Relational Database Schema Design for Uncertain Data

Sebastian Link
Department of Computer Science
The University of Auckland
New Zealand
s.link@auckland.ac.nz

Henri Prade
IRIT, CNRS and
Université de Toulouse III
France
henri.prade@irit.fr

ABSTRACT

We investigate the impact of uncertainty on relational database schema design. Uncertainty is modeled qualitatively by assigning to tuples a degree of possibility with which they occur, and assigning to functional dependencies a degree of certainty which says to which tuples they apply. A design theory is developed for possibilistic functional dependencies, including efficient axiomatic and algorithmic characterizations of their implication problem. Naturally, the possibility degrees of tuples result in a scale of different degrees of data redundancy. Scaled versions of the classical syntactic Boyce-Codd and Third Normal Forms are established and semantically justified in terms of avoiding data redundancy of different degrees. Classical decomposition and synthesis techniques are scaled as well. Therefore, possibilistic functional dependencies do not just enable designers to control the levels of data integrity and losslessness targeted but also to balance the classical trade-off between query and update efficiency. Extensive experiments confirm the efficiency of our framework and provide original insight into relational schema design.

CCS Concepts

•**Information systems** → **Database design and models; Uncertainty;** •**Theory of computation** → *Database constraints theory;*

Keywords

Axioms; Boyce-Codd normal form; Data redundancy; Implication problem; Third normal form; Possibility theory

1. INTRODUCTION

Relational databases were developed for applications with certain data, such as accounting, inventory, and payroll. Modern applications such as information extraction, data integration and cleaning, require techniques for uncertain data. Research on uncertain data has been prolific, yet two trends can be observed: Queries are the dominant focus point, and uncertainty is mostly modeled quantitatively as probabilistic data. Here, the impact of uncertainty on database schema design is studied qualitatively, targeting the efficient processing of frequent queries and updates.

In classical schema design, update inefficiencies are avoided by removing data redundancy caused by functional dependencies (FDs). Intuitively, if data is uncertain, then so is any redundancy that results from this data. The more possible it is for data redundancy to occur, the more FDs can cause this redundancy, and the harder the normalization effort will be to remove that redundancy. In other words, the removal of redundancy from less possible data requires normalization for a smaller number of FDs. This is great news as data that are less possible are intuitively subject to more updates. Therefore, most of the frequent updates can be supported efficiently with less normalization effort. In turn, less normalization results in better query efficiency. Our findings will establish a practical and precise framework that enables database designers to take full advantage of this intuitive impact of uncertainty.

We model uncertainty by assigning to each tuple a degree of possibility (p-degree) by which it is perceived to occur. The possibility degree comes from any given finite scale of linearly ordered values. The framework meets well the intuition and ability of people to reason qualitatively, and not quantitatively. After all, humans like to classify items into a few categories and are uncomfortable with assigning exact values such as probabilities. The framework results in a nested chain of possible worlds, each of which is a classical relation. The smallest world contains only tuples that are fully possible while the largest world contains all tuples, excluding those which are impossible to occur. Moreover, only the smallest world is regarded as the part of the database which is certain. The certainty by which an FD holds is derived from the p-degree of the smallest world in which the FD is violated. As extremes, FDs that are fully certain even hold in the largest world, and FDs that are not certain at all do not even hold in the smallest world. Our contributions are as follows:

1. We formalize uncertainty by assigning p-degrees to tuples in a database. The approach is well-founded as it results in a possibility distribution over possible worlds that form a linear chain of relations.

2. We define a possibilistic FD (pFD) as classical FDs with a degree of certainty (c-degree), derived from the p-degree of the smallest possible world in which it is violated.

CIKM'16 , October 24–28, 2016, Indianapolis, IN, USA

ISBN 978-1-4503-4073-1/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2983323.2983801

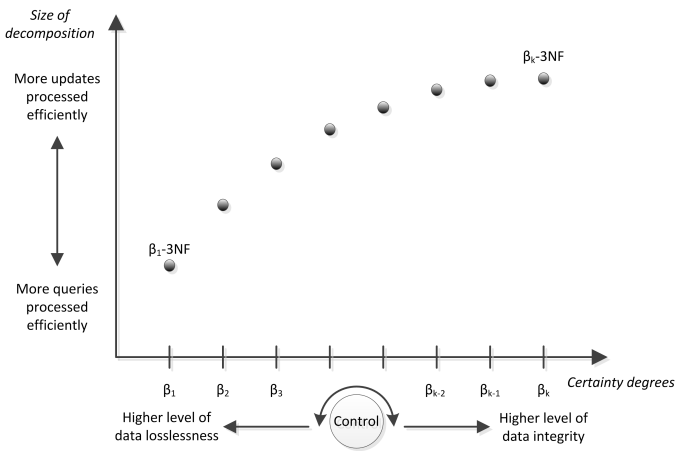

Figure 1: Exploring c-degrees as a mechanism to control levels of data integrity, data losslessness, query efficiency, and update efficiency

3. We establish a full design theory for pFDs, including axioms and algorithms for their implication problem.

4. We apply the design theory to establish a new normalization framework, including scaled versions of Boyce-Codd (BCNF) and Third normal forms (3NF), their semantic justification in terms of dependency-preservation and removal of data redundancy, as well as normalization algorithms. Our techniques provide database designers with a full scale of normalized schemata, from which better-informed selections of the final schema can be made. We envision two main use-case scenarios: i) An organization primarily focused on data integrity and losslessness may choose the FDs that apply to the target c-degree. In this case, our algorithms compute the normal forms that exactly meet that target. ii) For an organization focused on efficiently processing its workload of queries and updates, a database designer can use our techniques to derive the normal forms that best match the workload. In that case, it is clear what levels of data integrity and losslessness are achieved. Hence, c-degrees provide an efficient mechanism to balance the trade-offs between data integrity and losslessness, and between query and update efficiency. This is illustrated in Figure 1, showing how the selection of different c-degrees β_i results in different decompositions, for example, β_i-3NF for $i = 1, \ldots, k$. Our findings can justify classically de-normalized schema in terms of permitting data redundancy caused by pFDs whose c-degree is smaller than that targeted.

5. Extensive experiments with our algorithms confirm their efficiency in practice, and provide new insight into classical normalization trade-offs.

Organization. A motivating example with applications is introduced in Section 2. Related work is discussed in Section 3. Our data model of uncertainty is introduced in Section 4. PFDs are defined in Section 5, and their design theory is established in Section 6. Scaled versions of BCNF and 3NF are defined in Section 7, and their semantic justifications are derived. Normalization algorithms are established in Section 8. Experimental results are discussed in Section 9. Section 10 concludes and comments on future work. Proofs, more examples, algorithms and experiments can be found in the supplementary technical report [21].

Project	*Time*	*Manager*	*Room*	*Poss. degree*
Eagle	Mon, 9am	Ann	Aqua	α_1
Hippo	Mon, 1pm	Ann	Aqua	α_1
Kiwi	Mon, 1pm	Pete	Buff	α_1
Kiwi	Tue, 2pm	Pete	Buff	α_1
Lion	Tue, 4pm	Gill	Buff	α_1
Lion	Wed, 9am	Gill	Cyan	α_1
Lion	Wed, 11am	Bob	Cyan	α_2
Lion	Wed, 11am	Jack	Cyan	α_3
Lion	Wed, 11am	Pam	Lava	α_3
Tiger	Wed, 11am	Pam	Lava	α_4

Table 1: Possibilistic relation

2. APPLICATION AND MOTIVATION

As an example application, sufficiently simple to motive our research and explain our findings, we consider an employee who extracts information from web-sites about weekly project meetings in her company. Attributes involve *Project*, storing projects with unique names, *Time*, for the weekday and start time, *Manager*, for the managers of the project that attend, and *Room*, for the unique name of a room. The employee classifies the possibility with which tuples occur in the relation according to the levels of trust associated with the data source. Tuples from the official website are assigned p-degree α_1, indicating they are fully possible, tuples from a project manager's web-site are assigned α_2, tuples from a project member's web-site get degree α_3, and tuples that originate from rumors are assigned p-degree α_4. Implicitly, any other tuple has p-degree α_5, indicating that it is impossible to occur. A different interpretation may result from already held meetings, confirmed meetings, requested meetings, planned meetings, and all other meetings. The p-degrees may have numerical interpretations, e.g. $1 > 0.75 > 0.5 > 0.25 > 0$. Either way, the employee has chosen 5 p-degrees to assign qualitative levels of uncertainty to tuples, with top degree α_1 and bottom degree α_5. Table 1 shows a possibilistic relation (p-relation), that is, a relation where a p-degree is assigned to each tuple.

