that D is a database schema where all local constraints are functional dependencies. Then D is in Boyce-Codd-Heath normal form if and only if D is in PRFNF.* ▮

5 Update Anomalies in Probabilistic Databases

In this section we will propose several semantic normal forms that guarantee the absence of various update anomalies in any probabilistic databases. In essence, an update anomaly occurs whenever it does not suffice to show that all minimal keys are still satisfied after an update. The absence of such anomalies is desirable since checking key constraints is cheap, but checking FDs and MVDs is expensive. Again, we show that several of these semantic normal forms are equivalent to Fagin's 4NF proposal. For the remaining cases, we establish other syntactic characterizations.

5.1 Insertion Anomalies

Insertion anomaly normal form requires for the insertion of any tuple in any possible world of any probabilistic database that the resulting set of tuples is a relation whenever it is a relation with respect to all keys. Hence, it suffices to check that all keys are satisfied to permit an insertion.

Definition 4 *D is said to be in* probabilistic key-based insertion anomaly normal form *(PKINF) if and only if there is no probabilistic database (W, P) over D such that there is some world $w \in W$ and some relation r in w over relation schema (R, Σ) in D, and some R-tuple $t \notin r$ where $r \cup \{t\}$ satisfies Σ_k, but $r \cup \{t\}$ is not a relation over (R, Σ).*

Example 7 *The database schema D from Example 1 is not in PKINF. Indeed, an insertion of the tuple $(\mathit{article} : \mathit{myphone}, \mathit{supplier} : \mathit{pear}, \mathit{location} : \mathit{wuhan}, \mathit{cost} : \mathit{400})$ into w_1 would result in a set of tuples that satisfies Σ_k, i.e. the key article,location $\rightarrow R$ of R with respect to Σ_R, but it would violate the MVD supplier $\twoheadrightarrow$ location.*

It turns out that 4NF characterizes database schemata in PKINF.

Theorem 2 *D is in 4NF if and only if D is in PKINF.*

5.2 Deletion Anomalies

Deletion anomaly normal form abandons the deletion of tuples from relations of any possible world whenever the resulting set of tuples is a relation with respect to all keys, but not a relation with respect to the set of constraints.

Definition 5 *D is said to be in* probabilistic key-based deletion anomaly normal form *(PKDNF) if and only if there is no probabilistic database (W, P) over D such that there is some world $w \in W$ and some relation r in w over relation schema (R, Σ) in D, and some R-tuple $t \in r$ where $r - \{t\}$ satisfies Σ_k, but $r - \{t\}$ is not a relation over (R, Σ).*

Example 8 *The database schema D from Example 1 is not in PKDNF. Indeed, a deletion of the tuple $(\mathit{article} : \mathit{urphone}, \mathit{supplier} : \mathit{mango}, \mathit{location} : \mathit{busan}, \mathit{cost} : \mathit{375})$ from w_2 would result in a set of tuples that satisfies Σ_k, i.e. the key article,location $\rightarrow R$ of R with respect to Σ_R, but it would violate the MVD supplier $\twoheadrightarrow$ location.*

If a database schema is in 4NF, then it is also in PKDNF. However, there are schemata not in 4NF which are still in PKDNF. Such database schemata have only relation schemata whose set of FDs and MVDs is necessarily equivalent to a set of FDs only.

Theorem 3 *D is in PKDNF if and only if for every relation schema (R, Σ) in D, Σ is equivalent to a set of functional dependencies.*

5.3 Modification Anomalies

Modification anomaly normal forms abandon the modification of tuples from relations in any possible world whenever the resulting set of tuples maintains key uniqueness, but is not a relation with respect to the set of constraints. In practice, it is often desirable to maintain the identity of a tuple during modification. Since, in general, there can be multiple keys, there are several possible interpretations of maintaining the identity of a tuple: values on some key are maintained, values on the primary key are maintained, or values on every key are maintained. These interpretations result therefore in four different normal form proposals.

