Logical Foundations of Possibilistic Keys

Henning Koehler[1], Uwe Leck[2], Sebastian Link[3], and Henri Prade[4]

[1] School of Engineering & Advanced Technology, Massey University, New Zealand
[2] University of Wisconsin-Superior, Superior, WI, U.S.A.
[3] Department of Computer Science, University of Auckland, New Zealand
[4] IRIT, CNRS and Université de Toulouse III, France
h.koehler@massey.ac.nz,uleck@uwsuper.edu,
s.link@auckland.ac.nz, prade@irit.fr

Abstract. Possibility theory is applied to introduce and reason about the fundamental notion of a key for uncertain data. Uncertainty is modeled qualitatively by assigning to tuples of data a degree of possibility with which they occur in a relation, and assigning to keys a degree of certainty which says to which tuples the key applies. The associated implication problem is characterized axiomatically and algorithmically. It is shown how sets of possibilistic keys can be visualized as possibilistic Armstrong relations, and how they can be discovered from given possibilistic relations. It is also shown how possibilistic keys can be used to clean dirty data by revising the belief in possibility degrees of tuples.

Keywords: Armstrong relation, Axiomatization, Database, Data cleaning, Data mining, Implication, Key, Possibility theory, Query processing, Uncertain Data.

1 Introduction

Background. The notion of a key is fundamental for understanding the structure and semantics of data. For relational databases, keys were already introduced in Codd's seminal paper [10]. Here, a key is a set of attributes that holds on a relation if there are no two different tuples in the relation that have matching values on all the attributes of the key. Keys uniquely identify tuples of data, and have therefore significant applications in data cleaning, integration, modeling, processing, and retrieval.

Motivation. Relational databases were developed for applications with certain data, such as accounting, inventory and payroll. Modern applications, such as information extraction, radio-frequency identification (RFID) and scientific data management, data cleaning and financial risk assessment produce large volumes of uncertain data. For instance, RFID is used to track movements of endangered species of animals, such as Grizzly Bears. For such an application it is desirable to associate degrees of possibility (p-degrees) with which tuples occur in a relation. Here, p-degrees represent the trust in the RFID readings, which can be derived from the strength, or precision of the devices that send and receive the signals. Table 1 shows a possibilistic relation (p-relation), where each tuple is associated with an element of a finite scale of p-degrees: $\alpha_1 > \ldots > \alpha_{k+1}$. The top degree α_1 is reserved for tuples that are 'fully possible', the bottom degree α_{k+1} for tuples that are 'impossible' to occur. Intermediate degrees can

E. Fermé and J. Leite (Eds.): JELIA 2014, LNAI 8761, pp. 181–195, 2014.

be used, such as 'quite possible' (α_2), 'medium possible' (α_3) and 'somewhat possible' (α_4) when some linguistic interpretation is preferred.

The p-degrees enable us to express keys with different degrees of certainty. For example, to express that it is 'somewhat possible' that the same grizzly is in different zones within an hour we declare the key {*time*,*rfid*} to be 'quite certain', stipulating that no two distinct tuples are at least 'medium possible' and have matching values on *time* and *rfid*. Similarly, to say that it is 'quite possible' that different grizzlies are in the same zone at the same time we declare the key {*zone*,*time*} to be 'somewhat certain', stipulating that no two distinct tuples are at least 'quite possible' and have matching values on *zone* and *time*. We apply possibility theory to establish possibilistic keys (PKs) as a fundamental notion to identify tuples of uncertain data.

Table 1. A Possibilistic Relation and Its Nested Chain of Possible Worlds

Possibilistic Relation

zone	*time*	*rfid*	*object*	*p-degree*
Z0	10am	H0	Grizzly	α_1
Z1	10am	H1	Grizzly	α_1
Z1	12pm	H2	Grizzly	α_1
Z3	1pm	H2	Grizzly	α_1
Z3	1pm	H3	Grizzly	α_2
Z3	3pm	H3	Grizzly	α_3
Z4	3pm	H3	Grizzly	α_4

Worlds of Possibilistic Relation

	zone	*time*	*rfid*	*object*
	Z0	10am	H0	Grizzly
	Z1	10am	H1	Grizzly
	Z1	12pm	H2	Grizzly
r_1	Z3	1pm	H2	Grizzly
r_2	Z3	1pm	H3	Grizzly
r_3	Z3	3pm	H3	Grizzly
r_4	Z4	3pm	H3	Grizzly

Contributions. (1) In Section 2 we point out the lack of qualitative approaches to constraints on uncertain data. (2) We define a semantics for keys on uncertain relations in Section 3. Here, uncertainty is modeled qualitatively by degrees of possibility. The degrees bring forward a nested chain of possible worlds, with each being a classical relation that has some possibility. Hence, the more possible the smaller a relation is, and the more keys can identify tuples uniquely. For example, the possible worlds of the p-relation from Table 1 are shown in Figure 1. The key {*time*,*rfid*} is satisfied by r_3 but not by r_4, and {*zone*,*time*} is satisfied by r_1 but not by r_2. (3) In Section 4 we establish axiomatic and linear-time algorithmic characterizations for the implication problem of PKs. (4) We show in Section 5 how to visualize PK sets as a single Armstrong p-relation. That is, for any given PK set Σ we compute a p-relation that satisfies any given PK φ if and only if φ is implied by Σ. While the problem of finding an Armstrong p-relation is precisely exponential, our output p-relation is always at most quadratic in the size of a minimum-sized Armstrong p-relation. (5) Using hypergraph transversals, we show in Section 5 how to discover the PKs that hold on a given p-relation. Visualization and discovery provide a communication framework for data engineers and domain experts to jointly acquire the set of PKs that are semantically meaningful for a given application. (6) In Section 6 we apply PKs to clean dirty data, and to query processing in Section 7. (7) In Section 8 we conclude and briefly discuss future work.

2 Related Work

The application of possibilistic logic to keys empowers applications to reason qualitatively about the uniqueness of tuples consisting of uncertain data. Data cleaning, data fusion and uncertain databases are thus primary impact areas. Section 6 illustrates how possibilistic keys can clean dirty data by revising the beliefs in p-degrees of tuples. Possibilistic keys are soft constraints that data shall satisfy after their integration from different sources. In this sense, data engineers can apply possibilistic keys as a means to impose solutions to the correlation problem, which aims to establish whether some information pertains to the same object or different ones [21]. For example, by declaring the key $\{$*time*,*rfid*$\}$ on tuples that are at least medium possible, information about the same grizzly (*rfid*) at the same *time* is only recorded once in tuples that are at least medium possible (e.g. come from sufficiently trusted sources), while the key may be violated when tuples are present that are only somewhat possible.

