Possibilistic Cardinality Constraints and Functional Dependencies

Tania K. Roblot and Sebastian Link(✉)

Department of Computer Science, University of Auckland, Auckland, New Zealand
{t.roblot,s.link}@auckland.ac.nz

Abstract. Cardinality constraints and functional dependencies together can express many semantic properties for applications in which data is certain. However, modern applications need to process large volumes of uncertain data. So far, cardinality constraints and functional dependencies have only been studied in isolation over uncertain data. We investigate the more challenging real-world case in which both types of constraints co-occur. While more expressive constraints could easily be defined, they would not enjoy the computational properties we show to hold for our combined class. Indeed, we characterize the associated implication problem axiomatically and algorithmically in linear input time. We also show how to summarize any given set of our constraints as an Armstrong instance. These instances help data analysts consolidate meaningful degrees of certainty by which our constraints hold in the underlying application domain.

Keywords: Data semantics · Integrity constraints · Possibility theory · Requirements engineering · Uncertain data

1 Introduction

Background. Cardinality constraints (CCs) and functional dependencies (FDs) are fundamental for understanding the structure and semantics of data, and have a long and fruitful history in conceptual modeling, database theory and practice. CCs were introduced in the seminal paper by Chen [5], while FDs were introduced in the seminal paper by Codd [6]. We focus on cardinality constraints that define an upper bound on the number of objects that have matching values on a given set of attributes. For example, any project manager should not be looking after more than three projects at any period of time. An FD expresses that the values on some attributes uniquely determine the values on some other attributes. For example, every project has at most one manager. Due to their ability to express desirable properties of many application domains, CCs and FDs have been used successfully for core data management tasks, including database cleaning, design, integration, modeling, querying, and updating.

Motivation. Relational databases were developed for applications with certain data, including accounting, inventory and payroll. Modern applications, such

I. Comyn-Wattiau et al. (Eds.): ER 2016, LNCS 9974, pp. 133–148, 2016.
DOI: 10.1007/978-3-319-46397-1_11

Emp	*Dep*	*Mgr*	*Degree of possibility*
Nishikori	Tennis	Federer	α_1 *(fully)*
Date	Tennis	Federer	α_1 *(fully)*
Sakita	Physics	Gauss	α_1 *(fully)*
Sato	Maths	Gauss	α_1 *(fully)*
Nara	Tennis	Federer	α_2 *(somewhat)*
Musashimaru	Sumo	Hakuho	α_2 *(somewhat)*
Musashimaru	Sumo	Taiho	α_2 *(somewhat)*

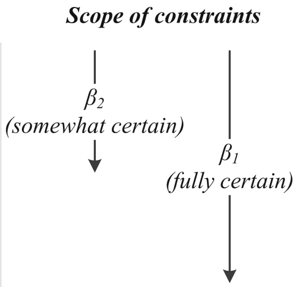

Fig. 1. A possibilistic instance and the scope by which constraints apply to its objects

as information extraction, sensors, and data integration produce large volumes of uncertain data. While different approaches to uncertainty in data exist, our running example considers a simple scenario in which a qualitative approach is applied to the integration of two data sources. The scenario maintains the levels of confidence associated with objects. Indeed, objects that occur in both sources are labeled 'fully possible', while objects that occur in only one source are labeled 'somewhat possible'. The information about the confidence of objects is clearly useful, but probability distributions are unavailable. Instead, a qualitative approach as founded in possibility theory is appropriate [9,10,35]. Figure 1 shows a possibilistic instance (p-instance) where each object is associated with a possibility degree (p-degree) from a finite scale: $\alpha_1 > \ldots > \alpha_{k+1}$. The top degree α_1 is reserved for objects that are 'fully possible', the bottom degree α_{k+1} for objects that are 'impossible' to occur. Intermediate degrees and their linguistic interpretations are used as preferred, such as 'somewhat possible' (α_2).

Interestingly, p-degrees enable us to express CCs and FDs with different degrees of certainty. For example, to express that it is 'impossible' that the same department and manager are associated with more than three employees we declare the CC $card(Dep, Mgr) \leq 3$ to be 'fully certain' by using the label β_1, stipulating that no combination of department and manager can feature in more than three objects that are at least 'somewhat possible'. Similarly, to say it is only 'somewhat possible' that departments with different managers exist we declare the FD $Dep \rightarrow Mgr$ as 'somewhat certain' by using the label β_2, stipulating that no department has more than one manager in 'fully possible' objects. We will investigate the combined class of CCs and FDs in this possibilistic data model.

Contributions and impact. Our contributions are as follows. (1) We show that the combination of CCs and FDs in a possibilistic data model constitutes a 'sweet spot' in terms of expressivity and computational behavior. In particular, we unify previous work under a more expressive framework that retains efficient computational properties. Slightly more expressive approaches result in non-axiomatizability, intractability, or even undecidability. (2) We establish a finite axiomatization and a linear-time decision algorithm for the associated implication problem. We illustrate applications from constraint maintenance, query optimization, and pivoting to eliminate data redundancy. (3) We establish an effective construction of Armstrong representations for any given set of our constraints. Here, we overcome the practical challenge that finite Armstrong instances do not frequently exist. We thus provide automated support for the acquisition of the constraints that are meaningful in a given application domain.

Organization. Section 2 discusses related work. Our data model is defined in Sect. 3. In Sect. 4 we characterize the implication problem axiomatically and algorithmically. Applications are highlighted in Sect. 5. Section 6 describes how to compute Armstrong representations. In Sect. 7 we conclude and discuss future work. Proofs are available in [39].

2 Related Work

FDs are probably the most studied class of constraints, due to their expressivity, computational behavior, and impact on practice. FDs were introduced in Codd's seminal paper [6], and are intrinsically linked to conceptual, logical, and physical database design [27,44]. Applications on the conceptual level include graphical reasoning [8] and pivoting [3,17]. CCs are an influential contribution of conceptual modeling to database constraints. They featured in Chen's seminal paper [5]. Cardinality constraints subsume the class of keys as a special case where the upper bound on the cardinality is fixed to 1. Keys are fundamental to most data models [4,11,16,21,22,26,29,30,42,46]. Most languages for conceptual design (description logics, ER, UML, ORM) come with means for specifying CCs. CCs have been studied extensively in database design [7,18,25,31,32,38,43].

Probability theory offers a popular quantitative approach to uncertain data [41]. Research about constraints on probabilistic data is in its infancy [4,40]. Probabilistic FDs, which specify a lower bound on the marginal probability that FDs exhibit on probabilistic databases, are not finitely axiomatizable.