Definition 6 *D is said to be in* probabilistic key-based modification normal form of type 1, type 2, type 3, type 4, *respectively, (PKMNF$_i$ for $i = 1, \ldots, 4$) if and only if there is no probabilistic database (W, P) over D such that there is some world $w \in W$ and some relation r in w over relation schema (R, Σ) in D, and some R-tuples $t \in r$ and $t' \notin r$*

- *with $t(K) = t'(K)$ for some key K with respect to Σ (for type 2),*
- *with $t(K) = t'(K)$ for the primary key K with respect to Σ (for type 3),*
- *with $t(K) = t'(K)$ for all keys K with respect to Σ (for type 4), respectively,*

where $(r - \{t\}) \cup \{t'\}$ satisfies Σ_k, but $(r - \{t\}) \cup \{t'\}$ is not a relation over (R, Σ).

Example 9 *The database schema D from Example 1 is not in $PKMNF_i$ for any $i \in \{1, \ldots, 4\}$. Indeed, a modification of the tuple $(article : urphone, supplier : mango, location : busan, cost : 375)$ from w_2 to $(article : urphone, supplier : pear, location : busan, cost : 375)$ would result in a set of tuples that satisfies Σ_k, i.e. the key article,location $\to R$ of R with respect to Σ_R, but it would violate the MVD supplier $\twoheadrightarrow$ location. Note that the modified tuple matches the replaced tuple on all keys of R with respect to Σ_R (there is only the key article,location $\to R$).*

It turns out that the first three normal forms for modification anomalies can be characterized by BCHNF in the case of FDs only, and by 4NF in the general case (assuming that some non-trivial FD occurs).

Theorem 4 *Let D be a database schema in which every relation schema (R, Σ) has a set Σ of functional dependencies only. Then BCHNF, PKMNF$_1$, PKMNF$_2$, and PKMNF$_3$ are equivalent.*

Theorem 5 *Let D be a database schema in which every relation schema (R, Σ) is such that Σ contains some non-trivial FD. Then 4NF, PKMNF$_1$, PKMNF$_2$, and PKMNF$_3$ are equivalent.*

The remaining normal form, PKMNF$_4$, can be characterized by a different normal form. For that, we require a few more definitions. We call a set Σ of FDs and MVDs reduced, if there is no dependency $\sigma \in \Sigma$ such that $\Sigma - \{\sigma\} \models \sigma$, and for every MVD $X \twoheadrightarrow Y \in \Sigma$ (FD $X \to Y \in \Sigma$) there is no MVD $X' \twoheadrightarrow Y' \in \Sigma^+_{\mathfrak{G}}$ (FD $X' \to Y' \in \Sigma^+_{\mathfrak{G}}$) where $X' \subset X$ or $\emptyset \subset Y' \subseteq Y$ holds. An attribute $A \in R$ is said to be *prime* with respect to Σ if $A \in X$ for some key $X \to R$ of R with respect to Σ. Let Σ denote a reduced set of FDs. Then (R, Σ) is in *prime attribute normal form* (PANF), if for every FD $X \to A \in \Sigma$, either X is a key of R with respect to Σ, or every attribute of XA is prime with respect to Σ. D is in PANF, if every relation schema (R, Σ) of D is in PANF.

Theorem 6 *Let D be a database schema in which every relation schema (R, Σ) has a set Σ of functional dependencies only. Then D is in PKMNF$_4$ if and only if D is in PANF.*

An MVD $X \twoheadrightarrow Y$ in a set Σ of FDs and MVDs over R is said to be *pure* if it is non-trivial, $X \to Y \notin \Sigma^+_{\mathfrak{G}}$ and $X \to R - XY \notin \Sigma^+_{\mathfrak{G}}$.

Theorem 7 *Let D be a database schema in which every relation schema (R, Σ) has a set Σ that contains some pure MVD. Then D is in PKMNF$_4$ if and only if for every relation schema (R, Σ) in D, every attribute of R is prime with respect to Σ.*

6 Normalization

The goal of this section is to address how the syntactic normal forms can actually be achieved, in order to guarantee efficient processing of updates on probabilistic databases.