Work on quantitative approaches to reason about uncertain data is huge, foremost probability theory [37]. The only study of keys on probabilistic databases we are aware of is [28], which is exclusively focused on query optimization. Qualitative approaches to uncertain data deal with either query languages or extensions of functional dependencies (FDs), with surveys found in [6,7] for example. Qualitative approaches to identify tuples of uncertain data have not been studied yet to the best of our knowledge. In particular, the notion of a possibilistic key is new. The only paper that considers schema design for uncertain databases is [35]. The authors develop an "FD theory for data models whose basic construct for uncertainty is alternatives" [35]. Their work is fundamentally different from our approach. Keys and FDs have also been included in description logic research [8,29,39], but have not been investigated yet for uncertain data.

Our contributions extend results on keys from classical relations, covered by the special case of two possibility degrees where $k = 1$. These include results on the implication problem [1,12], Armstrong relations [2,18,23,30] and the discovery of keys from relations [27,31], as well as data cleaning [4,9]. Keys have also been considered in other data models, including incomplete relations [24,38] and XML data [25,26]. Note that Armstrong relations are also an AI tool to acquire and reason about conditional independencies [20,32].

Possibilistic logic is a well-established tool for reasoning about uncertainty [13,16] with numerous applications in artificial intelligence [15], including approximate reasoning [40], non-monotonic reasoning [19], qualitative reasoning [36], belief revision [14,22,33], soft constraint satisfaction problems [5], decision-making under uncertainty [34], pattern classification and preferences [3]. Our results show that possibilistic logic is an AI framework that is suitable to extend the classical notion of a key from certain to uncertain data.

3 Possibilistic Keys

In this section we extend the classical relational model of data to model uncertain data qualitatively.

A relation schema, denoted by R, is a finite non-empty set 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 $\prod_{a \in R} dom(a)$ of the attributes' domains. 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 TRACKING with attributes *zone*, *time*, *rfid*, *object* from before. Tuples either belong or do not belong to a relation. For example, we cannot express that we have less confidence for the Grizzly identified by *rfid* value $H3$ to be in *zone* $Z3$ at $1pm$ than for the Grizzly identified by $H2$.

We model uncertain relations by assigning to each tuple some degree of possibility with which the tuple occurs in a relation. Formally, we have a *scale of possibility*, that is, a finite strict linear order $\mathcal{S} = (S, <)$ with $k + 1$ elements, denoted by $\alpha_1 > \cdots > \alpha_k > \alpha_{k+1}$. The elements $\alpha_i \in S$ are called *possibility degrees*, or p-degrees. The top p-degree α_1 is reserved for tuples that are 'fully possible' to occur in a relation, while the bottom p-degree α_{k+1} is reserved for tuples that are 'impossible' to occur. Humans like to use simple scales in everyday life to communicate, compare, or rank. Simple means to classify items qualitatively, rather than quantitatively by putting a precise value on it. Classical relations use two p-degrees, that is $k = 1$.

A *possibilistic relation schema* $(R, \mathcal{S})$, or p-relation schema, consists of a relation schema R and a possibility scale $\mathcal{S}$. A *possibilistic relation*, or p-relation, over $(R, \mathcal{S})$ consists of a relation r over R, and a function $Poss_r$ that assigns to each tuple $t \in r$ a p-degree $Poss_r(t) \in \mathcal{S}$. Table 1 shows a p-relation over (TRACKING, $\mathcal{S} = \{\alpha_1, \ldots, \alpha_5\}$).

P-relations enjoy a possible world semantics. For $i = 1, \ldots, k$ let r_i consist of all tuples in r that have p-degree at least α_i, that is, $r_i = \{t \in r \mid Poss_r(t) \geq \alpha_i\}$. 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 a possible world, since its possibility $\pi(r_{k+1}) = \alpha_{k+1}$ means 'impossible'. Vice versa, the possibility $Poss_r(t)$ of a tuple $t \in r$ is the maximum possibility $\max\{\alpha_i \mid t \in r_i\}$ of a world to which t belongs. If $t \notin r_k$, then $Poss_r(t) = \alpha_{k+1}$. Every tuple that is 'fully possible' occurs in every possible world, and is therefore also 'fully certain'. Hence, relations are a special case of uncertain relations. Figure 1 shows the possible worlds $r_1 \subsetneq r_2 \subsetneq r_3 \subsetneq r_4$ of the p-relation of Table 1.

We introduce possibilistic keys, or PKs, as keys with some degree of certainty. As keys are fundamental to applications with certain data, PKs will serve a similar role for application with uncertain data. A *key* $K \subseteq R$ is satisfied by a relation r over R, denoted by $\models_r K$, if there are no distinct tuples $t, t' \in r$ with matching values on all the attributes in K. For example, the key {*time, object*} is not satisfied by any relation $r_1, \ldots, r_4$. The key {*zone, time*} is satisfied by r_1, but not by r_2. The key {*zone, rfid*} is satisfied by r_2, but not by r_3. The key {*time, rfid*} is satisfied by r_3, but not by r_4. The key {*zone, time, rfid*} is satisfied by r_4.

The p-degrees of tuples result in degrees of certainty with which keys hold. Since {*zone, time, rfid*} holds in every possible world, it is fully certain to hold on r. As {*time, rfid*} is only violated in a somewhat possible world r_4, it is quite certain to hold on r. Since the smallest relation that violates {*zone, rfid*} is the medium possible world r_3, it is medium certain to hold on r. As the smallest relation that violates {*zone, time*} is the quite possible world r_2, it is somewhat certain to hold on r. Since $\{$*time, object*$\}$ is violated in the fully possible world r_1, it is not certain at all to hold on r.

Similar to a scale $\mathcal{S}$ of p-degrees for tuples we use a scale $\mathcal{S}^T$ of certainty degrees, or c-degrees, for keys. We use subscripted versions of the Greek letter β to denote c-degrees. 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 $i = 1, \ldots, k+1$. Hence, the certainty $C_r(K)$ with which the key K holds on the uncertain relation r is either the top degree β_1 if K is satisfied by r_k, or the minimum amongst the c-degrees β_{k+2-i} that correspond to possible worlds r_i in which K is violated, that is,

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

We can now define the semantics of possibilistic keys.