The results of our article unify various previous works under one, more expressive, framework. In fact, our framework subsumes (1) the sole class of possibilistic CCs [15,28], (2) the sole class of possibilistic FDs [35], and (3) the combined class of CCs and FDs over relational data (the special case of possibilistic data with only one degree of confidence, i.e. where $k = 1$) [18]. While our framework is strictly more expressive, it retains the good computational properties of previous work, making it special. Indeed, making our framework more expressive is likely to result in the loss of good computational behavior. For example, using numerical dependencies instead of FDs leaves the implication problem not finitely axiomatizable [14], using multivalued dependencies requires more elaborate possibilistic data models and the interaction with CCs is not well-understood [19,20,23,24,33,34,45], using conditional FDs leaves the implication problem coNP-complete [13], adding inclusion dependencies makes the implication problem undecidable [37], and adding lower bounds to the upper bounds of our CCs' results requires us to solve unsolved problems from combinatorial design theory, even in the special case where $k = 1$ [18]. Further restrictions on what we additionally include are always possible, but our focus here is the natural class of cardinality constraints with upper bounds and functional dependencies.

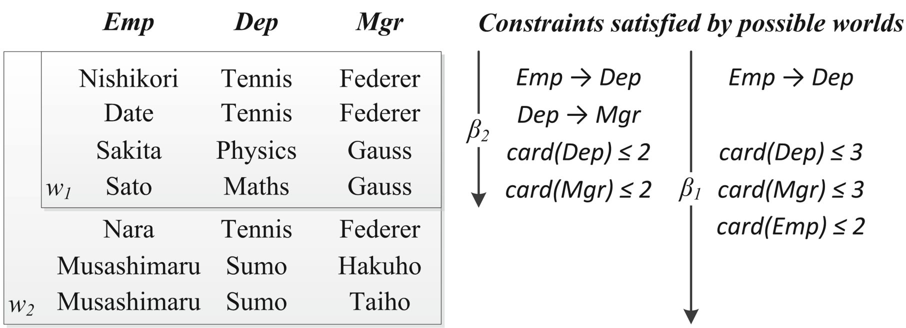

Fig. 2. Nested worlds of the p-instance from Fig. 1 and possibilistic constraints

3 Cardinality Constraints and Functional Dependencies

We extend object types that model certain objects in traditional conceptual modeling to model uncertain objects qualitatively. This allows us to extend CCs and FDs from their use on certain object types to uncertain object types.

An object type, denoted by O, is a finite non-empty set of *attributes*. Each attribute $A \in O$ has a *domain* $dom(A)$ of values. An *object* o over O is an element of the Cartesian product $\prod_{A \in O} dom(A)$. For $X \subseteq O$ we denote by $o(X)$ the *projection* of o on X. An *instance* over O is a set ι of objects over O. For example we use the object type Work with attributes *Emp*, *Dep*, and *Mgr*. Objects either belong or do not belong to an instance. For example, we cannot express that we have less confidence for Employee `Nara` to work in the department `Tennis` under Manager `Federer` than for the Employee `Nishikori`.

We model uncertain instances by assigning to each object some degree of possibility with which the object occurs in an instance. Formally, we have a *possibility scale*, or p-scale, that is, a strict linear order $\mathcal{S} = (S, <)$ with $k+1$ elements. We write $\mathcal{S} = \{\alpha_1, \dots, \alpha_{k+1}\}$ to declare that $\alpha_1 > \cdots > \alpha_k > \alpha_{k+1}$. The elements $\alpha_i \in S$ are called *possibility degrees*, or p-degrees. Here, α_1 is reserved for objects that are 'fully possible' while α_{k+1} is reserved for objects that are 'impossible' to occur in an instance. Humans like to use simple scales in everyday life to communicate, compare, or rank. Here, the word "simple" means that items are classified qualitatively rather than quantitatively by putting precise values on them. Classical instances use two p-degrees, i.e. $k = 1$.

A *possibilistic object type* $(O, \mathcal{S})$, or p-object type, consists of an object type O and a p-scale $\mathcal{S}$. A *possibilistic instance*, or p-instance, over $(O, \mathcal{S})$ consists of an instance ι over O, and a function *Poss* that assigns to each object $o \in \iota$ a p-degree $Poss(o) \in \mathcal{S} - \{\alpha_{k+1}\}$. We sometimes omit *Poss* when denoting a p-instance. Figure 1 shows a p-instance over $(\text{Work}, \mathcal{S} = \{\alpha_1, \alpha_2, \alpha_3\})$.

P-instances enjoy a possible world semantics. For $i = 1, \dots, k$ let w_i consist of all objects in ι that have p-degree at least α_i, that is, $w_i = \{o \in \iota \mid Poss(o) \geq \alpha_i\}$. Indeed, we have $w_1 \subseteq w_2 \subseteq \cdots \subseteq w_k$. If $o \notin w_k$, then $Poss(o) = \alpha_{k+1}$. Every object that is 'fully possible' occurs in every possible world, and is therefore also 'fully certain'. Hence, instances are a special case of uncertain instances. Figure 2 shows the possible worlds $w_1 \subsetneq w_2$ of the p-instance of Fig. 1.

As CCs and FDs are fundamental to applications with certain data, their possibilistic variants serve similar roles for applications with uncertain data. A *cardinality constraint* over object type O is an expression $card(X) \leq b$ where $X \subseteq O$, and b is a positive integer. The CC $card(X) \leq b$ over O is satisfied by an instance w over O, denoted by $\models_w card(X) \leq b$, if there are no $b+1$ distinct objects $o_1, \ldots, o_{b+1} \in w$ with matching values on all the attributes in X. For example, Fig. 2 shows that $card(Dep,Mgr) \leq 1$ is not satisfied by any instance w_1 or w_2, and $card(Dep,Mgr) \leq 2$ is satisfied by w_1, but not by w_2. A *functional dependency* over object type O is an expression $X \rightarrow Y$ where $X, Y \subseteq O$. The FD $X \rightarrow Y$ over O is satisfied by an instance w over O, denoted by $\models_w X \rightarrow Y$, if for any two objects $o_1, o_2 \in w$ the following holds: if $o_1(X) = o_2(X)$, then $o_1(Y) = o_2(Y)$. For example, Fig. 2 shows that $Dep \rightarrow Mgr$ is satisfied by w_1, but not by w_2, and $Emp \rightarrow Dep$ is satisfied by w_1 and w_2.