Definition 1 does not suggest a practical representation of probabilistic data. When the number of possible worlds is very large, it becomes infeasible to enumerate all possible worlds explicitly. Recent research on probabilistic databases has established several representation systems that provide means to represent any probabilistic database concisely [57]. These representation systems are used to evaluate queries on probabilistic databases efficiently. It is therefore our simple proposition to apply standard relational normalization techniques to the relational part of the representations of probabilistic databases. We describe this approach on *block-independent-disjoint* databases, or *BID* databases for short.

A BID database, is a probabilistic database where tuples are partitioned into blocks, such that all tuples in a block are mutually-exclusive probabilistic events, and all tuples from different blocks are independent probabilistic events. BID databases are a complete representation formalism of probabilistic databases when coupled with views expressed as conjunctive queries [57]. Indeed, one can add to each set R of attributes a new attribute K which represents unique identifiers for each possible world. Tuples from different possible worlds can thus be stored in a single relation, and distinguished by their possible world identifier. The possible worlds are stored over a BID schema with the singleton attribute K representing each possible world by an identifier k. The original probability distribution P where $P(w) = p_i$ is simply represented by setting $P(k) := p_k$. The original possible worlds can then easily be recovered by a conjunctive query view definition over the BID representation.

The crucial observation is that this BID representation also preserves the semantics given by the integrity constraints. Indeed, $w_1, \ldots, w_n$ are possible worlds of (R, Σ) if and only if $\bigcup_{i=1}^{n}\{i\} \times w_i$ is a relation over $(\{K\} \cup R, K\Sigma)$ where $K\Sigma = \{KX \to Y \mid X \to Y \in \Sigma\} \cup \{KX \twoheadrightarrow Y \mid X \twoheadrightarrow Y \in \Sigma\}$.

We therefore propose to normalize a probabilistic database by first representing the probabilistic database as a BID database, and then apply standard relational normalization techniques to the relation schemata $(\{K\} \cup R, K\Sigma)$ of the representation. These techniques may include BCHNF- and 4NF-decomposition, or 3NF synthesis [1, 12, 22]. We illustrate our proposal by a 4NF-decomposition of the running example.

Example 10 *The following database is a BID representation of the probabilistic database from Example 1. The possible worlds are represented by unique values on the extra attribute K, i.e. world w_1 by value 1 and world w_2 by value 2, and the world table features the probability distribution.*

Article A

K	*article*	*supplier*	*cost*
1	*mypad*	*pear*	*780*
1	*myphone*	*pear*	*350*
2	*urpad*	*mango*	*760*
2	*urphone*	*mango*	*375*

Location L

K	*article*	*location*
1	*mypad*	*wuhan*
1	*myphone*	*phuket*
2	*urpad*	*tokyo*
2	*urpad*	*busan*
2	*urphone*	*tokyo*
2	*urphone*	*busan*