Definition 1. *Let $(R, \mathcal{S})$ denote a p-relation schema. A possibilistic key (PK) over $(R, \mathcal{S})$ is an expression (K, β) where $K \subseteq R$ and $\beta \in \mathcal{S}^T$. A p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ satisfies the PK (K, β) if and only if $C_r(K) \geq \beta$.* □

Example 1. The p-relation from Table 1 satisfies the PK set Σ consisting of

- $(\{$*zone, time, rfid*$\}, \beta_1)$,
- $(\{$*time, rfid*$\}, \beta_2)$,
- $(\{$*zone, rfid*$\}, \beta_3)$, and
- $(\{$*zone, time*$\}, \beta_4)$.

It violates the PK $(\{$*zone, rfid*$\}, \beta_2)$ since $C_r(\{$*zone, rfid*$\}) = \beta_3 < \beta_2$.

4 Reasoning Tools

First, we establish a strong correspondence between the implication of PKs and keys. Let $\Sigma \cup \{\varphi\}$ denote a set of PKs over $(R, \mathcal{S})$. We say Σ *implies* φ, denoted by $\Sigma \models \varphi$, if every p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ that satisfies every PK in Σ also satisfies φ. We use $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ to denote the *semantic closure* of Σ.

Example 2. Let Σ be as in Example 1, and $\varphi = (\{$*zone*, *rfid*, *object*$\}, \beta_2)$. Then Σ does not imply φ as the following p-relation witnesses:

zone	*time*	*rfid*	*object*	Poss. degree
Z0	10am	H0	Grizzly	α_1
Z0	3pm	H0	Grizzly	α_3

4.1 The Magic of β-Cuts

For a PK set Σ over $(R, \mathcal{S})$ with $|\mathcal{S}| = k+1$ and c-degree $\beta \in \mathcal{S}^T$ where $\beta > \beta_{k+1}$, let $\Sigma_\beta = \{K \mid (K, \beta') \in \Sigma \text{ and } \beta' \geq \beta\}$ be the β-*cut* of Σ.

Theorem 1. *Let $\Sigma \cup \{(K, \beta)\}$ be a PK set over $(R, \mathcal{S})$ where $\beta > \beta_{k+1}$. Then $\Sigma \models (K, \beta)$ if and only if $\Sigma_\beta \models K$.*

Proof. Suppose $(r, Poss_r)$ is some p-relation over $(R, \mathcal{S})$ that satisfies Σ, but violates (K, β). In particular, $C_r(K) < \beta$ implies that there is some relation r_i that violates K and where $\beta_{k+2-i} < \beta$. Let $K' \in \Sigma_\beta$, where $(K', \beta') \in \Sigma$. Since r satisfies $(K, \beta') \in \Sigma$ we have $C_r(K') \geq \beta' \geq \beta$. If r_i violated K', then $\beta > \beta_{k+2-i} \geq C_r(K') \geq \beta$, a contradiction. Hence, r_i satisfies Σ_β and violates K.

Let r' denote some relation that satisfies Σ_β and violates K, w.l.o.g. $r' = \{t, t'\}$. Let r be the p-relation over $(R, \mathcal{S})$ that consists of r' and where $Poss_{r'}(t) = \alpha_1$ and $Poss_{r'}(t') = \alpha_i$, such that $\beta_{k+1-i} = \beta$. Then r violates (K, β) since $C_r(K) = \beta_{k+2-i}$, as $r_i = r'$ is the smallest relation that violates K, and $\beta_{k+2-i} < \beta_{k+1-i} = \beta$. For $(K', \beta') \in \Sigma$ we distinguish two cases. If r_i satisfies K', then $C_r(K') = \beta_1 \geq \beta$. If r_i violates K', then $K' \notin \Sigma_\beta$, i.e., $\beta' < \beta = \beta_{k+1-i}$. Therefore, $\beta' \leq \beta_{k+2-i} = C_r(K')$ as $r_i = r'$ is the smallest relation that violates K'. We conclude that $C_r(K') \geq \beta'$. Consequently, $(r, Poss_r)$ is a p-relation that satisfies Σ and violates (K, β). □

Example 3. Let $\Sigma \cup \{\varphi\}$ be as in Example 2. Theorem 1 says that Σ_{β_2} does not imply $(\{zone, rfid, object\}, \beta_2)$. The possible world r_3 of the p-relation from Example 2:

zone	*time*	*rfid*	*object*
Z0	10am	H0	Grizzly
Z0	3pm	H0	Grizzly

satisfies the key $\{$*time, rfid*$\}$ that implies both keys in Σ_{β_2}. However, r_3 violates the key $\{$*zone*, *rfid*, *object*$\}$.

4.2 Axiomatic Characterization

We determine the semantic closure by applying *inference rules* of the form

$$\frac{\text{premise}}{\text{conclusion}}\ \text{condition}\ .$$

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 $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be the *syntactic closure* of Σ under inferences by $\mathfrak{R}$. $\mathfrak{R}$ is *sound* (*complete*) if for every set Σ over every $(R, \mathcal{S})$ we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is a (finite) *axiomatization* if $\mathfrak{R}$ is both sound and complete.

For the set $\mathfrak{K}$ from Table 2 the attribute sets K, K' are subsets of a given R, and β, β' belong to a given $\mathcal{S}^T$. In particular, β_{k+1} denotes the bottom certainty degree.

Theorem 2. *The set $\mathfrak{K}$ forms a finite axiomatization for the implication problem of PKs.*

Proof. The soundness proof is straightforward and omitted. For completeness, we apply Theorem 1 and the fact that $\mathcal{K}'$ axiomatizes key implication. Let $(R, \mathcal{S})$ be a p-relation schema with $|\mathcal{S}| = k + 1$, and $\Sigma \cup \{(K, \beta)\}$ a PK set such that $\Sigma \models (K, \beta)$. We show that $\Sigma \vdash_{\mathfrak{K}} (K, \beta)$ holds.