The p-degrees of objects result in degrees of certainty by which constraints hold. Since $Emp \rightarrow Dep$ holds in every possible world, it is fully certain to hold on ι. As $Dep \rightarrow Mgr$ and $card(Dep,Mgr) \leq 2$ are only violated in a somewhat possible world w_2, they are somewhat certain to hold on ι. Since $card(Dep,Mgr) \leq 1$ is violated in the fully possible world w_1, it is not certain to hold on ι.

Similar to the scale $\mathcal{S}$ of p-degrees α_i for objects, we use a scale $\mathcal{S}^T$ of certainty degrees, or c-degrees, β_j for CCs and FDs. Formally, the correspondence between p-degrees in $\mathcal{S}$ and the c-degrees in $\mathcal{S}^T$ is defined by the mapping $\alpha_i \mapsto \beta_{k+2-i}$ for $i = 1, \ldots, k+1$. Hence, the certainty $C_\iota(\sigma)$ by which the CC $\sigma = card(X) \leq b$ or FD $\sigma = X \rightarrow Y$ holds on the uncertain instance ι is either the top degree β_1 if σ is satisfied by w_k, or the minimum amongst the c-degrees β_{k+2-i} that correspond to possible worlds w_i in which σ is violated, that is,

$$C_\iota(\sigma) = \begin{cases} \beta_1 & , \textit{if } \models_{w_k} \sigma \\ \min\{\beta_{k+2-i} \mid \not\models_{w_i} \sigma\} & , \textit{otherwise} \end{cases}.$$

We can now define the semantics of possibilistic CCs and FDs. Let $(O, \mathcal{S})$ denote a p-object type. A possibilistic CC (p-CC) over $(O, \mathcal{S})$ is an expression $(card(X) \leq b, \beta)$ where $card(X) \leq b$ denotes a CC over O and $\beta \in \mathcal{S}^T$. A p-instance $(\iota, Poss)$ over $(O, \mathcal{S})$ satisfies the p-CC $(card(X) \leq b, \beta)$ if and only if $C_\iota(card(X) \leq b) \geq \beta$. A possibilistic FD (p-FD) over $(O, \mathcal{S})$ is an expression $(X \rightarrow Y, \beta)$ where $X \rightarrow Y$ denotes an FD over O and $\beta \in \mathcal{S}^T$. A p-instance $(\iota, Poss)$ over $(O, \mathcal{S})$ satisfies the p-FD $(X \rightarrow Y, \beta)$ if and only if $C_\iota(X \rightarrow Y) \geq \beta$.

For example, Fig. 2 shows some of the p-CCs and p-FDs that the p-instance ι from Fig. 1 satisfies. The next example introduces the set Σ of p-CCs and p-FDs we will use as an example constraint set in the remainder of the article.

Example 1. Let Σ denote the set with the following p-CCs and p-FDs over p-object type (Work, $\mathcal{S} = \{\alpha_1, \alpha_2, \alpha_3\}$): $(Emp \rightarrow Dep, \beta_1)$, $(card(Dep,Mgr) \leq 3, \beta_1)$, $(Dep \rightarrow Mgr, \beta_2)$, and $(card(Mgr) \leq 2, \beta_2)$. □

4 Computational Problems and Their Solutions

We establish fundamental tools to reason about p-CCs and p-FDs. Their applicability will be illustrated in Sect. 5. First, we define the implication problem and then address its solution in terms of inference rules and algorithms.

Let $\Sigma \cup \{\varphi\}$ denote a set of p-CCs and p-FDs over $(O, \mathcal{S})$. We say Σ *implies* φ, denoted by $\Sigma \models \varphi$, if every p-instance $(\iota, Poss)$ over $(O, \mathcal{S})$ that satisfies every element of Σ also satisfies φ. We use $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ to denote the *semantic closure* of Σ. The *implication problem for p-CCs and p-FDs* is to decide, given any p-object type, and any set $\Sigma \cup \{\varphi\}$ of p-CCs and p-FDs over the p-object type, whether $\Sigma \models \varphi$ holds.

Example 2. Let Σ be as in Example 1. Further, let σ denote the CC $card(Dep) \leq 2$. Then the highest c-degree β such that (σ, β) is implied by Σ is β_2. Indeed, Σ does not imply $\varphi = (\sigma, \beta_1)$. We can create a p-instance that has 3 different objects, all of which have matching values for department and manager, but pairwise different employees, and 2 of those objects have p-degree α_1 while the remaining object has p-degree α_2. Then the c-degree of $card(Dep) \leq 2$ in ι is β_2, which means that $(card(Dep) \leq 2, \beta_1)$ is violated. Since the c-degrees of $Emp \rightarrow Dep$, $Dep \rightarrow Mgr$, and $card(Dep,Mgr) \leq 3$ in ι are β_1, and the c-degree of $card(Mgr) \leq 2$ in ι is β_2, ι satisfies Σ, but violates φ. □

4.1 Using β-Cuts

Our overarching goal is to extend the combined use of CCs and FDs from certain to uncertain data, while maintaining their good computational properties. The core notion for achieving this goal is that of a β-cut for a given set Σ of p-CCs and p-FDs and c-degree $\beta > \beta_{k+1}$. Informally, the β-cut Σ_β of Σ contains all CCs and FDs σ such that there is some p-CCs or p-FD (σ, β') in Σ where β' is at least β. That is, $\Sigma_\beta = \{\sigma \mid (\sigma, \beta') \in \Sigma \text{ and } \beta' \geq \beta\}$ is the *β-cut* of Σ. The following theorem shows how the β-cut can be used to reduce the implication problem for p-CCs and p-FDs to the implication problem of traditional CCs and FDs. The theorem does not hold for CCs with lower bounds or multivalued dependencies.

Theorem 1. *Let $\Sigma \cup \{(\sigma, \beta)\}$ be a set of p-CCs and p-FDs over $(O, \mathcal{S})$ where $\beta > \beta_{k+1}$. Then $\Sigma \models (\sigma, \beta)$ if and only if $\Sigma_\beta \models \sigma$.*

Theorem 1 allows us to apply achievements from CCs and FDs for certain data to p-CCs and p-FDs. It is a major tool to establish our results.