World Table W

K	*P*
1	*0.3*
2	*0.7*

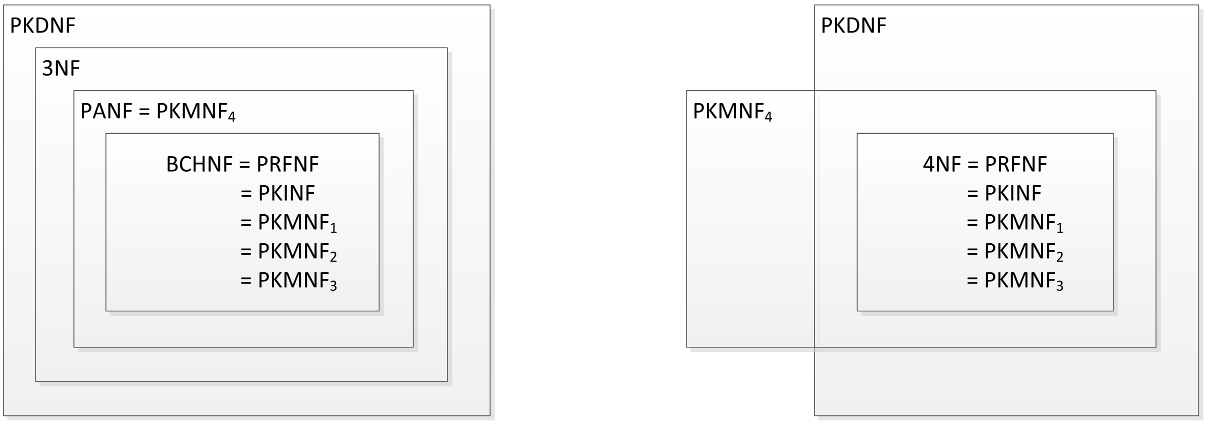

Figure 1: Achievements of BCHNF and 4NF for Probabilistic Databases

This relational database, i.e. the part without the world table, is in 4NF. In particular, it does not feature any redundant data value occurrences. The original database D is the conjunctive query view defined by:

$$D(x_a, x_s, x_c, x_l) : -A(k, x_a, x_s, x_c), L(k, x_a, x_l), W(k) \quad .$$

Note that the decomposition into $A(k, x_a, x_s, x_c)$ and $L(k, x_a, x_l)$ is a result of the functional dependency that the cost and supplier are functionally determined by the article and K, an FD implied by the constraints in Example 1 and the representation as a BID database.

7 Conclusion and Future Work

Probabilistic databases aim to manage efficiently large amounts of uncertain data. A popular approach is to define probabilistic databases as probability spaces over collections of possible worlds that are relational databases. Recent research has demonstrated that most queries over probabilistic databases can be handled efficiently by following this approach. The present paper shows further that this approach provides a simple, precise and natural framework to model the semantics of applications by relational integrity constraints. In particular, the findings on normal forms, their semantic justification, and normalization apply to probabilistic databases, too. Figure 1 contains a summary of what well-known syntactic normal forms achieve for probabilistic databases.

In future work, one should address global integrity constraints and their associated normal forms, as well as soft approaches to integrity constraints. For example, one may define a probabilistic functional dependency as a pair $(X \to Y, p)$ that is satisfied by a probabilistic database whenever the probabilities of the possible worlds in which $X \to Y$ is satisfied sum up to a value at least as big as p.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] G. Anthes. Happy birthday, RDBMS! *Commun. ACM*, 53(5):16–17, 2010.

[3] L. Antova, C. Koch, and D. Olteanu. $10^{(10^6)}$ worlds and beyond: efficient representation and processing of incomplete information. *VLDB J.*, 18(5):1021–1040, 2009.

[4] M. Arenas and L. Libkin. A normal form for XML documents. *ACM Trans. Database Syst.*, 29(1):195–232, 2004.

[5] M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and XML data. *J. ACM*, 52(2):246–283, 2005.

[6] W. W. Armstrong. Dependency structures of database relationships. *Information Processing*, 74:580–583, 1974.

[7] P. Atzeni and N. Morfuni. Functional dependencies and constraints on null values in database relations. *Information and Control*, 70(1):1–31, 1986.

[8] C. Beeri and P. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Trans. Database Syst.*, 4(1):30–59, 1979.

[9] C. Beeri, M. Dowd, R. Fagin, and R. Statman. On the structure of Armstrong relations for functional dependencies. *J. ACM*, 31(1):30–46, 1984.

[10] C. Beeri, R. Fagin, and J. H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *SIGMOD*, pages 47–61. ACM, 1977.

[11] J. Biskup. Inferences of multivalued dependencies in fixed and undetermined universes. *Theor. Comput. Sci.*, 10(1):93–106, 1980.

[12] J. Biskup, U. Dayal, and P. Bernstein. Synthesizing independent database schemas. In *SIGMOD*, pages 143–151, 1979.

[13] P. Buneman, S. Davidson, W. Fan, C. Hara, and W. Tan. Keys for XML. *Computer Networks*, 39(5):473–487, 2002.