For $\Sigma \models (K, \beta_{k+1})$ we have $\Sigma \vdash_{\mathfrak{K}} (K, \beta_{k+1})$ by applying $\mathcal{B}$. Let now $\beta < \beta_{k+1}$. From $\Sigma \models (K, \beta)$ we conclude $\Sigma_\beta \models K$ by Theorem 1. Since $\mathfrak{K}'$ is complete for key

Table 2. Axiomatization $\mathfrak{K}' = \{\mathcal{T}', \mathcal{S}'\}$ of Keys and $\mathfrak{K} = \{\mathcal{T}, \mathcal{S}, \mathcal{B}, \mathcal{W}\}$ of Possibilistic Keys

$$\frac{}{R}\ \text{(top, }\mathcal{T}'\text{)} \qquad \frac{K}{K \cup K'}\ \text{(superkey, }\mathcal{S}'\text{)}$$

$$\frac{}{(R, \beta)}\ \text{(top, }\mathcal{T}\text{)} \qquad \frac{}{(K, \beta_{k+1})}\ \text{(bottom, }\mathcal{B}\text{)} \qquad \frac{(K, \beta)}{(K \cup K', \beta)}\ \text{(superkey, }\mathcal{S}\text{)} \qquad \frac{(K, \beta)}{(K, \beta')}\ \beta' \leq \beta\ \text{(weakening, }\mathcal{W}\text{)}$$

implication, $\Sigma_\beta \vdash_{\mathfrak{K}'} K$ holds. Let $\Sigma_\beta^\beta = \{(K', \beta) \mid K' \in \Sigma_\beta\}$. Thus, the inference of K from Σ_β using $\mathcal{K}'$ can be turned into an inference of (K, β) from Σ_β^β by $\mathfrak{K}$, simply by adding β to each key in the inference. Hence, whenever $\mathcal{T}'$ or $\mathcal{S}'$ is applied, one applies instead $\mathcal{T}$ or $\mathcal{S}$, respectively. Consequently, $\Sigma_\beta^\beta \vdash_{\mathfrak{K}} (K, \beta)$. The definition of Σ_β^β ensures that every PK in Σ_β^β can be inferred from Σ by applying $\mathcal{W}$. Hence, $\Sigma_\beta^\beta \vdash_{\mathfrak{K}} (K, \beta)$ means that $\Sigma \vdash_{\mathfrak{K}} (K, \beta)$. □

4.3 Algorithmic Characterization

While $\mathfrak{K}$ enables us to enumerate all PKs implied by a PK set Σ, in practice it often suffices to decide whether a given PK φ is implied by Σ. Enumerating all implied PKs and checking whether φ is among them is neither efficient nor makes good use of φ.

Theorem 3. *Let $\Sigma \cup \{(K, \beta)\}$ denote a set of PKs over $(R, \mathcal{S})$ with $|\mathcal{S}| = k + 1$. Then Σ implies (K, β) if and only if $\beta = \beta_{k+1}$, or $K = R$, or there is some $(K', \beta') \in \Sigma$ such that $K' \subseteq K$ and $\beta' \geq \beta$.*

Proof. Theorem 1 shows for $i = 1, \ldots, k$ that Σ implies (K, β_i) if and only if Σ_β implies K. It is easy to observe from the axiomatization $\mathfrak{K}'$ of keys that Σ_β implies K if and only if $R = K$, or there is some $K' \in \Sigma_\beta$ such that $K' \subseteq K$ holds. As Σ implies (K, β_{k+1}), the theorem follows. □

Corollary 1. *An instance $\Sigma \models \varphi$ of the implication problem can be decided in time $\mathcal{O}(||\Sigma \cup \{\varphi\}||)$ where $||\Sigma||$ denotes the total number of symbol occurrences in Σ.* □

5 Acquisition Tools

New applications benefit from the ability of data engineers to acquire the PKs that are semantically meaningful in the domain of the application. For that purpose, data engineers communicate with domain experts. Now we establish two major tools that help data engineers to effectively communicate with domain experts. We follow the framework in Figure 1. Here, data engineers use our algorithm to visualize abstract PK sets Σ in form of some Armstrong p-relation r_Σ, which is then inspected jointly with domain experts. Domain experts may change r_Σ or supply entirely new data samples to the engineers. For that case we establish an algorithm that computes the set of PKs that hold in the data sample.

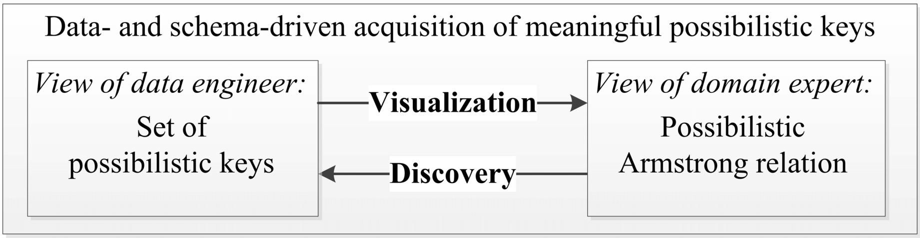

Fig. 1. Acquisition Framework for Possibilistic Keys

5.1 Structure and Computation of Visualizations

A p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ is *Armstrong* for a PK set Σ if and only if for all PKs φ over $(R, \mathcal{S})$, $(r, Poss_r)$ satisfies φ if and only if $\Sigma \models \varphi$. The maximum c-degree β by which a PK (K, β) is implied by Σ can 'simply be read-off' as the c-degree $C_r(K)$ of any Armstrong p-relation $(r, Poss_r)$ for Σ. Our first aim is to characterize the structure of Armstrong p-relations. We recall two notions from relational databases. The *agree set* of two tuples t, t' over R is the set $ag(t, t') = \{a \in R \mid t(a) = t'(a)\}$ of attributes on which t and t' have matching values. The agree set of a relation is the set $ag(r) = \{ag(t, t') \mid t, t' \in r \wedge t \neq t'\}$. Let Σ denote a set of keys over relation schema R. An *anti-key* of R with respect to Σ is a subset $A \subseteq R$ such that Σ does not imply the key A over R and for all $a \in R - A$, Σ implies the key $A \cup \{a\}$ over R. We denote by Σ^{-1} the set of all anti-keys of R with respect to Σ.

