Structural and Computational Properties of Possibilistic Armstrong Databases

Seyeong Jeong, Haoming Ma, Ziheng Wei, and Sebastian Link[0000−0002−1816−2863]

School of Computer Science
The University of Auckland, Auckland 1010, New Zealand
s.link@auckland.ac.nz

Abstract. We investigate structural and computational properties of Armstrong databases for a new class of possibilistic functional dependencies. We establish sufficient and necessary conditions for a given possibilistic relation to be Armstrong for a given set of possibilistic functional dependencies. We then use the characterization to compute Armstrong databases for any given set of these dependencies. The problem of finding an Armstrong database is precisely exponential in the input, but our algorithm computes an output whose size is always guaranteed to be at most quadratic in a minimum-sized output. Extensive experiments indicate that our algorithm shows good computational behavior on average. As our possibilistic functional dependencies have important applications in database design, our results indicate that Armstrong databases can effectively support business analysts during the acquisition of functional dependencies that are meaningful in a given application domain.

Keywords: Sample data, Functional dependency, Possibility theory

1 Introduction

Background. Functional dependencies (FDs) are fundamental for understanding the structure and semantics of data, and have a fruitful history in database theory and practice. In a formal sense, FDs are to database constraints what Horn clauses are to logic [8]. An FD expresses that the values on some attributes uniquely determine the values on some other attributes. For example, every person has only one mother. Due to their ability to express desirable properties of many application domains, FDs have been used successfully for core data management tasks, including cleaning [16], design [10, 11], integration [5], exchange [15], modeling [14, 17], querying [9], and updating [21].

Motivation. Relational databases were developed for applications with certain data, including accounting, inventory and payroll [6]. Modern applications, such as information extraction, sensors, and data integration produce large volumes of uncertain data [4, 18]. As an example application, sufficiently simple to motive our research and explain our findings, we consider an employee who extracts information from web-sites about weekly project meetings in her company. This

is a typical case where information about the confidence of objects is useful, but probability distributions are unavailable. In such cases, qualitative approaches are attractive, for example possibility theory [2, 7].

Figure 1 shows a possibilistic relation (p-relation) 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. Attributes involve *Project*, storing projects with unique names, *Time*, for the weekday and start time, *Manager*, for the managers of the project that attend, and *Room*, for the unique name of a room. The employee classifies the possibility with which tuples occur in the relation according to their source. Tuples from the official web-site are assigned p-degree α_1, indicating they are fully possible, tuples from a project manager's web-site are assigned α_2, tuples from a project member's web-site get degree α_3, and tuples that originate from rumors are assigned p-degree α_4. Implicitly, any other tuple has p-degree α_5, indicating that it is impossible to occur. A different interpretation may result from already held, confirmed, requested, planned, and all other meetings. The p-degrees may have numerical interpretations, e.g. $1 > 0.75 > 0.5 > 0.25 > 0$. Either way, the employee has chosen 5 p-degrees to assign qualitative levels of uncertainty to tuples, with top degree α_1 and bottom degree α_5.

Proj	*Time*	*Mgr*	*Room*	*p-deg.*
Eagle	Mon, 9am	Ann	Aqua	α_1
Hippo	Mon, 1pm	Ann	Aqua	α_1
Kiwi	Mon, 1pm	Pete	Buff	α_1
Kiwi	Tue, 2pm	Pete	Buff	α_1
Lion	Tue, 4pm	Gill	Buff	α_1
Lion	Wed, 9am	Gill	Cyan	α_1
Lion	Wed, 11am	Bob	Cyan	α_2
Lion	Wed, 11am	Jack	Cyan	α_3
Lion	Wed, 11am	Pam	Lava	α_3
Tiger	Wed, 11am	Pam	Lava	α_4