Example 3. Let Σ be as in Example 1. Then Σ_{β_1} consists of $card(Dep,Mgr) \leq 3$ and $Emp \rightarrow Dep$, while Σ_{β_2} contains Σ_{β_1} and includes $card(Mgr) \leq 2$ and $Dep \rightarrow Mgr$. Using knowledge about the interaction of CCs and FDs from relational data [18], we conclude that Σ_{β_1} does not imply $card(Dep) \leq 2$, but Σ_{β_2} does imply $card(Dep) \leq 2$. Theorem 1 shows then that Σ does not imply $(card(Dep) \leq 2, \beta_1)$, but Σ does imply $(card(Dep) \leq 2, \beta_2)$. In fact, the possible world w_1 of the p-instance ι from Example 2 satisfies Σ_{β_1}, and violates $card(Dep) \leq 2$. □

Table 1. Finite axiomatization of p-CCs and p-FDs

$\dfrac{}{(XY \rightarrow X, \beta_1)}$ (reflexivity)	$\dfrac{(X \rightarrow Y, \beta)}{(X \rightarrow XY, \beta)}$ (extension)	$\dfrac{(X \rightarrow Y, \beta) \quad (Y \rightarrow Z, \beta)}{(X \rightarrow Z, \beta)}$ (transitivity)
$\dfrac{}{(card(O) \leq 1, \beta_1)}$ (top)	$\dfrac{(card(X) \leq b, \beta)}{(card(X) \leq b+1, \beta)}$ (relax)	$\dfrac{(X \rightarrow Y, \beta) \quad (card(Y) \leq b, \beta)}{(card(X) \leq b, \beta)}$ (pullback)
$\dfrac{(card(X) \leq 1, \beta)}{(X \rightarrow Y, \beta)}$ (key)	$\dfrac{}{(\sigma, \beta_{k+1})}$ (bottom)	$\dfrac{(\sigma, \beta)}{(\sigma, \beta')}\beta' \leq \beta$ (weakening)

4.2 Axiomatic Characterization

A finite axiomatization allows us to effectively enumerate all implied p-CCs and p-FDs, that is, to determine the semantic closure $\Sigma^* = \{\sigma \mid \Sigma \models \sigma\}$ of Σ. A finite axiomatization facilitates human understanding of the interaction of the given constraints, and ensures all opportunities for the use of these constraints in applications can be exploited (Sect. 5). We determine the semantic closure by applying *inference rules* of the form $\dfrac{\text{premise}}{\text{conclusion}}$. 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 $(O, \mathcal{S})$ 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. Table 1 shows an axiomatization $\mathfrak{C}$ for p-CCs and p-FDs. Here, $(O, \mathcal{S})$ is an arbitrarily given p-object type, $X, Y \subseteq O$, b is a positive integer, $\beta, \beta' \in \mathcal{S}^T$ are c-degrees, and σ uniformly denotes either some CC or FD. In particular, β_{k+1} denotes the bottom c-degree in $\mathcal{S}^T$.

Theorem 2. *The set $\mathfrak{C}$ forms a finite axiomatization for the implication of possibilistic cardinality constraints and functional dependencies.* $\square$

The application of inference rules in $\mathfrak{C}$ from Table 1 is illustrated next.

Example 4. Consider Σ from Example 1. Applying pullback to $(Dep \rightarrow Mgr, \beta_2)$ and $(card(Mgr) \leq 2, \beta_2)$ results in $(card(Dep) \leq 2, \beta_2) \in \Sigma^+_{\mathfrak{C}}$. For an inference of $(card(Emp,Mgr) \leq 1, \beta_1)$ consider the following steps. Applying reflexivity infers $(Emp,Mgr \rightarrow Emp, \beta_1)$. Then we apply transitivity to $(Emp,Mgr \rightarrow Emp, \beta_1)$ and $(Emp \rightarrow Dep, \beta_1)$ to infer $(Emp,Mgr \rightarrow Dep, \beta_1)$. Next we apply extension to $(Emp,Mgr \rightarrow Dep, \beta_1)$ to infer $(Emp,Mgr \rightarrow Emp,Dep,Mgr, \beta_1)$. The top rule infers ($card(Emp,Dep,Mgr) \leq 1, \beta_1$). Finally, we apply pullback to ($Emp,Mgr \rightarrow Emp,Dep,Mgr, \beta_1$) and ($card(Emp,Dep,Mgr) \leq 1, \beta_1$) to infer ($card(Emp,Mgr) \leq 1, \beta_1) \in \Sigma^+_{\mathfrak{C}}$. □

4.3 Algorithmic Characterization

While $\mathfrak{C}$ enables us to enumerate all p-CCs and p-FDs that are implied by a set Σ of p-CCs and p-FDs, in practice it often suffices to decide whether a given p-CC or p-FD φ is implied by Σ. Enumerating all implied constraints and checking whether φ is among them is neither efficient nor makes good use of φ. However, our axiomatization $\mathfrak{C}$ provides us with the insight to develop efficient algorithms for deciding the associated implication problem.

First, Theorem 1 tells us that the implication of some p-CC or p-FD (σ, β) by Σ can be decided by considering the β-cut Σ_β. If σ denotes an FD $X \rightarrow Y$, then our axiomatization $\mathfrak{C}$ tells us that the decision only depends on the FDs in Σ_β and the cardinality constraints $card(X) \leq 1 \in \Sigma_\beta$, as the latter implies the FD $X \rightarrow O \in \Sigma^*_\beta$. For a given set Σ of cardinality constraints and functional dependencies, let Σ[FD] denote the set of FDs in Σ together with the FDs $X \rightarrow O$ for every $card(X) \leq 1 \in \Sigma$. The p-FD $(X \rightarrow Y, \beta)$ is therefore implied by Σ if and only if the FD $X \rightarrow Y$ is implied by Σ_β[FD]. The latter condition is equivalent to Y being a subset of the attribute set closure $X^+_{\Sigma_\beta[\text{FD}]} = \{A \in X \mid \Sigma_\beta[\text{FD}] \models X \rightarrow A\}$, which can be computed in linear time in the input set Σ_β[FD] [1]. This shows condition (i) in Theorem 3 below. If σ denotes a cardinality constraint $card(X) \leq b$, then our axiomatization $\mathfrak{C}$ tells us that the decision only depends on the existence of some cardinality constraint $card(Y) \leq b' \in \Sigma_\beta$ such that $Y \subseteq X^+_{\Sigma_\beta[\text{FD}]}$ and $b' \leq b$. The clause that $b' \leq b$ follows from the relax rule, and the clause that $Y \subseteq X^+_{\Sigma_\beta[\text{FD}]}$ follows from the pullback rule and the fact that $X^+_{\Sigma_\beta[\text{FD}]}$ is the maximal subset of O that is functionally determined by X given Σ_β[FD]. This shows condition (ii) in Theorem 3 below.