Theorem 4. *Let Σ denote a set of PKs, and let $(r, Poss_r)$ denote a p-relation over $(R, \mathcal{S})$ with $|\mathcal{S}| = k + 1$. Then $(r, Poss_r)$ is Armstrong for Σ if and only if for all $i = 1, \ldots, k$, the relation r_{k+1-i} is Armstrong for Σ_{β_i}. That is, for all $i = 1, \ldots, k$, $\Sigma_{\beta_i}^{-1} \subseteq ag(r_{k+1-i})$, and for all $K \in \Sigma_{\beta_i}$ and for all $X \in ag(r_{k+1-i})$, $K \nsubseteq X$.*

Proof. $(r, Poss_r)$ is Armstrong for Σ if and only if for all $i = 1, \ldots, k$, for all $K \subseteq R$, $\models_{(r,Poss_r)} (K, \beta_i)$ iff $\Sigma \models (K, \beta_i)$. However, $\models_{(r,Poss_r)} (K, \beta_i)$ iff $\models_{r_{k+1-i}} K$, and $\Sigma \models (K, \beta_i)$ iff $\Sigma_{\beta_i} \models K$. Therefore, $(r, Poss_r)$ is Armstrong for Σ if and only if for all $i = 1, \ldots, k$, r_{k+1-i} is an Armstrong relation for Σ_{β_i}. The second statement follows straight from the well-known result that a relation r is Armstrong for a set Σ of keys if and only if $\Sigma^{-1} \subseteq ag(r)$ and for all $K \in \Sigma$ and all $X \in ag(r)$, $K \nsubseteq X$ [11]. □

Theorem 4 shows that Algorithm 1 computes an Armstrong p-relation for input Σ. The algorithm computes for $i = 1, \ldots, k$ the set $\Sigma_{\beta_i}^{-1}$ incrementally. Starting with a tuple of p-degree α_1, for $i = k, \ldots, 1$, each $A \in \Sigma_{\beta_i}^{-1}$ is realized as an agree set by introducing a tuple that agrees with the previous tuple on A and has p-degree α_{k+1-i}, as long as A did not already occur for some larger i.

Example 4. We apply Algorithm 1 to the set Σ from Example 1. Using the first letters of each attribute we obtain

Algorithm 1. Visualize

Input: $R, \{\beta_1, \ldots, \beta_k\}, \Sigma$
Output: Possibilistic Armstrong Relation $(r, Poss_r)$ for Σ

$\Sigma_0^{-1} \leftarrow \{R - \{a\} \mid a \in R\}$;
for $i = 1, \ldots, k$ **do** ▷ Compute $\Sigma_{\beta_i}^{-1}$ incrementally
    $\Sigma_i \leftarrow \{K \mid (K, \beta_j) \in \Sigma$ and $j \leq i\}$,
    $\Sigma_i^{-1} \leftarrow$ ANTIKEYS$(R, \Sigma_i, \Sigma_{i-1}^{-1})$,
end for
for all $a \in R$ **do**
    $t_0(a) \leftarrow c_{a,0}$; ▷ $c_{a,i}$ are fresh constants
end for
$j \leftarrow 0$; $r \leftarrow \{t_0\}$; $Poss_r(t_0) \leftarrow \alpha_1$; $\Sigma_0 \leftarrow \emptyset$;
for $i = k$ downto 1 **do**
    for all $A \in \Sigma_i^{-1} - \Sigma_0$ **do**
        $j \leftarrow j + 1$;
        for all $a \in R$ **do** ▷ New tuple with agree set A
            if $a \in A$ **then** $t_j(a) \leftarrow t_{j-1}(a)$;
            else $t_j(a) \leftarrow c_{a,j}$;
            end if
        end for
        $Poss_r(t_j) \leftarrow \alpha_{k+1-i}$; ▷ and p-degree α_{k+1-i}
        $r \leftarrow r \cup \{t_j\}$;
    end for
    $\Sigma_0 \leftarrow \Sigma_0 \cup \Sigma_i^{-1}$;
end for
return $(r, Poss_r)$;

Subroutine ANTIKEYS(R, Σ, Σ^{-1})
Input: R, Σ set of keys in Σ_i, Σ^{-1} set of anti-keys in Σ_{i-1}^{-1}
Output: Σ^{-1} set of anti-keys for Σ_{β_i}
for all $K \in \Sigma$, $A \in \Sigma^{-1}$ with $K \subseteq A$ **do**
    $\Sigma^{-1} \leftarrow (\Sigma^{-1} - \{A\}) \cup \bigcup_{a \in K}\{A - \{a\}\}$;
end for
$\Sigma^{-1} \leftarrow \{A \mid \forall B \in \Sigma^{-1} - \{A\}(A \not\subseteq B)\}$;
return Σ^{-1};

- $\Sigma_1 = \{ztr\}$ and $\Sigma_{\beta_1}^{-1} = \{zto, \underline{tro}, zro\}$
- $\Sigma_2 = \{tr\}$ and $\Sigma_{\beta_2}^{-1} = \{zto, \underline{zro}\}$
- $\Sigma_3 = \{zr\}$ and $\Sigma_{\beta_3}^{-1} = \{\underline{zto}, ro, zo\}$, and
- $\Sigma_4 = \{zt\}$ and $\Sigma_{\beta_4}^{-1} = \{\underline{to}, \underline{zo}, \underline{ro}\}$.

Anti-keys are underlined when they are realized as agree sets of tuples in the possibilistic Armstrong relation:

zone	*time*	*rfid*	*object*	*Poss. degree*
$c_{z,0}$	$c_{t,0}$	$c_{R,0}$	$c_{o,0}$	α_1
$c_{z,1}$	$c_{t,0}$	$c_{R,1}$	$c_{o,0}$	α_1
$c_{z,1}$	$c_{t,2}$	$c_{R,2}$	$c_{o,0}$	α_1
$c_{z,3}$	$c_{t,3}$	$c_{R,2}$	$c_{o,0}$	α_1
$c_{z,3}$	$c_{t,3}$	$c_{R,4}$	$c_{o,0}$	α_2
$c_{z,3}$	$c_{t,5}$	$c_{R,4}$	$c_{o,0}$	α_3
$c_{z,6}$	$c_{t,5}$	$c_{R,4}$	$c_{o,0}$	α_4

Fitting substitution yields the p-relation from Table 1.