Naturally, the assignment of p-degrees results in a linearly ordered chain of possible worlds: For $i = 1, \ldots, 4$, the relation r_i consists of tuples with p-degree α_i or higher, i.e. α_j with $j \leq i$. The p-degree of world r_i is α_i. In particular, fully possible tuples occur in every possible world, and are therefore also fully certain to occur. The possible worlds of the p-relation in Table 1 are illustrated in Figure 2.

A given FD is either satisfied by the largest possible world or there is a smallest possible world in which it is violated. For example, the FD σ_1 : *Manager*, *Time* $\rightarrow$ *Room* is satisfied by the world r_4, and thus holds in every possible world. Consequently, it is assigned the top c-degree, denoted by β_1. The smallest relation that violates σ_2 : *Room*, *Time* $\rightarrow$ *Project* is r_4, that is, the FD is assigned the second highest c-degree, β_2. The smallest relation that violates σ_3 : *Project*, *Time* $\rightarrow$ *Manager* is r_3, that is, the FD is assigned the third highest c-degree, β_3. The smallest relation that violates σ_4 : *Project* $\rightarrow$ *Manager* is r_2, and the FD thus holds with c-degree β_4. The FD *Manager, Room* $\rightarrow$ *Time* is violated even by the smallest possible world r_1, and is thus assigned the bottom c-degree $\beta_5 = \beta_{k+1}$. Hence, the p-degree α_i of the smallest possible world r_i in which the FD is violated, determines the c-degree

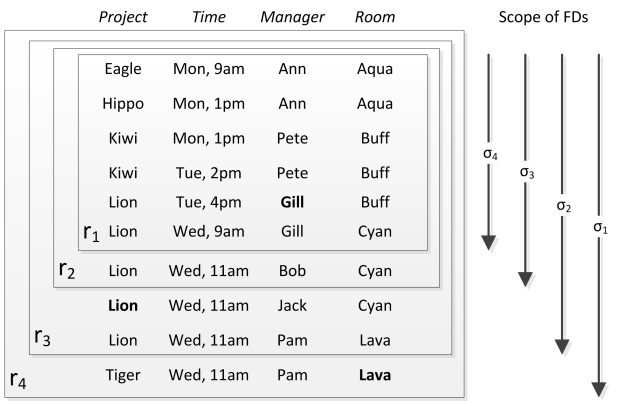

Figure 2: Worlds of p-relation from Table 1 with some redundant data values in bold and FD scopes

β_{k+2-i} with which the FD holds. The possible worlds on which the FDs $\sigma_1, \ldots, \sigma_4$ hold, i.e. their scope, are illustrated in Figure 2.

Clearly, the higher the possibility of a possible world the less tuples it contains and the more FDs may hold on it, and therefore, the more data occurrences may be redundant. Therefore, whenever we want to eliminate redundant data occurrences in more possible data, then we must expect the normalization effort to grow. Redundancy in the smallest possible world r_1, for example, could be caused by FDs that hold with any c-degree, except the bottom degree, while redundancy in the largest possible world r_4 can only be caused by FDs with the top c-degree, see Figure 2. In our running example, the bold occurrence of *Gill* in r_1 is redundant for the FD σ_4 : *Project* $\rightarrow$ *Manager* that holds with degree β_4. Indeed, any replacement of this data occurrence by a different value will result in a c-degree of σ_4 that is smaller than β_4, with which it must hold. This data value occurrence is therefore α_1-redundant, and could be caused by any pFD in Σ - all of which apply to tuples with p-degree α_1. The bold occurrence of the data value *Lion* in r_3 is α_3-redundant. It could be caused by any FD in Σ with c-degree β_1 or β_2, in this case by the FD σ_2 : *Room*, *Time* $\rightarrow$ *Project* with c-degree β_2. Finally, the bold occurrence of the data value *Lava* in r_4 is α_4-redundant. Its redundancy can only be caused by FDs that hold with c-degree β_1, in this case by σ_1 : *Manager*, *Time* $\rightarrow$ *Room*.

Therefore, the impact we expect uncertainty to have on schema design is that the more frequent updates are likely to be processed efficiently with less normalization effort. This means that more queries can be processed efficiently, as less joins are required. Uncertainty results in a greater variety of normalized schema designs from which the best match for the target workload of updates and queries can be chosen. In other words, uncertainty can be understood as an effective mechanism to better control the classical trade-off between update and query efficiency, see Figure 1. Figure 3 illustrates how different Boyce-Codd normal form decompositions apply to the p-relation of our running example.

3. RELATED WORK

Probabilistic databases have received much interest [25] due to the need to deal with uncertain data. Constraints present a key challenge here: "When the data is uncertain, constraints can be used to increase the quality of the data, and hence they are an important tool in managing data with uncertainties" [8]. Suciu et al. emphasize that "the main use of probabilities is to record the degree of uncertainty in the data and to rank the outputs to a query; in some applications, the exact output probabilities matter less to the user than the ranking of the outputs" [25]. This suggests that a qualitative approach to uncertainty, such as possibility theory [9], can avoid the high computational complexity in obtaining and processing probabilities, while guaranteeing the same qualitative outcome.

Research on probabilistic databases has naturally focused on query processing, e.g. [3, 13]. Common is the desire to extend trusted relational technology to handle uncertainty. This desire is also inherent in our approach: We show how to exploit relational normalization to design database schemata for applications with uncertain data.

No previous research has taken advantage of uncertainty information to introduce different degrees of data redundancy. These different degrees of data redundancy enable us to introduce a whole variety of normal forms each targeting different workloads and different levels of data integrity and losslessness. The linear order of our p-degrees is the counterpart of the linear order of probabilities. Extensions to partial orders are possible, but not our focus.

A companion paper [22] discusses the origins of our research in possibility theory, contains an up-to-date review of FDs for uncertain data, and characterizes pFD implication by that of Horn clauses in possibilistic logic. Design theory, normal forms, normalization, redundancy, and experiments are new contributions of the current article. Keys and cardinality constraints have also been investigated in the context of our possibilistic model [12, 14].

Few papers address schema design for uncertain data [5, 24]. Das Sarma, Ullman and Widom develop an "FD theory for data models whose basic construct for uncertainty is *alternatives*" [24]. Their work is thus fundamentally different from our approach. This is also true for Chaudhry, Moyne and Rundensteiner who model fuzziness in an extended Entity-Relationship model [5].

Finally, classical findings on relational schema design enabled us to develop our approach towards schema design for uncertain data. FDs are already mentioned in the seminal paper by Codd [6]. Armstrong axiomatized FDs [1], and linear-time algorithms to decide their implication problem are known [2]. These enable relational schema design, including 3NF [4], BCNF [7] and their semantic justification in terms of dependency-preservation and elimination of data redundancy [19, 27]. Deciding whether a given schema is in 3NF is NP-complete, and in P-time for BCNF, but deciding whether a projection of a given schema is in BCNF is coNP-complete [2]. Tsou and Fischer gave an algorithm to compute a lossless BCNF decomposition in P-time [26]. Our experiments reveal original insight into the trade-off between classical 3NF synthesis and BCNF decomposition.

4. UNCERTAIN DATABASES

We summarize the possibilistic data model from [22]. The main motivation for this data model is the schema design framework we develop in the current article.