Theorem 3. *Let Σ denote a set of p-CCs and p-FDs over $(O, \mathcal{S})$ with $|\mathcal{S}| = k+1$. Then (i) Σ implies $(X \rightarrow Y, \beta)$ if and only if $Y \subseteq X^+_{\Sigma_\beta[FD]}$, and (ii) Σ implies $(card(X) \leq b, \beta)$ if and only if $X^+_{\Sigma_\beta[FD]} = O$, or there is some $card(Y) \leq b' \in \Sigma_\beta$ such that $Y \subseteq X^+_{\Sigma_\beta[FD]}$ and $b' \leq b$.*

The worst-case complexity bound in the following result follows from the well-known fact that the computation of $X^+_{\Sigma[FD]}$ is linear in the total number of attribute occurrences in $\Sigma[FD]$ [1], and this size of $\Sigma[FD]$ is bounded by $|O| \times |\Sigma|$ where $|S|$ denotes the number of elements in S.

Corollary 1. *An instance $\Sigma \models \varphi$ of the implication problem for p-CCs and p-FDs can be decided in time $\mathcal{O}(|O| \times |\Sigma \cup \{\varphi\}|)$.* □

We illustrate the use of Theorem 3 on our running example.

Example 5. Let Σ be as in Example 1. Then we can use Theorem 3 to decide whether the p-CC $(card(Dep) \leq 2, \beta_2)$ is implied by Σ. Indeed, $Dep^{*}_{\Sigma_{\beta_2}[\text{FD}]} = \{Dep, Mgr\}$ and $card(Mgr) \leq 2 \in \Sigma_{\beta_2}$. Similarly, Σ implies $(card(Emp,Mgr) \leq 1, \beta_1)$ since $\{Emp,Mgr\}^{+}_{\Sigma_{\beta_1}[\text{FD}]} = O$. $\square$

5 Applications

We give a series of examples that illustrate core data processing areas on which our solutions have an impact. These include more efficient update and query operations, as well as schema decompositions to avoid data redundancy.

Non-redundant Constraint Maintenance. Constraints ensure data integrity. Whenever database instances are updated, it must be validated that the updated instance satisfies all the given constraints. Data integrity therefore comes at the cost of enforcing it. However, it is redundant to validate any implied constraints, because every instance that satisfies the remaining constraints already satisfies the implied constraints. Unnecessary costs for implied constraints are removed by computing a non-redundant cover of the given constraint set. This is done by successively removing any constraint $\sigma \in \Sigma$ from Σ whenever $\Sigma - \{\sigma\}$ implies σ. Having an efficient algorithm to decide implication means that we also have an efficient algorithm to compute a non-redundant set of constraints. Note that the time complexity refers to the schema size, which is negligible in comparison to the size of the instance. Furthermore, the larger database instances are the more time we save by validating non-redundant sets of constraints. We will now illustrate these ideas on our running example from the introduction. Some of the p-CCs and p-FDs satisfied by the p-instance in Fig. 1 include: $(Emp \rightarrow Dep, \beta_1)$, $(card(Dep) \leq 3, \beta_1)$, $(card(Mgr) \leq 3, \beta_1)$, $(card(Emp) \leq 2, \beta_1)$, $(card(Emp,Dep) \leq 2, \beta_1)$, $(Emp \rightarrow Dep, \beta_2)$, $(Dep \rightarrow Mgr, \beta_2)$, $(Emp \rightarrow Mgr, \beta_2)$, $(card(Dep) \leq 2, \beta_2)$, $(card(Mgr) \leq 2, \beta_2)$, $(card(Dep,Mgr) \leq 2, \beta_2)$, and $(card(Emp, Mgr) \leq 3, \beta_2)$. This set is redundant, and a non-redundant subset that implies all constraints of the given set is shown in Fig. 2.

Query Optimization. Knowing which constraints hold on a given instance also assists us with making the evaluation of queries more efficient. Take, for example, the query

`SELECT DISTINCT` *Emp* `FROM` WORK `WHERE` $p\text{-}degree{=}\alpha_1$;

and assume it is evaluated on the p-instance from Fig. 1. Since the p-instance satisfies the p-CCs and p-FDs in Fig. 2, and these constraints imply the p-CC $card(Emp \leq 1, \beta_2)$, a query optimizer that can reason about our constraints is able to conclude that the `DISTINCT` clause in the query above is superfluous. The elimination of this clause can save considerable evaluation time because the ordering of tuples and removing of duplicates is an expensive operation. For another query evaluated on the same p-instance consider

```
SELECT Dep, COUNT(Emp) FROM WORK WHERE p-degree=α1
GROUP BY Dep HAVING Count(Emp)≤ 3;
```

which lists the departments together with the number of their 'certain' employees, if that number does not exceed 3. A query optimizer able to determine that the p-CC $(card(Dep) \leq 3, \beta_2)$ is implied by the satisfied set of p-CCs and p-FDs, can remove the `HAVING` clause from the query without affecting the result.

Removing Data Redundancy by Pivoting. The goal of pivoting is to decompose object schemata at design time in an effort to reduce data redundancy and optimize constraint validation time during the lifetime of the target database. We briefly use our running example to illustrate the impact of possibilistic constraints on pivoting. For this purpose, consider again the (possible worlds of the) p-instance in Fig. 2.

Each occurrence of the *Mgr*-value `Federer` in world w_1 is redundant in the sense that any update of this occurrence to a different value would result in a violation of the p-FD $(Dep \to Mgr, \beta_2)$. In contrast, the occurrence of `Federer` in w_2 is not redundant, because the p-FD $(Dep \to Mgr, \beta_2)$ only applies to objects with p-degree α_1. In other words, we could decompose the schema WORK into the two schemata $\{Dep, Mgr, ID1\}$ and $\{Emp, ID1\}$ for objects with p-degree α_1. For objects with p-degree α_2 we could decompose WORK into the two schemata $\{Emp, Dep, ID2\}$ and $\{Mgr, ID2\}$, based on the p-FD $(Emp \to Dep, \beta_1)$. That is, our framework enables us to first apply a horizontal decomposition of the given database instance into w_1 and $w_2 - w_1$, and then apply traditional pivoting to decompose the schemata with respect to the β-cuts Σ_{β_2} and Σ_{β_1}, respectively. The resulting decomposition of the p-instance from Fig. 1 would look like:

Dep	*Mgr*	*ID*1
Tennis	Federer	1
Physics	Gauss	2
Maths	Gauss	3

Emp	*ID*1
Nishikori	1
Date	1
Sakita	2
Sato	3

Emp	*Dep*	*ID*2
Nara	Tennis	1
Musashimaru	Sumo	2

Mgr	*ID*2
Federer	1
Hakuho	2
Taiho	2

in which all redundant data value occurrences have been removed. In addition, the original cardinality constraint $(card(Dep, Mgr) \leq 2, \beta_2)$ now becomes a cardinality constraint stipulating that each $ID1$ value in the $\{Dep, Mgr, ID1\}$ instance should occur in at least 1 and at most 2 objects of the $\{Emp, ID1\}$ instance.