Theorem 5. *Algorithm 1 computes an Armstrong p-relation for Σ whose size is at most quadratic in that of a minimum-sized Armstrong p-relation for Σ.*

Proof. The soundness of Algorithm 1 follows from Theorem 4, which also shows that for $\Sigma^{-1} = \bigcup_{i=1}^{k} \Sigma_i^{-1}$ we have $|\Sigma^{-1}| \leq ag(r) \leq \binom{|r|}{2}$. The inequalities establish the lower bound in $\frac{1}{2} \cdot \sqrt{1 + 8 \cdot |\Sigma^{-1}|} \leq |r| \leq |\Sigma^{-1}| + 1$. The upper bound follows from Algorithm 1. Hence, the p-relation computed by Algorithm 1 is at most quadratic in the size of a minimum-sized Armstrong p-relation for Σ. □

Finding Armstrong p-relations is precisely exponential. That means that there is an algorithm for computing an Armstrong p-relation whose running time is exponential in the size of Σ, and that there is some set Σ in which the number of tuples in each minimum-sized Armstrong p-relation for Σ is exponential thus, an exponential amount of time is required in this case simply to write down the p-relation.

Theorem 6. *Finding an Armstrong p-relation for a PK set Σ is precisely exponential in the size of Σ.*

Proof. Algorithm 1 computes an Armstrong p-relation for Σ in time at most exponential in its size. Some PK sets Σ have only Armstrong p-relations with exponentially many tuples in the size of Σ. For $R = \{a_1, \ldots, a_{2n}\}$, $\mathcal{S} = \{\alpha_1, \alpha_2\}$ and $\Sigma = \{(\{a_1, a_2\}, \beta_1), \ldots, (\{a_{2n-1}, a_{2n}\}, \beta_1)\}$ with size $2 \cdot n$, Σ^{-1} consists of the 2^n anti-keys $\bigcup_{j=1}^{n} X_j$ where $X_j \in \{a_{2j-1}, a_{2j}\}$. □

Armstrong p-relations for some other PK sets Σ' only require a number of tuples that is logarithmic in the size of Σ'. Such a set Σ' is given by the 2^n PKs $(\bigcup_{j=1}^{n} X_j, \beta_1)$ where $X_j \in \{a_{2j-1}, a_{2j}\}$. In fact, Algorithm 1 computes an Armstrong p-relation for Σ' with $n + 1$ tuples.

5.2 Discovery

Given a p-relation we may ask for which set Σ it is Armstrong. Algorithm 2 computes a cover Σ of the set of PKs satisfied by a given p-relation. A cover of some PK set Θ is a PK set Σ where $\Sigma^* = \Theta^*$. A *hypergraph* (V, E) consists of a vertex set V and a set E of subsets of V, called hyperedges. A set $T \subseteq V$ is a *transversal* of (V, E) if for all $H \in E, T \cap H \neq \emptyset$ holds. A transversal T of (V, E) is *minimal* if there is no transversal

T' of (V, E) such that $T' \subsetneq T$ [17]. Algorithm 2 computes the minimal transversals of the hypergraph that has the underlying attributes as vertex set and minimal disagree sets of tuples from world r_i as hyperedges. These form a cover of the set of keys that hold on r_i. The corresponding PKs thus hold with c-degree at least β_{k+1-i}. Using Theorem 3 we select PKs not implied by the other PKs as output.

Algorithm 2. Discover

Input: $(r, Poss_r)$ over $(R, \{\beta_1, \ldots, \beta_{k+1}\})$
Output: Cover Σ of PKs that are satisfied by $(r, Poss_r)$

for $i = 1, \ldots, k$ **do**
  $\textit{dis-ag}(r_i) \leftarrow \min\{X \subseteq R \mid \exists t, t' \in r_i \forall a \in R(t(a) \neq t'(a) \leftrightarrow a \in X)\}$;
  $\mathcal{H}_i \leftarrow (R, \textit{dis-ag}(r_i))$;
  $\Sigma_i \leftarrow \{(K, \beta_{k+1-i}) \mid K \in \mathit{Tr}(\mathcal{H}_i)\}$;
end for
$\Sigma \leftarrow \bigcup_{i=1}^{k} \Sigma_i$;
$\Sigma \leftarrow \{(K, \beta) \in \Sigma \mid \neg\exists(K', \beta') \in \Sigma(K' \subseteq K \wedge \beta' > \beta)\}$;
return Σ;

Example 5. We apply Algorithm 2 to the p-relation from Table 1. Using the first letters of each attribute we obtain

- $\textit{dis-ag}(r_1) = \{zr, tr, zt\}$ and $\Sigma_1 = \{(zr, \beta_4), (tr, \beta_4), (zt, \beta_4)\}$
- $\textit{dis-ag}(r_2) = \{zt, r\}$ and $\Sigma_2 = \{(zr, \beta_3), (tr, \beta_3)\}$
- $\textit{dis-ag}(r_3) = \{t, r\}$ and $\Sigma_3 = \{(tr, \beta_2)\}$, and
- $\textit{dis-ag}(r_4) = \{z, t, r\}$ and $\Sigma_3 = \{(ztr, \beta_1)\}$.

A cover Σ for the PKs that hold on the p-relation consists of (ztr, β_1), (tr, β_2), (zr, β_3), and (zt, β_4).

Theorem 7. *Algorithm 2 computes a cover of the set of PKs that are satisfied by the given p-relation r in time $\mathcal{O}(m + n^2)$ where $m := |R|^2 \times |r_k|^2 \times |\textit{dis-ag}(r_k)|$ and $n := \prod_{X \in \textit{dis-ag}(r_k)} |X|$.*

Proof. The soundness follows from the result that the keys of a relation are the minimal transversals of the disagree sets in the relation [11,31], and Theorem 3. The collection $\textit{dis-ag}(r_i)$ is computed in time $\mathcal{O}(m)$. The set of all minimal transversals for the simple hypergraph $\mathcal{H}_i$ is computed in time $\mathcal{O}(n^2)$. Algorithm 2 can compute the minimal hypergraphs incrementally with additional disagree sets discovered from tuples with lower p-degrees. □

6 Data Cleaning

In this section we illustrate an application of possibilistic keys for data cleaning purposes. The classical data cleaning problem can be stated as follows: Given a relation r and a set Σ of keys, find a relation $r' \subseteq r$ of maximum cardinality such that r' satisfies Σ. For example, the relation r

r					
zone	*time*	*rfid*	*object*	$Poss_r$	$Poss'_r$
Z3	1pm	H2	Grizzly	α_1	α_1
Z3	1pm	H3	Grizzly	α_1	α_2
Z3	3pm	H3	Grizzly	α_1	α_3
Z4	3pm	H3	Grizzly	α_1	α_4

violates the set $\Sigma = \{zt, zr, tr\}$ of keys. Solutions to the classical data cleaning problem would be the relations r_1 consisting of the first and third tuple, r_2 consisting of the first and last tuple, and r_3 consisting of the second and last tuple. Each solution requires us to remove at least two tuples from the relation. In this sense, classical data cleaning removes valuable information from the given relation.