A relation schema is a finite non-empty set R of *attributes*. Each attribute $A \in R$ has a *domain* $dom(A)$ of values. A *tuple* t over R is an element of the Cartesian product

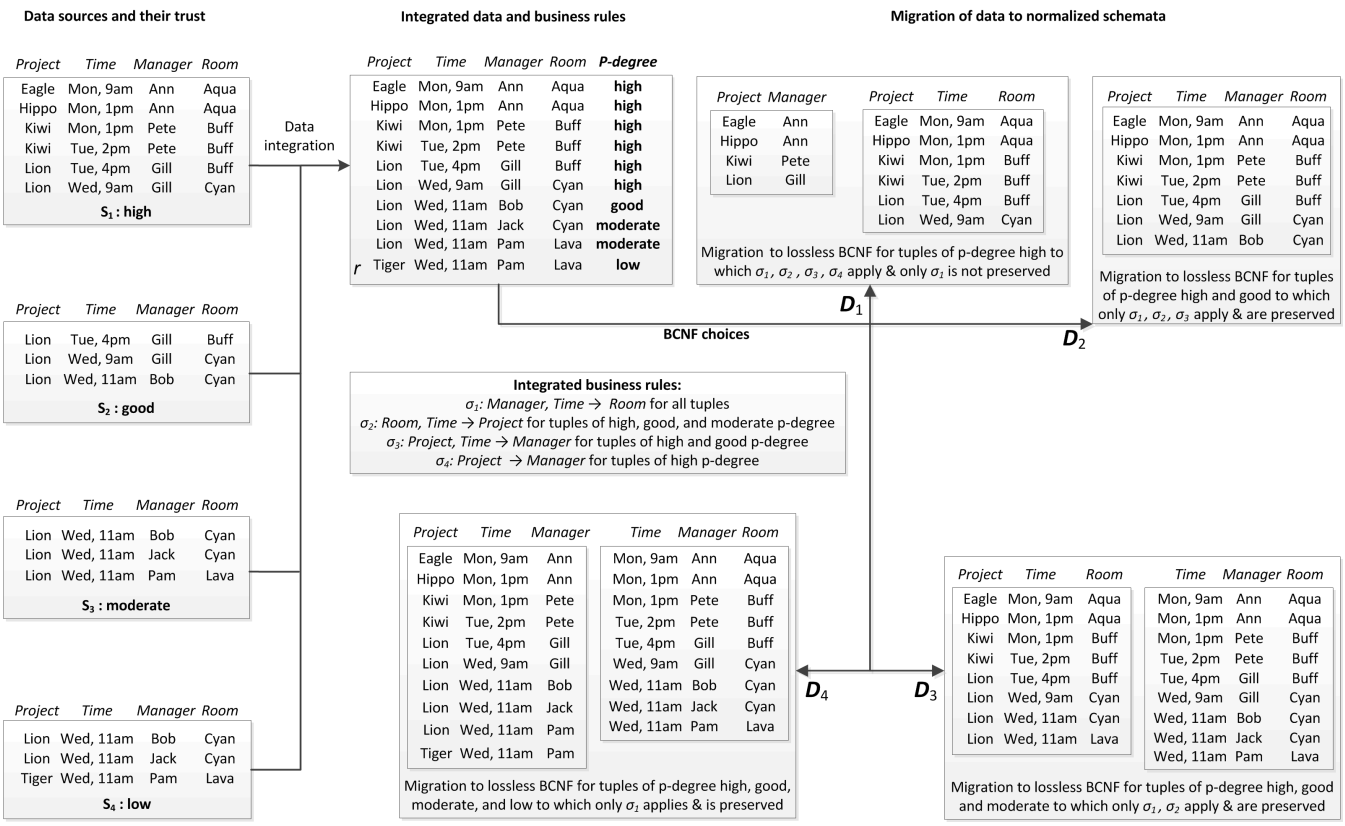

Figure 3: Using degrees of uncertainty for data integration and migration into normalized schemata

$\prod_{A \in R} dom(A)$. For $X \subseteq R$ we denote by $t(X)$ the *projection* of t on X. A *relation* over R is a finite set r of tuples over R. As example we use the relation schema MEETING={*Project*, *Time*, *Manager*, *Room*} from Section 2.

Tuples either belong to a relation or they do not, there is no room for uncertainty. For example, we cannot express less confidence for Bob attending a meeting of project Lion on Wednesday at 11am in room Cyan than for Gill attending this meeting. Database support for new applications, such as data integration, requires us to make the most out of available uncertainty information.

We define uncertain relations as relations where each tuple is associated with some confidence. The confidence expresses with which degree of possibility a tuple occurs in a relation. Formally, we model the confidence as a *scale of possibility*, i.e., a finite strict linear order $\mathcal{S} = (S, <)$ with $k+1$ elements where k is some positive integer, which we denote by $\alpha_1 > \cdots > \alpha_k > \alpha_{k+1}$. The elements α_i are *possibility degrees* (p-degrees). The top p-degree α_1 is reserved for tuples that are 'fully possible' while the bottom p-degree α_{k+1} is for tuples that are 'not possible at all', that is 'impossible', to occur in a relation. Humans like to use simple scales in everyday life, for instance to communicate, compare, or rank. Simple usually means to classify items qualitatively, rather than quantitatively by putting a precise value on it. Classical relations use a scale with two elements, i.e. where $k = 1$. As a running example, we will use a scale of five p-degrees, ranging from *fully possible* (α_1), to *rather possible* (α_2), to *medium possible* (α_3), to *somewhat possible* (α_4), and finally to *not possible at all* (α_5).

Formally, a *possibilistic relation schema* (p-schema) $(R, \mathcal{S})$ consists of a relation schema R and a possibility scale $\mathcal{S}$. A *possibilistic relation* (p-relation) over $(R, \mathcal{S})$ consists of a relation r over R, together with a function $Poss_r$ that maps each tuple $t \in r$ to a p-degree $Poss_r(t)$ in the possibility scale $\mathcal{S}$. For example, Table 1 shows a p-relation over $(\text{MEETING}, \mathcal{S} = \{\alpha_1, \ldots, \alpha_5\})$.

P-relations enjoy a well-founded semantics in terms of possible worlds. In fact, a p-relation gives rise to a possibility distribution over possible worlds of relations. For $i = 1, \ldots, k$ let r_i denote the relation that consists of all tuples in r that have a p-degree of at least α_i, i.e., $r_i = \{t \in r \mid Poss_r(t) \geq \alpha_i\}$. The linear order of the p-degrees results in a (reversed) linear order of possible worlds of relations. Indeed, we have $r_1 \subseteq r_2 \subseteq \cdots \subseteq r_k$. The possibility distribution π_r for this linear chain of possible worlds is defined by $\pi_r(r_i) = \alpha_i$. Note that r_{k+1} is not considered to be a possible world, since its possibility $\pi(r_{k+1}) = \alpha_{k+1}$ means 'not possible at all'. Vice versa, the possibility $Poss_r(t)$ of a tuple $t \in r$ is the possibility of the smallest possible world in which t occurs, i.e., the maximum possibility $\max\{\alpha_i \mid t \in r_i\}$ of a world to which t belongs. If $t \notin r_k$, then $Poss_r(t) = \alpha_{k+1}$. The top p-degree α_1 takes on a distinguished role: every tuple that is 'fully possible' occurs in every possible world - and is thus - 'fully certain'. This formally confirms our intuition that uncertain relations subsume relations (of fully certain tuples) as a special case. Figure 2 shows the possible worlds $r_1 \subsetneq r_2 \subsetneq r_3 \subsetneq r_4$ of the p-relation in Table 1.

5. POSSIBILISTIC FDS

We introduce possibilistic FDs (pFDs) as classical FDs with a degree of certainty. In the same way FDs are fundamental to relational database design, pFDs will play a fundamental role to schema design for uncertain data.

Recall that an FD $X \to Y$ over relation schema R is satisfied by a relation r over R whenever every pair of tuples in r that have matching values on all the attributes in X have also matching values on all the attributes in Y. For example, the FD σ_5 : *Manager, Room* $\to$ *Time* is not satisfied by any relation $r_1, \ldots, r_4$. The FD σ_4 : *Project* $\to$ *Manager* is satisfied by r_1, but not by r_2 and therefore not by r_3 and r_4. The FD σ_3 : *Project, Time* $\to$ *Manager* is satisfied by r_1 and r_2, but not by r_3 and therefore not by r_4. The FD σ_2 : *Time, Room* $\to$ *Project* is satisfied by r_1, r_2, and r_3, but not by r_4. Finally, the FD σ_1 : *Manager, Time* $\to$ *Room* is satisfied by all relations $r_1, \ldots, r_4$.

Naturally, the p-degrees of tuples that define an uncertain relation also define degrees of certainty with which FDs hold in the uncertain relation. Intuitively, since σ_1 is satisfied in every possible world, it is fully certain (β_1) to hold in r. As σ_2 is only violated in a somewhat possible world r_4, it is quite certain (β_2) to hold in r. Since σ_3 is only violated in a medium possible world r_3, it is medium certain (β_3) to hold in r. As σ_4 is only violated in a quite possible world r_2, it is somewhat certain (β_4) to hold in r. Finally, as σ_5 is violated in the fully possible world r_1, it is not certain at all (β_5) to hold in r.