6 Armstrong Instances and Representations

We establish computational support for the acquisition of p-CCs and p-FDs that are meaningful in a given application domain. A major inhibitor to the acquisition is the mismatch in expertise between business analysts and domain experts.

The former know database concepts but not the domain, while the latter know the domain but not database concepts. To facilitate effective communication between them, Armstrong instances serve as data samples that perfectly represent the current set of constraint sets. We will sketch how to compute Armstrong instances for any given set of p-CCs and p-FDs, which analysts and experts can jointly inspect to consolidate the set of meaningful constraints.

We first restate the original definition of an Armstrong database [12] in our context. A p-instance ι is said to be *Armstrong* for a given set Σ of p-CCs and p-FDs on a given p-object type $(O, \mathcal{S})$ if and only if for all p-CCs and p-FDs φ over $(O, \mathcal{S})$ it is true that ι satisfies φ if and only if Σ implies φ. As such, Armstrong p-instances exhibit for each cardinality constraint and functional dependency the largest c-degree for which it is implied by the given set Σ.

Example 6. The p-instance from Fig. 1 is Armstrong for the set of p-CCs and p-FDs from Fig. 2. $\square$

We will now explain how to compute an Armstrong p-instance ι for an arbitrarily given set Σ of p-CCs and p-FDs.

For every attribute subset X and every c-degree β_i, we compute the smallest $b_{X,i}$ such that $(card(X) \leq b_{X,i}, \beta_i)$ is implied by Σ. We start with $b_{X,i} = \infty$, and set $b_{X,i} = 1$, if $X^+_{\Sigma_{\beta_i}[\text{FD}]} = O$ holds. Otherwise, we set $b_{X,i}$ to b whenever there is some $card(Y) \leq b \in \Sigma_{\beta_i}$ such that $Y \subseteq X^+_{\Sigma_{\beta_i}[\text{FD}]}$ and $b < b_{X,i}$, see Theorem 3 (ii). Now it suffices to introduce $b_{X,i}$ objects into ι with p-degree α_{k+1-i} and matching values $c_{A,i}$ on all $A \in X$ and unique values on all $A \notin X$. This ensures that all p-CCs implied by Σ are satisfied in ι and all p-CCs not implied by Σ are violated. Several optimizations reduce the number of objects in an Armstrong p-instance: If $b_{X,i} = 1$, no objects need to be introduced in ι. If $Y \subset X$ and $b_{Y,i} = b_{X,i}$, then it suffices to introduce $b_{X,i}$ objects, because they also violate $(card(Y) \leq b_{Y,i}, \beta_i)$. For $j > i$ and $b_{X,j} \leq b_{X,i}$ for which $b_{X,j}$ objects with (at most) p-degree α_{k+1-j} have already been introduced, it suffices to introduce further $b_{X,i} - b_{X,j}$ objects of p-degree α_{k+1-i}, again with matching values $c_{X,j}$ on all $A \in X$ and unique values on all $A \notin X$.

As an illustration, Fig. 3 shows for all attribute subsets X and c-degrees β_1 and β_2 the associated cardinalities $b_{X,i}$ for our running example from Example 1. The bold attribute sets are those that require the insertion of objects into an Armstrong p-instance for the given Σ.

In general, we still need to ensure that all p-FDs not implied by Σ are violated. For all $A \in X$ and every c-degree β_i, we compute all maximal attribute subsets X such that $A \notin X^+_{\Sigma_{\beta_i}[\text{FD}]}$, i.e., for all $B \notin (XA)$ we have $A \in (XB)^+_{\Sigma_{\beta_i}[\text{FD}]}$. These sets are known as the *maximal sets* for $\Sigma_{\beta_i}[\text{FD}]$ and can be computed by an algorithm given in [36]. For each set X that is maximal with respect to $\Sigma_{\beta_i}[\text{FD}]$, we introduce two objects with p-degree α_{k+1-i} and matching values $c_{A,i}$ on all $A \in X$ and unique values on all $A \notin X$. Again, some optimizations reduce the number of objects in the final Armstrong p-instance: If X is maximal with respect to $\Sigma_{\beta_i}[\text{FD}]$ and $\Sigma_{\beta_j}[\text{FD}]$ and $i < j$, then it suffices to introduce the two objects with p-degree α_{k+1-j}. Finally, we do not need

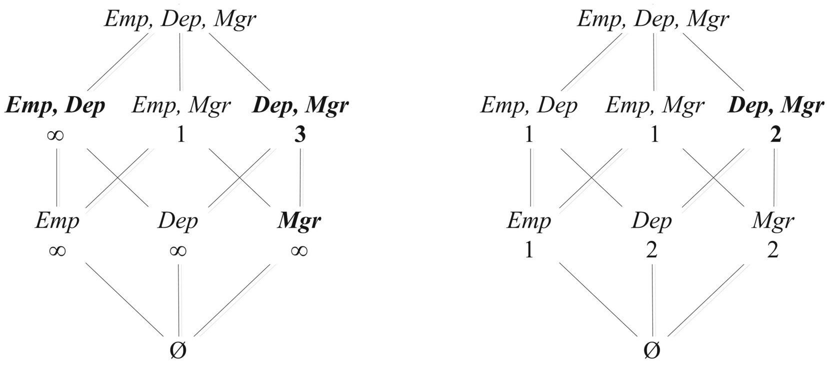

Fig. 3. Attribute sets X with cardinalities $b_{X,i}$ for $i = 1, 2$ from left to right

to introduce the two objects for the maximal set X, if $b_{Y,j}$ objects have previously been introduced for some $j \geq i$ where $X \subseteq Y$ and X is only maximal for attributes $A \notin Y - X$ with respect to $\Sigma_{\beta_i}[\text{FD}]$.

Continuing with the construction of an Armstrong p-instance for the given set Σ from Example 1, the following table lists the attribute subsets (only one maximal set in each case here) that are maximal for the given attributes and $\Sigma_{\beta_i}[\text{FD}]$.