We now introduce possibilistic data cleaning as a means to minimize the removal of tuples from a p-relation. For this purpose, we exploit the c-degrees of PKs to "reduce" the given p-degrees of tuples such that all PKs will be satisfied.

Given two p-relations $r_1 = (r', Poss_{r'})$ and $r_2 = (r, Poss_r)$ we say that r_1 is a *p-subrelation* of r_2, denoted by $r_1 \subseteq_p r_2$, if and only if $r'_i \subseteq r_i$ for $i = 1, \ldots, k$. The p-subset relationship is simply the partial order of functions induced by the ordering on p-degrees, that is, $r_1 \subseteq_p r_2$ if and only if $Poss_{r'}(t) \leq Poss_r(t)$ holds for all tuples t. The *p-cardinality* of the p-relation $(r, Poss_r)$ is the mapping $\mathcal{C} : \alpha_i \mapsto |r_i|$ for $i = 1, \ldots, k$. We compare p-cardinalities with respect to the lexicographical order, that is,

$$\mathcal{C}_1 <_L \mathcal{C}_2 :\Leftrightarrow \exists \alpha_i . \begin{array}{l} \mathcal{C}_1(\alpha_i) < \mathcal{C}_2(\alpha_i) \wedge \\ \mathcal{C}_1(\alpha_j) = \mathcal{C}_2(\alpha_j) \ \forall \alpha_j < \alpha_i \end{array}$$

The *possibilistic data cleaning problem* is: Given a p-relation r and set Σ of PKs, find a p-subrelation $r' \subseteq_p r$ of maximal p-cardinality so that Σ holds on r'.

A point that is perhaps controversial in our problem definition is the use of the lexicographic order $<_L$ in defining our target function to optimize. We chose this linearization of the natural partial order between p-cardinalities over other candidates for two reasons. Firstly, by maximizing $|r'_k| = |r'|$, the number of tuples completely "lost" during data cleaning is minimized. Secondly, it will allow us to develop more efficient algorithms for computing it.

For example, the p-relation $(r, Poss_r)$ violates the PK set

$$\Sigma = \{(zt, \beta_4), (zr, \beta_3), (tr, \beta_1)\}.$$

However, if we change the p-degree of the second tuple to α_2, the p-degree of the third tuple to α_3, and the p-degree of the last tuple to α_4, then the resulting p-relation $(r, Poss'_r)$ satisfies Σ. Note that none of the p-degrees had to be set to the bottom degree α_5. That is, every tuple in the cleaned p-relation $(r, Poss'_r)$ is at least somewhat possible to occur.

7 Query Processing

We demonstrate the benefit of PKs on query processing. Therefore, we add the attribute *p-degree* to the relation schema TRACKING with attributes *zone*, *time*, *rfid*, *object*.

Suppose we are interested in finding out which grizzly bears have been tracked in which zone, but we are only interested in answers that come from ‘certain’ or ‘quite possible’ tuples in the database. A user might enter the following SQL query:

```
SELECT DISTINCT zone, rfid, p-degree
FROM TRACKING
WHERE p-degree = α1 OR p-degree = α2
ORDER BY p-degree ASC
```

zone	*rfid*	*p-degree*
Z0	H0	α_1
Z1	H1	α_1
Z1	H2	α_1
Z3	H2	α_1
Z3	H3	α_2

which removes duplicate answers, and orders them with decreasing p-degree. When applied to the p-relation from Table 1, the query returns the answers on the right.

Firstly, our framework allows users to ask such queries - having available the p-degrees of tuples. Secondly, answers can be ordered according to the p-degree a huge benefit for users. Thirdly, the example shows how our framework can be embedded with standard technology, here SQL. Finally, recall our PK $(\{zone, rfid\}, \beta_3)$ which holds on the set of tuples that have p-degree α_1 or α_2. Consequently, the query answers satisfy the key $\{zone, rfid\}$ and the `DISTINCT` clause becomes superfluous. A query optimizer, capable of reasoning about PKs, can remove the `DISTINCT` clause from the input query without affecting its output. This optimization saves response time when answering queries, as duplicate elimination is an expensive operation and therefore not executed by default in SQL databases. PKs, and the ability to reason about them, have therefore direct applications to query processing.

8 Conclusion and Future Work

Possibilistic keys have been introduced to efficiently identify tuples of uncertain data. Uncertainty is modeled qualitatively by applying the AI framework of possibilistic logic to the fundamental database concept of keys. Tools were established to efficiently reason about possibilistic keys, to visualize and discover them effectively. Together, these tools can be used by data engineers to acquire the possibilistic keys that are semantically meaningful for a given application domain. It was further illustrated how possibilistic keys can be used to clean dirty data and enhance query processing. The results show that possibilistic keys can benefit applications with uncertain data, very much in the same way that keys benefit applications with certain data. It is future work to implement our algorithms in the form of a design tool, to apply possibility theory to other classes of popular database concepts, and to find efficient solutions to the possibilistic data cleaning problem we introduced.