In summary, the certainty with which an FD holds in an uncertain relation corresponds to the possibility of the smallest possible world in which the FD is violated. Therefore, similar to a scale $\mathcal{S}$ of possibility degrees for tuples we use a scale $\mathcal{S}^T$ of certainty degrees (c-degrees) for FDs. We commonly use subscripted versions of the Greek letter β to denote c-degrees associated with FDs. Formally, the correspondence between p-degrees in $\mathcal{S}$ and the c-degrees in $\mathcal{S}^T$ can be defined by the mapping $\alpha_i \mapsto \beta_{k+2-i}$ for $i = 1, \ldots, k+1$. Formally then, the certainty $C_r(X \to Y)$ with which the FD $X \to Y$ holds in the uncertain relation r is the top degree β_1 whenever $X \to Y$ is satisfied by r_k, or otherwise the minimum amongst the c-degrees β_{k+2-i} that correspond to possible worlds r_i in which $X \to Y$ is violated, that is, $C_r(X \to Y)$

$$= \begin{cases} \beta_1 & \text{, if } r_k \text{ satisfies } X \to Y \\ \min\{\beta_{k+2-i} \mid\not\models_{r_i} X \to Y\} & \text{, otherwise} \end{cases} .$$

We can now define the syntax and semantics of pFDs.

Definition 1. *A* possibilistic FD *(pFD) over a p-schema $(R, \mathcal{S})$ is an expression $(X \to Y, \beta)$ where $X, Y \subseteq R$ and $\beta \in S^T$. A p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ is said to* satisfy *the pFD $(X \to Y, \beta)$ if and only if $C_r(X \to Y) \geq \beta$.*

The p-relation $(r, Poss_r)$ of our running example satisfies: (σ_1, β_1), (σ_2, β_2), (σ_3, β_3), (σ_4, β_4), and (σ_5, β_5). It violates the pFD (σ_3, β_2) since $C_r(\sigma_3) = \beta_3 < \beta_2$.

6. SCALED DESIGN THEORY

Classical normalization is founded on the theory of FDs. Consequently, we now establish a design theory of pFDs as a foundation for normalizing uncertain data.

6.1 The Magic of β-Cuts

We link the implication of pFDs and FDs. Let $\Sigma \cup \{\varphi\}$ denote a set of pFDs over a p-schema $(R, \mathcal{S})$. We say that Σ *implies* φ, denoted by $\Sigma \models \varphi$, if every p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ that satisfies every pFD in Σ also satisfies φ.

Example 1. *Let Σ consist of the following four pFDs $(Manager, Time \to Room, \beta_1)$, $(Room, Time \to Project, \beta_2)$, $(Project, Time \to Manager, \beta_3)$, $(Project \to Manager, \beta_4)$ over* (Meeting, $\{\alpha_1, \ldots, \alpha_5\}$). *Further, let φ denote the pFD $(Room, Time \to Manager, \beta_2)$. Then Σ does not imply φ as the following p-relation witnesses:*

Project	*Time*	*Manager*	*Room*	*Poss. degree*
Lion	*Wed, 3pm*	*Gill*	*Cyan*	α_1
Lion	*Wed, 3pm*	*Robert*	*Cyan*	α_3

.

For a set Σ of pFDs on some p-schema $(R, \mathcal{S})$ and c-degree $\beta \in \mathcal{S}^T$ where $\beta > \beta_{k+1}$, let

$$\Sigma_\beta = \{X \to Y \mid (X \to Y, \beta') \in \Sigma \text{ and } \beta' \geq \beta\}$$

be the *β-cut* of Σ. The major strength of our framework is engraved in the following result.

Theorem 1. *Let $\Sigma \cup \{(X \to Y, \beta)\}$ be a set of pFDs over a p-schema $(R, \mathcal{S})$ where $\beta > \beta_{k+1}$. Then $\Sigma \models (X \to Y, \beta)$ if and only if $\Sigma_\beta \models X \to Y$.* □

Example 2. *Let Σ and φ be as in Example 1, in particular Σ does not imply φ. Theorem 1 says that Σ_{β_2} does not imply Room, Time $\to$ Manager. Indeed, the possible world r_3 of the p-relation from Example 1*

Project	*Time*	*Manager*	*Room*
Lion	*Wed, 3pm*	*Gill*	*Cyan*
Lion	*Wed, 3pm*	*Robert*	*Cyan*

satisfies the two classical FDs Manager, Time $\to$ Room and Room, Time $\to$ Project that form Σ_{β_2}, and violates the FD Room, Time $\to$ Manager.

6.2 Scaled Armstrong Axioms

The *semantic closure* $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ contains all dependencies implied by Σ. One can utilize a syntactic approach to compute Σ^* by applying *inference rules* of the form $\frac{\text{premise}}{\text{conclusion}}$ condition, where rules without a premise are called *axioms*. For a set $\mathfrak{R}$ of inference rules let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\sigma_1, \ldots, \sigma_n$ such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or is the conclusion that results from an application of an inference rule in $\mathfrak{R}$ to some premises in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. Let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ denote the *syntactic closure* of Σ under inferences by $\mathfrak{R}$. $\mathfrak{R}$ is *sound* (*complete*) if for every p-schema $(R, \mathcal{S})$ and for every set Σ we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* if $\mathfrak{R}$ is both sound and complete.

PFDs enjoy the axiomatization $\mathfrak{P}$ from Table 2. It subsumes Armstrong's axioms [1] for FDs as the special case where the scale $\mathcal{S}^T$ consists of just two c-degrees. In these rules all attribute sets X, Y, Z are subsets of the arbitrarily given relation schema R, and the c-degrees β and β' are elements of the arbitrarily given certainty scale $\mathcal{S}^T$. In particular, β_{k+1} denotes the bottom c-degree.

$$\frac{}{(XY \to Y, \beta)}\ (\text{reflexivity, } \mathcal{R}) \qquad \frac{(X \to Y, \beta)}{(X \to XY, \beta)}\ (\text{extension, } \mathcal{E})$$

$$\frac{(X \to Y, \beta) \quad (Y \to Z, \beta)}{(X \to Z, \beta)}\ (\text{transitivity, } \mathcal{T})$$

$$\frac{}{(X \to Y, \beta_{k+1})}\ (\text{bottom, } \mathcal{B}) \qquad \frac{(X \to Y, \beta)}{(X \to Y, \beta')}\ \beta' < \beta\ (\text{weakening, } \mathcal{W})$$

Table 2: Axiomatization $\mathfrak{P} = \{\mathcal{R}, \mathcal{E}, \mathcal{T}, \mathcal{B}, \mathcal{W}\}$ of pFDs

THEOREM 2. *The set $\mathfrak{P} = \{\mathcal{R}, \mathcal{E}, \mathcal{T}, \mathcal{B}, \mathcal{W}\}$ forms a finite axiomatization for the implication of pFDs.* □

EXAMPLE 3. *Let Σ and φ be as in Example 1, and $\varphi' = (Room, Time \to Manager, \beta_3)$. We show an inference of φ' from Σ by $\mathfrak{P}$. Attributes are abbreviated by their first letters.*

$$\dfrac{\dfrac{\dfrac{(RT \to P, \beta_2)}{\mathcal{W} : (RT \to P, \beta_3)}}{\mathcal{E} : (RT \to RTP, \beta_3)} \quad \dfrac{}{\mathcal{R} : (RTP \to PT, \beta_3)}}{\mathcal{T} : \dfrac{(RT \to PT, \beta_3) \qquad (PT \to M, \beta_3)}{(RT \to M, \beta_3)}}$$

Of course, φ cannot be inferred from Σ by $\mathfrak{P}$, as Example 1 and the soundness of $\mathfrak{P}$ show.

6.3 Scaled Decision Algorithm

In practice it is often unnecessary to compute Σ^*. Rather frequent is the problem of deciding if Σ implies φ.