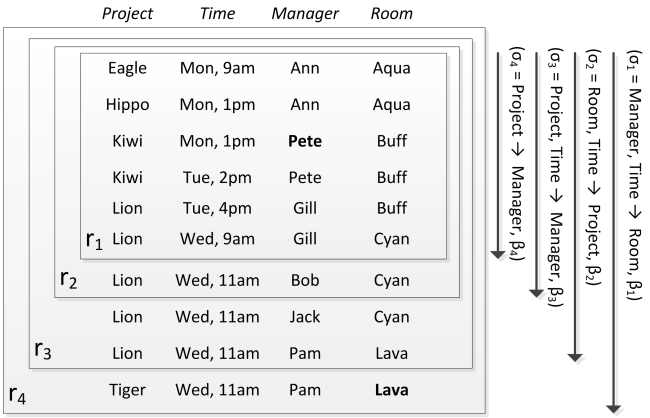
Fig. 1. Running example of a p-relation

Naturally, the assignment of p-degrees results in a linearly ordered chain of possible worlds: For $i = 1, \ldots, 4$, the relation r_i consists of tuples with p-degree α_i or higher, i.e. α_j with $j \leq i$. The p-degree of world r_i is α_i. In particular, fully possible tuples occur in every possible world, and are therefore also fully certain to occur. The possible worlds of the p-relation in Figure 1 are illustrated in Figure 2. Interestingly, p-degrees enable us to express classical FDs with different degrees of certainty (c-degree). For example, the FD $\sigma_1 = \mathit{Manager}, \mathit{Time} \rightarrow \mathit{Room}$ is satisfied by the world r_4, and thus holds in every possible world. Consequently, it is assigned the top c-degree, denoted by β_1. The smallest relation that violates $\sigma_2 = \mathit{Room}, \mathit{Time} \rightarrow \mathit{Project}$ is r_4, that is, the FD is assigned the second highest c-degree, β_2. The smallest relation that violates $\sigma_3 = \mathit{Project}, \mathit{Time} \rightarrow \mathit{Manager}$ is r_3, that is, the FD is assigned the third highest c-degree, β_3. The smallest relation that violates $\sigma_4 = \mathit{Project} \rightarrow \mathit{Manager}$ is r_2, and the FD thus holds with c-degree β_4. The FD $\mathit{Manager}, \mathit{Room} \rightarrow \mathit{Time}$ is violated even by the smallest possible world r_1, and is thus assigned the bottom c-degree $\beta_5 = \beta_{k+1}$.

Hence, the p-degree α_i of the smallest possible world r_i in which the FD is violated, determines the c-degree β_{k+2-i} with which the FD holds. A classical FD together with a c-degree was introduced as a possibilistic FD (pFD) in [12]. In the article, the possibilistic grounding of the pFDs was developed in depth and their difference in expressivity to previous work was explained. In addition, pFDs of this kind correspond to possibilistic Horn clauses, covering the classical equivalence between classical FDs and Horn clauses as the special case where only two p-degrees are present. The main motivation for pFDs is schema normalization [13].

In a nutshell, the possibilistic model [12] enables one to assign different degrees to the classical notion of data redundancy [22]. This makes it possible to define and compute different degrees of classical normal forms (Boyce-Codd and Third Normal Forms [1]), each eliminating/minimizing different degrees of data redundancy [13]. For example, to eliminate redundant data value occurrences in $r_4 - r_3$ (e.g. **Lava**), it suffices to normalize with pFDs of c-degree β_1. In contrast, to eliminate redundant data value occurrences in r_1 (e.g. **Pete**), one must normalize with pFDs of any c-degree. However, input to such normalization algorithms are sets of meaningful pFDs, as identified by teams of business analysts and database designers.

Fig. 2. Worlds of p-relation and scope of pFDs



The problem and Armstrong models. A challenging problem for design teams of the target database is therefore to identify the set of pFDs that are meaningful within the given application domain. For this purpose, the design team communicates with domain experts, and have to overcome a mismatch in expertise: The design team knows database concepts but not the domain, while domain experts know the domain but not database concepts. As humans learn a lot from good examples, it is likely that examples constitute an effective tool in helping design teams to identify more of the meaningful pFDs. Similar to the case of classical FDs, we view Armstrong databases as perfect examples [3, 11, 14]. In fact, an Armstrong database for a given set of constraints from a fixed class is a single database that satisfies all given constraints and violates all those

constraints from the class that are not implied by the given set. As such, an Armstrong database satisfies all the constraints currently perceived to be meaningful by the design team and explicitly violates every constraint not perceived to be meaningful. In particular, an Armstrong database for a given set of pFDs has the astonishing property that every pFD holds with the highest c-degree in the Armstrong database with which it is implied by the given pFD set. For example, the p-relation in Figure 1 is Armstrong for the four given pFDs shown in Figure 2. The FD *Manager, Time* $\rightarrow$ *Project* has c-degree β_2, as the smallest world which violates it is r_4. If this FD were to hold with full certainty in the application domain, i.e. c-degree β_1, domain experts would simply notice such violation in the Armstrong database. More generally, domain experts would notice when a meaningful constraint is violated by the Armstrong database (namely whenever it is incorrectly perceived as meaningless by the design team) and point this out to the design team. Our aim is to investigate structural and computational properties of Armstrong databases for pFDs, both in theory and in implementations and experiments. Our overarching goal is to improve the acquisition of requirements, i.e., to increase the number of meaningful pFDs that are recognized as such. This would generalize known results from the pure relational model of data [11], subsumed as the special case of our possibilistic model with two available p-degrees.

Contributions. Our contributions can be summarized as follows. (1) We establish sufficient and necessary conditions for a given p-relation to be Armstrong for a given set of pFDs, subsuming the characterization of classical Armstrong relations for FDs in terms of maximal, agree, and closed sets as a special case [3, 14]. (2) While the problem of computing an Armstrong p-relation for a given set of p-FDs in precisely exponential in the input, we establish an algorithm that is guaranteed to compute an Armstrong p-relation whose size is always guaranteed to be at most quadratic in the size of a minimum-sized Armstrong p-relation, again generalizing classical results [3, 14]. (3) Our algorithm is transferred into practice by an implementation. (4) Extensive experiments with our implementation show two extreme cases where output of exponential and logarithmic size in the input are produced, respectively. For randomly created inputs and fixed schema sizes, output sizes display logarithmic growth and output times display constant behavior in the number of available p-degrees, and for fixed numbers of available p-degrees, output sizes and times both display low-degree polynomial growth in the size of the schema. Our results provide a technical platform for using Armstrong databases during the requirements acquisition of pFDs.