	Emp	*Dep*	*Mgr*
Σ_{β_1}	$\{Dep, Mgr\}$	$\{Mgr\}$	$\{Emp, Dep\}$
Σ_{β_2}	$\{Dep, Mgr\}$	$\{Mgr\}$	$\emptyset$

Indeed, only the set $\{Mgr\}$ that is maximal for $\Sigma_{\beta_2}[\text{FD}]$ requires us to insert two objects. In particular, the maximal set $X = \{Dep, Mgr\}$ for $\Sigma_{\beta_2}[\text{FD}]$ is already covered by the $b_{X,2} = 2$ objects introduced previously, see Fig. 3, and the maximal set $\emptyset$ is covered because the p-FD $(\emptyset \rightarrow Mgr, \beta_2)$ is already violated after two objects with different *Mgr* values have been introduced. Similarly, all the maximal sets for $\Sigma_{\beta_1}[\text{FD}]$ have already been covered.

The outlined algorithm ensures that Armstrong p-instances exist for every given set Σ of p-CCs and p-FDs, and that they are computed in time exponential in input. Since there are cases where the minimum number of required objects is exponential in the given input, which is known for traditional FDs [2], no polynomial-time algorithm can exist. However, as our running example illustrates we still need to deal with the following occurring case, which occurs frequently in practice. There are attribute subsets X and c-degrees β_i such that $b_{X,i} = \infty$, that is, there is no finite upper bound b such that $(card(X) \leq b, \beta_i)$ is implied by the input Σ. It follows that every Armstrong p-instance is necessarily infinite, which seems to make our acquisition strategy unfit for its intended purpose. However, we apply the following representation trick that overcomes this challenge. Instead of introducing $b_{X,i}$ different objects with matching values $c_{A,i}$

on all $A \in X$ and unique values on all $A \notin X$, we introduce one single object with $c_{A,i}$ on all $A \in X$ and $*$ on all $A \notin X$, plus its cardinality $b_{X,i}$ in a new column **card**. This single object represents the $b_{X,i}$ different objects, in particular, $*$ for unique values in all columns outside of X. If the objects result from a maximal set, then the cardinality is simply 2. Representations resulting from this transformation of (finite or infinite) Armstrong p-instances are called *Armstrong p-representations* for Σ. We can show that the optimizations applied in our computation result in representations that are bounded by the size of minimum-sized Armstrong p-representations (those Armstrong p-representations with the least number of objects) and the number of given constraints.

Theorem 4. *Given an arbitrary set Σ of poss-CCs and poss-FDs over some given p-object type, the outlined algorithm computes an Armstrong p-representation ζ for Σ whose size is bounded by that of a minimum-sized Armstrong p-representation ζ^{min} for Σ and the number of elements in Σ as follows: $|\zeta| \leq |\zeta^{min}| \times (|\Sigma| + |\zeta^{min}|)$.* □

This construction yields the following Armstrong p-instance for the given set Σ of p-CCs and p-FDs from Example 1.

Emp	*Dep*	*Mgr*	*p-degree*	card
$*$	Tennis	Federer	α_1	2
$*$	$*$	Gauss	α_1	2
$*$	Tennis	Federer	α_2	1
Musashimaru	Sumo	$*$	α_2	∞
$*$	$*$	Taiho	α_2	∞

We list some of the observations we can make by inspecting this Armstrong p-instance. First of all, the given constraint set Σ has not captured any 'fully certain' finite bounds on the cardinalities by which (*Emp*,*Dep*)-objects or (*Mgr*)-objects occur. Indeed, the combination (Musashimaru, Sumo) can occur infinitely many times when 'somewhat possible' objects are involved, and the same applies to (Taiho). In contrast, Σ does guarantee the uniqueness of any (*Emp*)-objects that are 'fully possible', and a maximum cardinality of two on any (*Dep*,*Mgr*)-objects that are 'fully possible'. Similarly, any non-trivial FD $Mgr \rightarrow A$ is not even 'somewhat certain'. The FD $Dep \rightarrow Mgr$ is 'somewhat certain', because there are two 'somewhat possible' occurrences of the (Sumo) department, but in combination with different managers. While the FD $Emp \rightarrow Dep$ is 'fully certain', the FD $Emp \rightarrow Mgr$ is only 'somewhat certain', because there are two 'somewhat possible' occurrences of the employee (Musashimaru), but each occurrence is in combination with different managers.

7 Conclusion and Future Work

Cardinality constraints and functional dependencies naturally co-occur in most aspects of life. Consequently, they have received invested interest from the conceptual modeling community over the last three decades. In contrast to various previous works, we have studied cardinality constraints and functional dependencies over uncertain data. Uncertainty has been modeled qualitatively by applying the framework of possibility theory. Our results show that cardinality constraints and functional dependencies form a 'sweet spot' in terms of both expressivity and good computational behavior, as more expressive classes of constraints behave poorly. In particular, we have established a finite axiomatization and a linear time algorithm to decide the implication problem associated with our class, and illustrated their applicability to conceptual design, update and query efficiency. We have also established an algorithm that computes for every given set of our constraints an Armstrong representation. These representations embody the exact certainty with which any constraint in our class is currently perceived to hold by data analysts. The analysts can show our Armstrong representations to domain experts in order to jointly consolidate the actual certainty with which cardinality constraints and functional dependencies shall hold in a given application domain.

Our framework opens up several questions for future investigation, including a detailed study and performance tests for our applications, the interaction with yet other constraint classes despite the limits outlined, and empirical evaluations for the usefulness of Armstrong representations. It is further interesting to investigate possibilistic approaches to more expressive data models, such as SQL with partial and duplicate information, XML, RDF, or graph databases.

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

References

1. Beeri, C., Bernstein, P.: Computational problems related to the design of normal form relational schemas. ACM Trans. Database Syst. **4**(1), 30–59 (1979)
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. Biskup, J., Menzel, R., Polle, T., Sagiv, Y.: Decomposition of relationships through pivoting. In: Thalheim, B. (ed.) ER 1996. LNCS, vol. 1157, pp. 28–41. Springer, Heidelberg (1996). doi:10.1007/BFb0019913
4. Brown, P., Link, S.: Probabilistic keys for data quality management. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 118–132. Springer, Heidelberg (2015). doi:10.1007/978-3-319-19069-3_8
5. Chen, P.P.: The Entity-Relationship model - toward a unified view of data. ACM Trans. Database Syst. **1**(1), 9–36 (1976)
6. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)