PROBLEM: IMPLICATION	
INPUT:	Relation schema R, Scale $\mathcal{S}$ with $k+1$ possibility degrees, Set $\Sigma \cup \{\varphi\}$ of pFDs over $(R, \mathcal{S})$
OUTPUT:	Yes, if $\Sigma \models \varphi$, and No, otherwise

Computing Σ^* to check if $\varphi \in \Sigma^*$ is hardly efficient nor makes use of the input φ. We can exploit Theorem 1 to derive an algorithm that decides the implication problem for pFDs in time linear in the input. Given a pFD set $\Sigma \cup \{(X \to Y, \beta)\}$ we return *true* if $\beta = \beta_{k+1}$, otherwise we check if $\Sigma_\beta \models X \to Y$. The latter test can be done in linear time by computing the attribute set closure $X^+_{\Sigma_\beta} = \{A \in R \mid \Sigma \vdash_{\mathfrak{A}} X \to A\}$ of X for Σ_β [2]. We use $||\Sigma||$ for the total number of attributes in Σ, and $|X|$ for the cardinality of X. We obtain the following result.

THEOREM 3. *The implication problem $\Sigma \models \varphi$ of pFDs can be decided in time $\mathcal{O}(||\Sigma \cup \{\varphi\}||)$.* □

EXAMPLE 4. *Let Σ and φ be as in Example 1, and $\varphi' = (Room, Time \to Manager, \beta_3)$. It follows that φ is not implied by Σ as* CLOSURE(MEETING,*RT*,Σ_{β_2})=*RTP. Further, φ' is implied by Σ as* CLOSURE(MEETING,*RT*,Σ_{β_3})=*RTPM.*

7. SCALED NORMAL FORMS

In relational databases, Boyce-Codd normal form (BCNF) syntactically characterizes relation schemata that are guaranteed to be free of data redundancy in any relations over the schema, in terms of FDs [27]. Third normal form (3NF) syntactically characterizes relation schemata that are guaranteed to have the least amount of data redundancy in their relations amongst all schemata on which all FDs can be enforced locally [19]. In relations, data redundancy is treated uniformly for all data, and the elimination of all data redundancy requires normalization for all FDs.

In p-relations, different data can occur with different p-degrees. Therefore, data redundancy may occur with different p-degrees, too. Intuitively, the smaller the p-degree for which data redundancy is to be eliminated, the smaller the normalization effort will be. We will introduce notions of data redundancy that are tailored to the p-degree of tuples in which they occur. This results in a whole range of semantic normal forms by which data redundancy of growing p-degrees are eliminated. We then characterize each of these semantic normal forms by a corresponding syntactic normal form, and establish strong correspondences with BCNF and 3NF in relational databases.

7.1 Scaled Data Redundancy

Building on Section 2 we now introduce notions of data redundancy that are tailored to the p-degree of tuples. We exploit the classical proposal by Vincent [27]. Let R denote a relation schema, A an attribute of R, t a tuple over R, and Σ a set of FDs over R. A *replacement* of $t(A)$ is a tuple $\bar{t}$ over R such that: i) for all $\bar{A} \in R - \{A\}$ we have $\bar{t}(\bar{A}) = t(\bar{A})$, and ii) $\bar{t}(A) \neq t(A)$. For a relation r over R that satisfies Σ and $t \in r$, the data value occurrence $t(A)$ in r is *redundant* for Σ iff for every replacement $\bar{t}$ of $t(A)$, $\bar{r} := (r - \{t\}) \cup \{\bar{t}\}$ violates some FD in Σ. A relation schema R is in *Redundancy-Free normal form* (RFNF) for a set Σ of FDs iff there are no relation r over R that satisfies Σ, tuple $t \in r$, and attribute $A \in R$ such that the data value occurrence $t(A)$ is redundant for Σ [27].

DEFINITION 2. *Let $(R, \mathcal{S})$ denote a p-schema, Σ a set of pFDs over $(R, \mathcal{S})$, $A \in R$ an attribute, $(r, Poss_r)$ a p-relation over $(R, \mathcal{S})$ that satisfies Σ, and t a tuple in r_i. The data value occurrence $t(A)$ is* α_i-redundant *iff $t(A)$ is redundant for $\Sigma_{\alpha_i} = \{X \to Y \mid (X \to Y, \beta) \in \Sigma \text{ and } \beta \geq \beta_{k+1-i}\}$.* □

In our motivating example the occurrences of *Lava*, *Lion*, and *Gill* are α_4-, α_3- and α_1-redundant, respectively. Importantly, α_i-redundant data value occurrences can only be caused by pFDs $(X \to Y, \beta)$ that apply to the world of the occurrence, i.e. where $\beta \geq \beta_{k+1-i}$. Hence, α_1-redundancy can be caused by pFDs with any c-degree $\beta_1, \ldots, \beta_k$, while α_k-redundancy can only be caused by pFDs with c-degree β_1. We have arrived at the following definition.

DEFINITION 3. *A p-schema $(R, \mathcal{S})$ is in α_i-Redundancy-Free Normal Form (α_i-RFNF) for a set Σ of pFDs over $(R, \mathcal{S})$ iff there do not exist a p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ that satisfies Σ, an attribute $A \in R$, and a tuple $t \in r_i$ such that $t(A)$ is α_i-redundant.* □

(MEETING, $\mathcal{S}$) is not in α_4-RFNF, α_3-RFNF, nor α_1-RFNF, but it is in α_2-RFNF for Σ. The negative results follows

from Figure 2, but the satisfaction of the α_2-RFNF condition is not obvious. The next result shows that α_i-RFNF characterizes p-schemata that permit only p-relations whose possible world r_i is free from data redundancy caused by the classical FDs that apply to it.

THEOREM 4. *$(R, \mathcal{S})$ is in α_i-RFNF for Σ if and only if R is in RFNF for Σ_{α_i}.*

7.2 Scaled BCNF

We now characterize α-RFNF, which is a semantic normal form, purely syntactically. This is achieved by scaling the classical BCNF condition. A relation schema R is in Boyce-Codd normal form (BCNF) for an FD set Σ over R iff for all $X \to Y \in \Sigma^+_{\mathfrak{A}}$ where $Y \not\subseteq X$, we have $X \to R \in \Sigma^+_{\mathfrak{A}}$. Here, $\Sigma^+_{\mathfrak{A}}$ is the syntactic closure of Σ for the Armstrong axioms $\mathfrak{A}$ [1]. While α-RFNF is defined in terms of the semantics in the possible world with p-degree α, scaled BCNF is defined in terms of the syntax of the given pFDs with their c-degrees.

DEFINITION 4. *A p-schema $(R, \mathcal{S})$ is in β-*Boyce-Codd Normal Form *for a set Σ of pFDs over $(R, \mathcal{S})$ iff for every pFD $(X \to Y, \beta) \in \Sigma^+_{\mathfrak{P}}$ where $Y \not\subseteq X$, we have $(X \to R, \beta) \in \Sigma^+_{\mathfrak{P}}$.* □

Sets Σ and Θ are covers for one another iff $\Sigma^* = \Theta^*$ holds. Being in β-BCNF for Σ is independent of a cover for Σ. That is, for any cover Σ', $(R, \mathcal{S})$ is in β-BCNF for Σ iff $(R, \mathcal{S})$ is in β-BCNF for Σ'. The β-BCNF condition on a pFD set Σ is equivalent to the BCNF condition on the FD set Σ_β.

THEOREM 5. *$(R, \mathcal{S})$ is in β-BCNF for a set Σ if and only if R is in BCNF for Σ_β.*

We are now in a position to characterize the semantic α_i-RFNF by the syntactic β_{k+1-i}-BCNF.

THEOREM 6. *For $i = 1, \ldots, k$, $(R, \mathcal{S})$ with $|\mathcal{S}| = k + 1$ is in α_i-RFNF for Σ iff $(R, \mathcal{S})$ is in β_{k+1-i}-BCNF for Σ.*

One may wonder about the efficiency of checking if a given p-schema $(R, \mathcal{S})$ is in β-BCNF for a set Σ. Indeed, it suffices to check some pFDs in Σ instead of checking all pFDs in $\Sigma^+_{\mathfrak{P}}$.