[14] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.

[15] E. F. Codd. Further normalization of the database relational model. In *Courant Computer Science Symposia 6: Data Base Systems*, pages 33–64, 1972.

[16] E. F. Codd. Recent investigations in relational data base systems. In *IFIP Congress*, pages 1017–1021, 1974.

[17] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: diamonds in the dirt. *Commun. ACM*, 52(7):86–94, 2009.

[18] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[19] J. Demetrovics. On the number of candidate keys. *Inf. Proc. Lett.*, 7:266–269, 1978.

[20] D. Dey and S. Sarkar. Generalized normal forms for probabilistic relational data. *IEEE Trans. Knowl. Data Eng.*, 14(3):485–497, 2002.

[21] J. Diederich and J. Milton. New methods and fast algorithms for database normalization. *ACM Trans. Database Syst.*, 13(3):339–365, 1988.

[22] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst.*, 2(3):262–278, 1977.

[23] R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.

[24] P. C. Fischer, L. V. Saxton, S. J. Thomas, and D. Van Gucht. Interactions between dependencies and nested relational structures. *J. Comput. Syst. Sci.*, 31(3):343–354, 1985.

[25] Z. Galil. An almost linear-time algorithm for computing a dependency basis in a relational database. *J. ACM*, 29(1):96–102, 1982.

[26] C. Hara and S. Davidson. Reasoning about nested functional dependencies. In *Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 91–100, Philadelphia, U.S.A., May 31 - June 2 1999. ACM Press.

[27] S. Hartmann, M. Kirchberg, and S. Link. A subgraph-based approach towards functional dependencies for XML. In *SCI Conference*, pages 200–205, 2003.

[28] S. Hartmann, M. Kirchberg, and S. Link. Design by example for SQL table definitions with functional dependencies. *VLDB J.*, 21(1):121–144, 2012.

[29] S. Hartmann, U. Leck, and S. Link. On Codd families of keys over incomplete relations. *The Computer Journal*, 54(7):1166–1180, 2011.

[30] S. Hartmann and S. Link. Multi-valued dependencies in the presence of lists. In *PODS*, pages 330–341, 2004.

[31] S. Hartmann and S. Link. On a problem of Fagin concerning multivalued dependencies in relational databases. *Theor. Comput. Sci.*, 353(1-3):53–62, 2006.

[32] S. Hartmann and S. Link. Characterising nested database dependencies by fragments of propositional logic. *Ann. Pure Appl. Logic*, 152(1-3):84–106, 2008.

[33] S. Hartmann and S. Link. Efficient reasoning about a robust XML key fragment. *ACM Trans. Database Syst.*, 34(2):Article 10, 2009.

[34] S. Hartmann and S. Link. Expressive, yet tractable XML keys. In *EDBT*, pages 357–367, 2009.

[35] S. Hartmann and S. Link. The implication problem of data dependencies over SQL table definitions: Axiomatic, algorithmic and logical characterizations. *ACM Trans. Database Syst.*, 37(2):13, 2012.

[36] S. Hartmann, S. Link, and K.-D. Schewe. Reasoning about functional and multi-valued dependencies in the presence of lists. In *FoIKS*, pages 134–154, 2004.

[37] S. Hartmann, S. Link, and K.-D. Schewe. Weak functional dependencies in higher-order data models. In *FoIKS*, pages 134–154, 2004.

[38] S. Hartmann, S. Link, and T. Trinh. Constraint acquisition for entity-relationship models. *Data Knowl. Eng.*, 68(10):1128–1155, 2009.

[39] I. J. Heath. Unacceptable file operations in a relational data base. In *SIGFIDET Workshop*, pages 19–33, 1971.

[40] S. Kolahi. Dependency-preserving normalization of relational and XML data. *J. Comput. Syst. Sci.*, 73(4):636–647, 2007.

[41] W. Langeveldt and S. Link. Empirical evidence for the usefulness of Armstrong relations on the acquisition of meaningful functional dependencies. *Inf. Syst.*, 35(3):352–374, 2010.