7. Currim, F., Neidig, N., Kampoowale, A., Mhatre, G.: The CARD system. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 433–437. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16373-9_31
8. Demetrovics, J., Molnár, A., Thalheim, B.: Graphical reasoning for sets of functional dependencies. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 166–179. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30464-7_14
9. Dubois, D., Prade, H.: Possibility theory and its applications: Where do we stand? In: Kacprzyk, J., Pedrycz, W. (eds.) Springer Handbook of Computational Intelligence, pp. 31–60. Springer, Heidelberg (2015)
10. Dubois, D., Prade, H.: Practical methods for constructing possibility distributions. Int. J. Intell. Syst. **31**(3), 215–239 (2016)
11. Fagin, R.: A normal form for relational databases that is based on domains and keys. ACM Trans. Database Syst. **6**(3), 387–415 (1981)
12. Fagin, R.: Horn clauses and database dependencies. J. ACM **29**(4), 952–985 (1982)
13. Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for capturing data inconsistencies. ACM Trans. Database Syst. **33**(2), 94–115 (2008)
14. Grant, J., Minker, J.: Inferences for numerical dependencies. Theor. Comput. Sci. **41**, 271–287 (1985)
15. Hall, N., Köhler, H., Link, S., Prade, H., Zhou, X.: Cardinality constraints on qualitatively uncertain data. Data Knowl. Eng. **99**, 126–150 (2015)
16. Hannula, M., Kontinen, J., Link, S.: On the finite and general implication problems of independence atoms and keys. J. Comput. Syst. Sci. **82**(5), 856–877 (2016)
17. Hartmann, S.: Decomposing relationship types by pivoting and schema equivalence. Data Knowl. Eng. **39**(1), 75–99 (2001)
18. Hartmann, S.: On the implication problem for cardinality constraints and functional dependencies. Ann. Math. Artif. Intell. **33**(2–4), 253–307 (2001)
19. Hartmann, S., Link, S.: Multi-valued dependencies in the presence of lists. In: Beeri, C., Deutsch, A. (eds.) Proceedings of the Twenty-Third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 14–16 June 2004, Paris, France, pp. 330–341. ACM (2004)
20. Hartmann, S., Link, S.: On a problem of Fagin concerning multivalued dependencies in relational databases. Theor. Comput. Sci. **353**(1–3), 53–62 (2006)
21. Hartmann, S., Link, S.: Efficient reasoning about a robust XML key fragment. ACM Trans. Database Syst. **34**(2) (2009)
22. Hartmann, S., Link, S.: Expressive, yet tractable XML keys. In: Kersten, M.L., Novikov, B., Teubner, J., Polutin, V., Manegold, S. (eds.) EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, 24–26 March, 2009, Proceedings. ACM International Conference Proceeding Series, vol. 360, pp. 357–367. ACM (2009)
23. Hartmann, S., Link, S., Schewe, K.-D.: Reasoning about functional and multivalued dependencies in the presence of lists. In: Seipel, D., Turull-Torres, J.M. (eds.) FoIKS 2004. LNCS, vol. 2942, pp. 134–154. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24627-5_10
24. Hartmann, S., Link, S., Schewe, K.: Functional and multivalued dependencies in nested databases generated by record and list constructor. Ann. Math. Artif. Intell. **46**(1–2), 114–164 (2006)
25. Jones, T.H., Song, I.Y.: Analysis of binary/ternary cardinality combinations in entity-relationship modeling. Data Knowl. Eng. **19**(1), 39–64 (1996)

26. Köhler, H., Leck, U., Link, S., Zhou, X.: Possible and certain keys for SQL. VLDB J. **25**(4), 571–596 (2016)
27. Köhler, H., Link, S.: SQL schema design: Foundations, normal forms, and normalization. In: Özcan, F., Koutrika, G., Madden, S. (eds.) Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, 26 June–01 July 2016, pp. 267–279. ACM (2016)
28. Koehler, H., Link, S., Prade, H., Zhou, X.: Cardinality constraints for uncertain data. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) ER 2014. LNCS, vol. 8824, pp. 108–121. Springer, Heidelberg (2014). doi:10.1007/978-3-319-12206-9_9
29. Köhler, H., Link, S., Zhou, X.: Possible and certain SQL keys. PVLDB **8**(11), 1118–1129 (2015)
30. Köhler, H., Link, S., Zhou, X.: Discovering meaningful certain keys from incomplete and inconsistent relations. IEEE Data Eng. Bull. **39**(2), 21–37 (2016)
31. Lenzerini, M., Nobili, P.: On the satisfiability of dependency constraints in entity-relationship schemata. Inf. Syst. **15**(4), 453–461 (1990)
32. Liddle, S.W., Embley, D.W., Woodfield, S.N.: Cardinality constraints in semantic data models. Data Knowl. Eng. **11**(3), 235–270 (1993)
33. Link, S.: Charting the completeness frontier of inference systems for multivalued dependencies. Acta Inf. **45**(7–8), 565–591 (2008)
34. Link, S.: Characterisations of multivalued dependency implication over undetermined universes. J. Comput. Syst. Sci. **78**(4), 1026–1044 (2012)
35. Link, S., Prade, H.: Possibilistic functional dependencies and their relationship to possibility theory. IEEE Trans. Fuzzy Syst. **24**(3), 757–763 (2016)
36. Mannila, H., Räihä, K.J.: Design by example: an application of Armstrong relations. J. Comput. Syst. Sci. **33**(2), 126–141 (1986)
37. Mitchell, J.C.: The implication problem for functional and inclusion dependencies. Inf. Control **56**(3), 154–173 (1983)
38. Queralt, A., Artale, A., Calvanese, D., Teniente, E.: OCL-lite: finite reasoning on UML/OCL conceptual schemas. Data Knowl. Eng. **73**, 1–22 (2012)
39. Roblot, T.: Cardinality constraints for probabilistic and possibilistic databases. Ph.D. thesis, Department of Computer Science, The University of Auckland, New Zealand (2016)
40. Roblot, T., Link, S.: Probabilistic cardinality constraints. In: Johannesson, P., Lee, M.L., Liddle, S.W., Opdahl, A.L., López, Ó.P. (eds.) ER 2015. LNCS, vol. 9381, pp. 214–228. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25264-3_16
41. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, Boston (2011)
42. Thalheim, B.: On semantic issues connected with keys in relational databases permitting null values. Elektronische Informationsverarbeitung und Kybernetik **25**(1/2), 11–20 (1989)
43. Thalheim, B.: Fundamentals of cardinality constraints. In: Pernul, G., Tjoa, A.M. (eds.) ER 1992. LNCS, vol. 645, pp. 7–23. Springer, Heidelberg (1992). doi:10.1007/3-540-56023-8_3
44. Thalheim, B.: Entity-relationship modeling - foundations of database technology. Springer, Heidelberg (2000)
45. Thalheim, B.: Conceptual treatment of multivalued dependencies. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 363–375. Springer, Heidelberg (2003). doi:10.1007/978-3-540-39648-2_29
46. 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)