THEOREM 7. *A p-schema $(R, \mathcal{S})$ is in β-BCNF for a set Σ of pFDs over $(R, \mathcal{S})$ iff for every pFD $(X \to Y, \beta') \in \Sigma$ where $\beta' \geq \beta$ and $Y \not\subseteq X$ we have $(X \to R, \beta) \in \Sigma^+_{\mathfrak{P}}$.*

EXAMPLE 5. *Let* (MEETING, $\mathcal{S}$) *and Σ be as in Example 1. Using Theorem 7 we can observe that the schema is neither in β_1-, nor β_2−, nor β_4-BCNF for Σ, but it is in β_3-BCNF for Σ. By Theorem 6 we conclude that the schema is neither in α_4-, nor α_3-, nor α_1-RFNF for Σ, but it is in α_2-RFNF for Σ. By Theorem 5 it follows that* MEETING *is neither in BCNF for Σ_{β_1}, nor Σ_{β_2}, nor Σ_{β_4}, but it is in BCNF for Σ_{β_3}. Finally, by Theorem 4, it follows that* MEETING *is neither in RFNF for Σ_{α_4}, nor Σ_{α_3}, nor Σ_{α_1}, but it is in RFNF for Σ_{α_2}.*

7.3 Scaled Third Normal Form

We now introduce β-Third normal form (3NF), whose goal is to ensure that all FDs can be enforced locally, without the need of joining relations to check for consistency of updates. Recall the classical 3NF condition: A relation schema R is in *3NF* for a given FD set Σ iff for every FD $X \to A \in \Sigma^+_{\mathfrak{A}}$ where $A \notin X$, we have $X \to R \in \Sigma^+_{\mathfrak{A}}$ or A is a prime attribute. An attribute A is *prime* iff it occurs in some minimal key for Σ. An attribute subset X of R is a *key* of R for Σ iff $X \to R \in \Sigma^+_{\mathfrak{A}}$. A key X of R is *minimal* for Σ iff every proper subset Y of X is not a key of R for Σ.

We now introduce analogous concepts for the possibilistic setting. Given a p-schema $(R, \mathcal{S})$, a c-degree $\beta \in \mathcal{S}^T$, and a set Σ of pFDs over $(R, \mathcal{S})$, a subset X of R is a *β-key* of R for Σ iff $(X \to R, \beta) \in \Sigma^+_{\mathfrak{P}}$. A β-key X of R for Σ is a *β-minimal key* iff every proper subset Y of X is not a β-key of R for Σ. An attribute $A \in R$ is *β-prime* iff it is contained in some β-minimal key X of R for Σ.

DEFINITION 5. *A p-schema $(R, \mathcal{S})$ is in β-Third Normal Form (3NF) for a set Σ of pFDs over $(R, \mathcal{S})$ iff for every pFD $(X \to A, \beta) \in \Sigma^+_{\mathfrak{P}}$ where $A \notin X$, we have $(X \to R, \beta) \in \Sigma^+_{\mathfrak{P}}$ or A is a β-prime attribute.* □

Theorem 5 characterized β-BCNF for the pFD set Σ in terms of classical BCNF for the β-cut Σ_β. An analogous result holds for 3NF.

THEOREM 8. *$(R, \mathcal{S})$ is in β-3NF for a set Σ if and only if R is in 3NF for Σ_β.*

Deciding if a schema is in β-3NF can be done by checking some pFDs in Σ instead of checking all pFDs in $\Sigma^+_{\mathfrak{P}}$.

THEOREM 9. *A p-schema $(R, \mathcal{S})$ is in β-3NF for a set Σ of pFDs over $(R, \mathcal{S})$ iff for every pFD $(X \to Y, \beta') \in \Sigma$ where $\beta' \geq \beta$ and $Y \not\subseteq X$, we have $(X \to R, \beta) \in \Sigma^+_{\mathfrak{P}}$ or every attribute $A \in Y - X$ is a β-prime.*

EXAMPLE 6. *Let* (MEETING, $\mathcal{S}$) *and Σ be as in Example 1. By Theorem 9, the schema is in neither β_1- nor β_2-3NF for Σ, but it is in β_3-3NF and in β_4-3NF for Σ. Finally, by Theorem 8, it follows that* MEETING *is neither in 3NF for Σ_{β_1} nor Σ_{β_2}, but it is in 3NF for Σ_{β_3} and in 3NF for Σ_{β_4}.*

8. SCALED NORMALIZATION

We establish algorithms to design relational schemata for applications with uncertain data. For that purpose, we normalize a given p-schemata $(R, \mathcal{S})$ for the given pFD set Σ. Our strategy is to fix a c-degree $\beta \in \mathcal{S}^T$ that determines which possible worlds we normalize for which FDs. We pursue BCNF decompositions to obtain lossless decompositions free from any data redundancy but potentially not dependency-preserving (that is, some FDs may require validation on the join of some relations). We also pursue 3NF-synthesis to obtain lossless, dependency-preserving decompositions where the level of data redundancy is minimal for all decompositions that are dependency-preserving. Applying our strategy to different c-degrees provides organizations with a whole scale of normalized schemata, each targeted at different levels of data integrity, data losslessness, and the efficiency of different updates and queries.

8.1 Scaled BCNF Decomposition

We recall classical terminology. A *decomposition* of relation schema R is a set $\mathcal{D} = \{R_1, \ldots, R_n\}$ of relation schemata such that $R_1 \cup \cdots \cup R_n = R$. For $R_j \subseteq R$ and FD set Σ over R, $\Sigma[R_j] = \{X \to Y \mid X \to Y \in \Sigma^+_{\mathfrak{A}}$ and $X, Y \subseteq R_j\}$ is the *projection* of Σ onto R_j. A decomposition $\mathcal{D}$

of a relation schema R with FD set Σ is *lossless* iff every relation r over R that satisfies Σ is the join of its projections on the elements of $\mathcal{D}$, that is, $r = \bowtie_{R_j \in \mathcal{D}} r[R_j]$. Here, $r[R_j] = \{t(R_j) \mid t \in r\}$. A *BCNF* decomposition of a relation schema R with FD set Σ is a decomposition $\mathcal{D}$ of R where every $R_j \in \mathcal{D}$ is in BCNF for $\Sigma[R_j]$. Theorem 5 motivates Definition 6.

DEFINITION 6. *An α_{k+1-i}-lossless BCNF decomposition of a p-schema $(R, \{\alpha_1, \ldots, \alpha_{k+1}\})$ for the pFD set Σ is a lossless BCNF decomposition of R for Σ_{β_i}.* □

Instrumental to Definition 6 is the following decomposition theorem. It covers the classical decomposition theorem [23] as the special case of having just one possible world.

THEOREM 10. *Let $(X \to Y, \beta_i)$ be a pFD that satisfies the p-relation $(r, Poss_r)$ over the p-schema $(R, \mathcal{S})$. Then $r_{k+1-i} = r_{k+1-i}[XY] \bowtie r_{k+1-i}[X(R-Y)]$, that is, the possible world r_{k+1-i} of r is the lossless join of its projections on XY and $X(R-Y)$.*

Therefore, an α_{k+1-i}-lossless BCNF decomposition for a pFD set Σ can be obtained by performing a classical lossless BCNF decomposition for the β_i-cut Σ_{β_i} of Σ. This suggests a simple lossless BCNF decomposition strategy.

PROBLEM: Scaled BCNF Decomposition	
INPUT:	P-Relation Schema $(R, \mathcal{S})$ Set Σ of pFDs over $(R, \mathcal{S})$ Certainty degree $\beta_i \in \mathcal{S}^T$
OUTPUT:	α_{k+1-i}-lossless BCNF decomposition of $(R, \mathcal{S})$ for Σ
METHOD:	Perform a lossless BCNF decomposition of R for Σ_{β_i}

We illustrate the decomposition on our running example.

EXAMPLE 7. *Let (MEETING, $\mathcal{S}$) and Σ be as in Example 1. As (MEETING, $\mathcal{S}$) is not in β_2-BCNF for Σ, we perform an α_3-lossless BCNF decomposition for Σ_{β_2}. The result is the decomposition D_3 from Figure 3: $R_1 = \{Project, Room, Time\}$ with projected FD set $\Sigma_{\beta_2}[R_1] = \{Room, Time \to Project\}$, and $R_2 = \{Manager, Room, Time\}$ with projected FD set $\Sigma_{\beta_2}[R_2] = \{Manager, Time \to Room\}$. Every FD in Σ_{β_2} is implied by $\Sigma_{\beta_2}[R_1] \cup \Sigma_{\beta_2}[R_2]$.* □

The last example is rather special, since one cannot hope to preserve all FDs in the BCNF decomposition process. This is illustrated with another example.