[42] M. Levene and G. Loizou. Axiomatisation of functional dependencies in incomplete relations. *Theor. Comput. Sci.*, 206(1-2):283–300, 1998.

[43] M. Levene and G. Loizou. Database design for incomplete relations. *ACM Trans. Database Syst.*, 24(1):80–125, 1999.

[44] T. W. Ling. An analysis of multivalued and join dependencies based on the entity-relationship approach. *Data Knowl. Eng.*, 1(3):253–271, 1985.

[45] T. W. Ling, F. W. Tompa, and T. Kameda. An improved third normal form for relational databases. *ACM Trans. Database Syst.*, 6(2):329–346, 1981.

[46] T. W. Ling and L.-L. Yan. NF-NR: A practical normal form for nested relations. *Journal of Systems Integration*, 4(4):309–340, 1994.

[47] S. Link. Charting the completeness frontier of inference systems for multivalued dependencies. *Acta Inf.*, 45(7-8):565–591, 2008.

[48] S. Link. On the implication of multivalued dependencies in partial database relations. *Int. J. Found. Comput. Sci.*, 19(3):691–715, 2008.

[49] S. Link. Characterisations of multivalued dependency implication over undetermined universes. *J. Comput. Syst. Sci.*, 78(4):1026–1044, 2012.

[50] C. Lucchesi and S. Osborn. Candidate keys for relations. *J. Comput. Syst. Sci.*, 17(2):270–279, 1978.

[51] H. Mannila and K.-J. Räihä. Design by example: An application of Armstrong relations. *J. Comput. Syst. Sci.*, 33(2):126–141, 1986.

[52] E. Meijer and G. M. Bierman. A co-relational model of data for large shared data banks. *Commun. ACM*, 54(4):49–58, 2011.

[53] M. Rys. Scalable SQL. *Commun. ACM*, 54(6):48–53, 2011.

[54] Y. Sagiv, C. Delobel, D. S. Parker Jr., and R. Fagin. An equivalence between relational database dependencies and a fragment of propositional logic. *J. ACM*, 28(3):435–453, 1981.

[55] A. D. Sarma, O. Benjelloun, A. Y. Halevy, S. U. Nabar, and J. Widom. Representing uncertain data: models, properties, and algorithms. *VLDB J.*, 18(5):989–1019, 2009.

[56] A. D. Sarma, J. D. Ullman, and J. Widom. Schema design for uncertain databases. In *Proceedings of the 3rd Alberto Mendelzon International Workshop on Foundations of Data Management*, volume 450 of *CEUR Workshop Proceedings*, 2009.

[57] D. Suciu, D. Olteanu, C. Ré, and C. Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[58] B. Thalheim. On semantic issues connected with keys in relational databases permitting null values. *Elektronische Informationsverarbeitung und Kybernetik*, 25(1-2):11–20, 1989.

[59] B. Thalheim. *Dependencies in relational databases*. Teubner, 1991.

[60] B. Thalheim. The number of keys in relational and nested relational databases. *Discrete Applied Mathematics*, 40(2), 1992.

[61] B. Thalheim. *Entity-relationship modeling - foundations of database technology*. Springer, 2000.

[62] B. Thalheim. Conceptual treatment of multivalued dependencies. In *Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003)*, volume 2813 of *Lecture Notes in Computer Science*, pages 363–375. Springer, 2003.

[63] M. Vincent. Semantic foundations of 4NF in relational database design. *Acta Inf.*, 36(3):173–213, 1999.

[64] M. Vincent, J. Liu, and C. Liu. Strong functional dependencies and their application to normal forms in XML. *ACM Trans. Database Syst.*, 29(3):445–462, 2004.

[65] G. Weddell. Reasoning about functional dependencies generalized for semantic data models. *ACM Trans. Database Syst.*, 17(1):32–64, 1992.

[66] M. Wu. The practical need for Fourth Normal Form. In *ACM SIGCSE*, pages 19–23, 1992.