Acknowledgement. This research is supported by the Marsden fund council from Government funding, administered by the Royal Society of New Zealand.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress, pp. 580–583 (1974)
2. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. J. ACM 31(1), 30–46 (1984)

3. Benferhat, S., Dubois, D., Prade, H.: Towards a possibilistic logic handling of preferences. Appl. Intell. 14(3), 303–317 (2001)
4. Beskales, G., Ilyas, I.F., Golab, L.: Sampling the repairs of functional dependency violations under hard constraints. PVLDB 3(1) (2010)
5. Bistarelli, S., Codognet, P., Rossi, F.: Abstracting soft constraints: Framework, properties, examples. Artif. Intell. 139(2), 175–211 (2002)
6. Bosc, P., Dubois, D., Prade, H.: Fuzzy functional dependencies – an overview and a critical discussion. In: Proc. of the 3rd IEEE Inter. Conf. on Fuzzy Systems (FUZZ-IEEE 1994), Orlando, FL, June 26-29, pp. 325–330 (1994)
7. Bosc, P., Pivert, O.: On the impact of regular functional dependencies when moving to a possibilistic database framework. Fuzzy Sets and Systems 140(1), 207–227 (2003)
8. Calvanese, D., De Giacomo, G., Lenzerini, M.: Keys for free in description logics. In: Baader, F., Sattler, U. (eds.) Proceedings of the 2000 International Workshop on Description Logics (DL 2000), Aachen, Germany, August 17-19. CEUR Workshop Proceedings, vol. 33, pp. 79–88. CEUR-WS.org (2000)
9. Chu, X., Ilyas, I.F., Papotti, P.: Holistic data cleaning: Putting violations into context. In: Jensen, C.S., Jermaine, C.M., Zhou, X. (eds.) 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, pp. 458–469. IEEE Computer Society (2013)
10. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM 13(6), 377–387 (1970)
11. Demetrovics, J., Katona, G.O.H.: Extremal combinatorial problems of database models. In: Biskup, J., Demetrovics, J., Paredaens, J., Thalheim, B. (eds.) MFDBS 1987. LNCS, vol. 305, pp. 99–127. Springer, Heidelberg (1988)
12. Diederich, J., Milton, J.: New methods and fast algorithms for database normalization. ACM Trans. Database Syst. 13(3), 339–365 (1988)
13. Dubois, D., Lang, J., Prade, H.: Automated reasoning using possibilistic logic: Semantics, belief revision, and variable certainty weights. IEEE Trans. Knowl. Data Eng. 6(1), 64–71 (1994)
14. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. Artif. Intell. 50(2), 223–239 (1991)
15. Dubois, D., Prade, H.: Fuzzy set and possibility theory-based methods in artificial intelligence. Artif. Intell. 148(1-2), 1–9 (2003)
16. Dubois, D., Prade, H., Schockaert, S.: Stable models in generalized possibilistic logic. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14. AAAI Press (2012)
17. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. SIAM J. Comput. 24(6), 1278–1304 (1995)
18. Fagin, R.: Horn clauses and database dependencies. J. ACM 29(4), 952–985 (1982)
19. Gärdenfors, P., Makinson, D.: Nonmonotonic inference based on expectations. Artif. Intell. 65(2), 197–245 (1994)
20. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inf. Comput. 91(1), 128–141 (1991)
21. Grabisch, M., Prade, H.: The correlation problem in sensor fusion in a possibilistic framework. Int. J. Intell. Syst. 16(11), 1273–1283 (2001)
22. Grove, A.: Two modellings for theory change. Journal of Philosophical Logic 17(2), 157–170 (1988)
23. Hartmann, S., Kirchberg, M., Link, S.: Design by example for SQL table definitions with functional dependencies. VLDB J 21(1), 121–144 (2012)

24. Hartmann, S., Leck, U., Link, S.: On Codd families of keys over incomplete relations. Comput. J. 54(7), 1166–1180 (2011)
25. Hartmann, S., Link, S.: Efficient reasoning about a robust XML key fragment. ACM Trans. Database Syst. 34(2) (2009)
26. Hartmann, S., Link, S.: Numerical constraints on XML data. Inf. Comput. 208(5), 521–544 (2010)
27. Heise, A., Quiane-Ruiz, J.-A., Abedjan, Z., Jentzsch, A., Naumann, F.: Scalable discovery of unique column combinations. PVLDB 7(4), 301–312 (2013)
28. Jha, A.K., Rastogi, V., Suciu, D.: Query evaluation with soft-key constraints. In: Lenzerini, M., Lembo, D. (eds.) Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, Vancouver, BC, Canada, June 9-11, pp. 119–128. ACM (2008)
29. Lutz, C., Areces, C., Horrocks, I., Sattler, U.: Keys, nominals, and concrete domains. J. Artif. Intell. Res. (JAIR) 23, 667–726 (2005)
30. Mannila, H., Räihä, K.J.: Design by example: An application of Armstrong relations. J. Comput. Syst. Sci. 33(2), 126–141 (1986)
31. Mannila, H., Räihä, K.J.: Algorithms for inferring functional dependencies from relations. Data Knowl. Eng. 12(1), 83–99 (1994)
32. Niepert, M., Gyssens, M., Sayrafi, B., Gucht, D.V.: On the conditional independence implication problem: A lattice-theoretic approach. Artif. Intell. 202, 29–51 (2013)
33. Qi, G., Wang, K.: Conflict-based belief revision operators in possibilistic logic. In: Hoffmann, J., Selman, B. (eds.) Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada, July 22-26. AAAI Press (2012)
34. Sabbadin, R., Fargier, H., Lang, J.: Towards qualitative approaches to multi-stage decision making. Int. J. Approx. Reasoning 19(3-4), 441–471 (1998)
35. Sarma, A.D., Ullman, J.D., Widom, J.: Schema design for uncertain databases. In: Arenas, M., Bertossi, L.E. (eds.) Proceedings of the 3rd Alberto Mendelzon International Workshop on Foundations of Data Management, Arequipa, Peru, May 12-15. CEUR Workshop Proceedings, vol. 450. CEUR-WS.org (2009)
36. Shen, Q., Leitch, R.: Fuzzy qualitative simulation. IEEE Transactions on Systems, Man, and Cybernetics 23(4), 1038–1061 (1993)
37. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2011)
38. Thalheim, B.: On semantic issues connected with keys in relational databases permitting null values. Elektronische Informationsverarbeitung und Kybernetik 25(1/2), 11–20 (1989)
39. Toman, D., Weddell, G.E.: On keys and functional dependencies as first-class citizens in description logics. J. Autom. Reasoning 40(2-3), 117–132 (2008)
40. Zadeh, L.A.: Approximate reasoning based on fuzzy logic. In: Buchanan, B.G. (ed.) Proceedings of the Sixth International Joint Conference on Artificial Intelligence, IJCAI 1979, Tokyo, Japan, August 20-23, vol. 2, pp. 1004–1010. William Kaufmann (1979)