EXAMPLE 8. *Let (MEETING, $\mathcal{S}$) and Σ be as in Example 1. As (MEETING, $\mathcal{S}$) is not in β_4-BCNF for Σ, an α_1-lossless BCNF decomposition for Σ_{β_4} results in D_1 from Figure 3: $R_1 = \{Manager, Project\}$ with projected FD set $\Sigma_{\beta_4}[R_1] = \{Project \to Manager\}$, and $R_2 = \{Project, Room, Time\}$ with projected FD set $\Sigma_{\beta_4}[R_2] = \{Room, Time \to Project; Project, Time \to Room\}$. The FD $Manager, Time \to Room$ is not implied by $\Sigma_{\beta_4}[R_1] \cup \Sigma_{\beta_4}[R_2]$.* □

A decomposition $\mathcal{D}$ of relation schema R with FD set Σ is *dependency-preserving* if and only if $\Sigma^+_{\mathfrak{A}} = (\cup_{R_j \in \mathcal{D}} \Sigma[R_j])^+_{\mathfrak{A}}$.

DEFINITION 7. *A β-dependency-preserving decomposition of a p-schema $(R, \mathcal{S})$ for the pFD set Σ is a dependency-preserving decomposition of R for Σ_β.* □

The α_3-lossless BCNF decomposition from Example 7 is β_2-dependency-preserving, but the α_1-lossless BCNF decomposition from Example 8 is not β_4-dependency-preserving. In fact, for (MEETING, $\mathcal{S}$) and Σ as in Example 1 there is no β_4-dependency-preserving, α_1-lossless BCNF decomposition of Σ. In practice, lost dependencies can only be validated by joining relations after inserts or modification. For example, to validate the FD *Manager, Time → Room* after an update, one would have to join R_1 and R_2 from Example 8. This can be prohibitively expensive.

8.2 Scaled 3NF Synthesis

3NF synthesis guarantees dependency-preservation, but cannot guarantee the elimination of all data redundancy in terms of FDs. Recently, Third normal form was shown to pay the smallest possible price, in terms of data redundancy, for achieving dependency-preservation [19]. We will now equip our schema design framework for uncertain data with an appropriate 3NF synthesis strategy.

A 3NF decomposition of a relation schema R for an FD set Σ is a decomposition $\mathcal{D}$ of R where every $R_j \in \mathcal{D}$ is in 3NF for $\Sigma[R_j]$. Theorem 8 motivates the following definition.

DEFINITION 8. *A β_i-dependency-preserving, α_{k+1-i}-lossless 3NF decomposition of a p-schema $(R, \{\alpha_1, \ldots, \alpha_{k+1}\})$ for the pFD set Σ is a dependency-preserving, lossless 3NF decomposition of R for Σ_{β_i}.* □

Due to Theorem 10 a β_i-dependency-preserving, α_{k+1-i}-lossless 3NF synthesis for a pFD set Σ can simply be obtained by performing a classical dependency-preserving lossless 3NF synthesis for the β_i-cut Σ_{β_i} of Σ.

PROBLEM: Scaled 3NF Synthesis	
INPUT:	P-relation schema $(R, \mathcal{S})$ Set Σ of pFDs over $(R, \mathcal{S})$ Certainty degree $\beta_i \in \mathcal{S}^T$
OUTPUT:	β_i-dependency-preserving, α_{k+1-i}-lossless 3NF decomposition of $(R, \mathcal{S})$ for Σ
METHOD:	Perform a dependency-preserving, lossless 3NF synthesis of R for Σ_{β_i}

We illustrate the synthesis on our running example.

EXAMPLE 9. *Let (MEETING, $\mathcal{S}$) and Σ be as in Example 1. As (MEETING, $\mathcal{S}$) is not in β_2-3NF for Σ, we perform an α_3-lossless, β_2-dependency-preserving 3NF synthesis. The result consists of $R_1 = \{Project, Room, Time\}$ with projected FD set $\Sigma_{\beta_2}[R_1] = \{Room, Time \to Project\}$, and $R_2 = \{Manager, Room, Time\}$ with projected FD set $\Sigma_{\beta_2}[R_2] = \{Manager, Time \to Room\}$. Note that R_1 is in BCNF for $\Sigma_{\beta_2}[R_1]$, and R_2 is in BCNF for $\Sigma_{\beta_2}[R_2]$, as observed in Example 7.* □

9. EXPERIMENTS

We analyze the results of our experiments for the scaled versions of BCNF decompositions and 3NF syntheses. This also includes new insights into the traditional problems.

Our core analysis concerns the trade-offs between the elimination of data redundancy achieved by BCNF decompositions and the preservation of FDs achieved by 3NF syntheses, as well as between query and update efficiency. The simplest general measure for the query and update efficiency

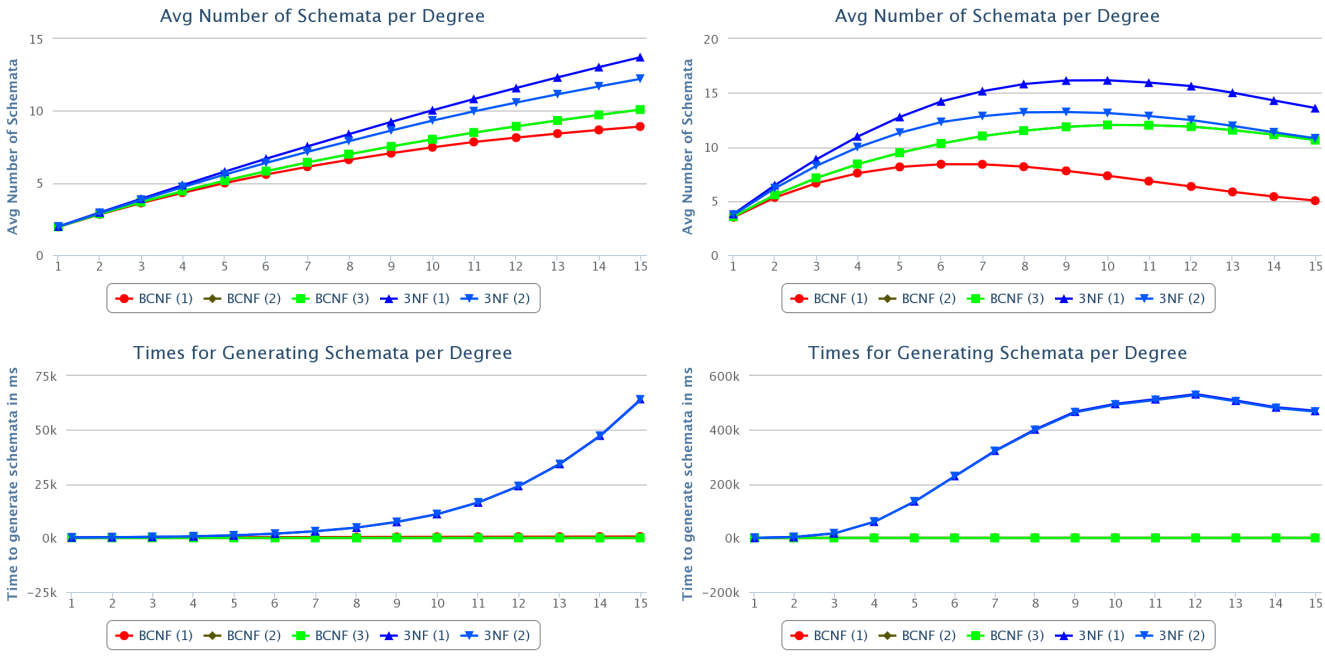

Figure 4: Average size of decompositions and average times to generate them, taken over 5,000 runs each, where the input relation schemata have $n = 15$ attributes, and the maximum pFD set size is n in the left column and $5n$ in the right column

of a decomposition is the *number of its schemata* - to which we refer as the size of the decomposition. Intuitively, the more schemata a decomposition contains the more updates and the less queries it can support efficiently. In particular, less schemata require less joins, the main source of complexity in query evaluation. The normalization effort can be measured either in view of the given c-degree β (possibilistic view), or in the size of its β-cut (classical view). Our analysis should be read with both views in mind.

For our analysis we compare five normalization algorithms: BCNF(1) is the classical BCNF decomposition algorithm that takes exponential time and space, BCNF(2) is Tsou and Fischer's P-time algorithm, BCNF(3) is the variant of BCNF(2) where the projection of original FD sets is only determined for the output schemata, 3NF(1) is the 3NF synthesis algorithm that first computes a non-redundant, L-reduced cover in which all FDs have a singleton attribute on the right-hand side, and 3NF(2) is the 3NF synthesis algorithm that first computes a canonical cover.

The top row of Figure 4 shows the graphs of two experiments. For each of 15 c-degrees $\beta_1, \ldots, \beta_{15}$, 5,000 sets of at most $a \cdot n$ pFDs over a relation schema with $n = 15$ attributes were created, and β_i-BCNF decompositions and β_i-3NF syntheses created, using the 5 algorithms. The graph with $a = 1$ is shown on the left and $a = 5$ on the right.

The experiments show nearly uniform results: The smallest-sized decomposition is BCNF(1), followed by BCNF(2,3) which agree, followed by 3NF(2), and then 3NF(1). The results are intuitive as 3NF produces more schemata to accommodate the preservation of all FDs and thereby sacrificing the elimination of some data redundancy. The graphs show the price for producing a BCNF decomposition in P-time, which results from not having to check the BCNF condition of intermediate results and thereby producing superfluous schemata not required by BCNF(1). It is interesting that this price is still mostly smaller than that to accommodate dependency-preservation. However, we will see later that the superfluous schemata in BCNF(2,3) are unlikely to help with dependency-preservation. Finally, 3NF(2) generates less schemata than 3NF(1) as a canonical cover contains less FDs than the other cover considered. The difference between BCNF(1) and 3NF(2) tells us how many more schemata are required to guarantee dependency-preservation. The results suggest to prefer BCNF decomposition in terms of query efficiency and, obviously, in terms of elimination of data redundancy. They do not mean to prefer BCNF decomposition in terms of update efficiency. Indeed, BCNF decompositions are only more efficient for updates of redundant data value occurrences caused by FDs preserved during the decomposition, while update inefficiencies result from having to join schemata to validate FDs that were not preserved during the decomposition.

The graphs further show that the bigger the size of the input relation schema, the bigger the size of the decomposition - observing the graphs on the left and that on the right. However, the story is different in terms of growing FD size. When there are about as many given FDs as underlying attributes, then the average size of decompositions grows linearly. When there are significantly more given FDs than underlying attributes, then a global maximum is observable in the average size of decompositions. This is explained by the fact that there will be a point when there are sufficiently many FDs that turn attribute sets into keys, in which case further decompositions become unnecessary. Not surprisingly, this saturation point comes earlier for BCNF(1) than for any of the other algorithms considered. For larger FD

	α_{k+1-i}-lossless	Free from α_{k+1-i} data redundancy	β_i-dependency-preserving
β_i-BCNF	✔	✔	
β_i-3NF	✔		✔

Table 3: Achievements of β-BCNF and β-3NF

sets, higher levels of data integrity achieve the same levels of query efficiency as significantly lower levels of data integrity.

The bottom of Figure 4 shows the average times to generate the decompositions for our two experiments. As observed for the average size of decompositions, both experiments show nearly uniform results for the average times: BCNF(2) takes the lowest time, followed by BCNF(3), followed by BCNF(1), and followed by 3NF(1,2). These results are intuitive, too. BCNF(2) demonstrates its major strength in comparison to the other algorithms, with a maximum average time of 8ms for $a = 1$ and 13ms for $a = 5$. BCNF(3) requires slightly more time with a maximum average time of 12ms for $a = 1$ and 35ms for $a = 5$, resulting from the projections of the original FD sets to the final output schemata at the end. BCNF(1) requires more time with a maximum average time of 630ms for $a = 1$ and 627ms for $a = 5$, as a result of projecting the original FD sets to all intermediate and final schemata. Finally, 3NF(1,2) require significantly more time with a maximum average time of 64s for $a = 1$ and 530s for $a = 5$.

Two main observations illustrate inherent trade-offs. For BCNF decompositions firstly, smaller-sized decompositions come at the price of requiring more time to be generated. In other words, polynomial-time BCNF decompositions result from not having to spend time on validating the BCNF condition, thereby generating additional schemata that are superfluous in terms of update efficiency but may significantly add to the inefficiency of query evaluation. For 3NF decompositions secondly, the price to pay for dependency-preservation is a higher complexity in terms of the size of the decompositions and the time taken to generate them.

10. CONCLUSION AND FUTURE WORK

We developed a full framework to design relational schemata for applications with uncertain data. The fact that FDs are closed downwards, i.e. hold on every sub-relation of a relation that satisfies them, enabled us to introduce different levels of data redundancy. We established a customized segmentation of the well-known Boyce-Codd and Third normal forms, extending all of their properties to each segment, Table 3. Certainty degrees of FDs provide a mechanism for organizations to control the levels of data integrity, data losslessness, query efficiency, and update efficiency, Figure 1.

Several problems warrant future research. As relational schema design for certain data depends on the identification of FDs that are meaningful for the given application, schema design for uncertain data depends on the identification of meaningful pFDs. This motivates research about Armstrong relations [11] in the possibilistic setting. Further normal forms should be customized to their use for uncertain data, including 4NF [10, 27] as well as Inclusion Dependency normal form [20]. Recently, the relational normalization toolbox has been extended to incomplete relations [15, 16, 17, 18]. It would be interesting to develop a normalization framework for possibilistic incomplete relations.

11. REFERENCES

[1] W. W. Armstrong. Dependency structures of data base relationships. In *IFIP Congress*, pages 580–583, 1974.

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

[3] O. Benjelloun, A. D. Sarma, A. Y. Halevy, M. Theobald, and J. Widom. Databases with uncertainty and lineage. *VLDB J.*, 17(2):243–264, 2008.

[4] P. A. Bernstein. Synthesizing third normal form relations from functional dependencies. *ACM TODS*, 1(4):277–298, 1976.

[5] N. A. Chaudhry, J. R. Moyne, and E. A. Rundensteiner. An extended database design methodology for uncertain data management. *Inf. Sci.*, 121(1-2):83–112, 1999.

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

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

[8] N. N. Dalvi and D. Suciu. Management of probabilistic data: foundations and challenges. In *PODS*, pages 1–12, 2007.

[9] D. Dubois and H. Prade. Possibility theory. In R. A. Meyers, editor, *Computational Complexity*, pages 2240–2252. Springer New York, 2012.

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

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

[12] N. Hall, H. Köhler, S. Link, H. Prade, and X. Zhou. Cardinality constraints on qualitatively uncertain data. *Data Knowl. Eng.*, 99:126–150, 2015.

[13] A. K. Jha and D. Suciu. Probabilistic databases with markoViews. *PVLDB*, 5(11):1160–1171, 2012.

[14] H. Köhler, U. Leck, S. Link, and H. Prade. Logical foundations of possibilistic keys. In *JELIA*, pages 181–195, 2014.

[15] H. Köhler, U. Leck, S. Link, and X. Zhou. Possible and certain keys for SQL. *VLDB J.*, 25(4):571–596, 2016.

[16] H. Köhler and S. Link. SQL schema design: Foundations, normal forms, and normalization. In *SIGMOD*, pages 267–279, 2016.

[17] H. Köhler, S. Link, and X. Zhou. Possible and certain SQL keys. *PVLDB*, 8(11):1118–1129, 2015.

[18] H. Köhler, S. Link, and X. Zhou. Discovering meaningful certain keys from incomplete and inconsistent relations. *IEEE Data Eng. Bull.*, 39(2):21–37, 2016.

[19] S. Kolahi and L. Libkin. An information-theoretic analysis of worst-case redundancy in database design. *ACM TODS*, 35(1), 2010.

[20] M. Levene and M. W. Vincent. Justification for inclusion dependency normal form. *IEEE TKDE*, 12(2):281–291, 2000.

[21] S. Link and H. Prade. Relational database schema design for uncertain data. Technical Report CDMTCS-469, The University of Auckland, 2014.

[22] S. Link and H. Prade. Possibilistic functional dependencies and their relationship to possibility theory. *IEEE Trans. Fuzzy Systems*, 24(3):757–763, 2016.

[23] J. Rissanen. Independent components of relations. *ACM TODS*, 2(4):317–325, 1977.

[24] A. D. Sarma, J. D. Ullman, and J. Widom. Schema design for uncertain databases. In *AMW*, 2009.

[25] D. Suciu, D. Olteanu, C. Ré, and C. Koch. *Probabilistic Databases*. Morgan & Claypool Publishers, 2011.

[26] D.-M. Tsou and P. C. Fischer. Decomposition of a relation scheme into Boyce-Codd normal form. *SIGACT News*, 14(3):23–29, 1982.

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