Organization. Section 2 discusses background from the relational model. Our possibilistic model is defined in Section 3. In Section 4 we characterize Armstrong p-relations for pFDs, and show how to compute them. Our tool is briefly discussed in Section 5. Section 6 presents the results of our experiments. Finally, we conclude in Section 7 and briefly discuss future work. Most proofs have been omitted to meet space requirements.

2 Armstrong relations for functional dependencies

FDs are probably the most studied class of constraints, due to their expressivity, computational behavior, and impact on practice. This applies to most of the existing data models, ranging over conceptual, relational, object-relational, Web, graph, and uncertain models. FDs were already introduced in Codd's seminal paper [6]. In this section we give a concise summary about the structural and computational properties of Armstrong relations for classical FDs from the relational model. Subsequently, these will be extended to our possibilistic model.

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 $\prod_{A \in R} dom(A)$. For $X \subseteq R$ we denote by $t(X)$ the *projection* of t on X. An *relation* over R is a finite set r of tuples over R. In our example, $R = \textsc{Web}$ has attributes *Project*, *Time*, *Manager*, and *Room*. Figure 2 shows examples of relations and their tuples. For attribute subsets X, Y we write XY for their set union, and identify singletons with their element.

A *functional dependency* (FD) over R is an expression $X \to Y$ where $X, Y \subseteq R$. A relation r *satisfies* $X \to Y$ iff for all $t, t' \in r$, $t(X) = t'(X)$ implies that $t(Y) = t'(Y)$. In Figure 2, relation r_3 satisfies the FDs *Manager*, *Time* $\to$ *Room* and *Room*, *Time* $\to$ *Project*, but not the FD *Project*, *Time* $\to$ *Manager*. A relation r satisfies a given FD set Σ iff r satisfies all $\sigma \in \Sigma$. For a set $\Sigma \cup \{\varphi\}$ of FDs, Σ *implies* φ iff every relation that satisfies Σ also satisfies φ. If $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4$ from Figure 2, then Σ implies the FD *Manager*, *Time* $\to$ *Project*, but Σ does not imply the FD *Manager*, *Room* $\to$ *Time*.

A relation r is *Armstrong* for Σ iff r satisfies Σ and for every FD φ not implied by Σ, r does not satisfy φ. Consequently, an Armstrong relation for Σ satisfies an FD φ if and only if φ is implied by Σ. In Figure 2, the relation r_1 is Armstrong for the FD set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. As r_1 satisfies the FD *Manager*, *Time* $\to$ *Project*, this FD is implied by Σ; and as r_1 violates *Manager*, *Room* $\to$ *Time*, this FD is not implied by Σ. The left of Figure 3 shows also an Armstrong relation for Σ. In fact, up to renaming, it is the same relation as r_1.

We are now introducing further concepts that will allow us to summarize the characterization of Armstrong relations for classical FDs. For a given relation r and distinct tuples $t, t' \in r$, the *agree set* $ag(t, t')$ of t, t' is the set of attributes $A \in R$ such that $t(A) = t'(A)$ holds. The *agree set* of r is the set of agree sets for all pairs of distinct tuples $t, t' \in r$. For example, the agree set of the first two tuples in relation r_1 of Figure 2 is $\{$*Manager*, *Room*$\}$, and the agree set of r_1 consists of the following attribute sets: $\{$*Manager*, *Room*$\}$, $\emptyset$, $\{$*Time*$\}$, $\{$*Project*, *Manager*, *Room*$\}$, $\{$*Room*$\}$, and $\{$*Project*, *Manager*$\}$.

A set X of attributes is *closed* under Σ iff $X = \{A \in R \mid \Sigma \text{ implies } X \to A\}$. For R, Σ, $cl_\Sigma(R)$ is the set of attribute sets closed under Σ. For example, if $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ from Figure 2, then the following attribute sets are closed: $\emptyset$, $\{$*Time*$\}$, $\{$*Manager*$\}$, $\{$*Room*$\}$, $\{$*Project*, *Manager*$\}$, $\{$*Manager*, *Room*$\}$, $\{$*Project*, *Manager*, *Room*$\}$, and $\{$*Project*, *Time*, *Manager*, *Room*$\}$ itself.

A set X of attributes is *maximal* for an attribute A under Σ iff Σ does not imply $X \to A$, but for all $B \in R - (XA)$, Σ implies $XB \to A$. That is,

X is maximal for A with the property that X does not functionally determine A. The maximal sets for R under Σ is the union of the maximal sets for each attribute of R under Σ. For example, if $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ from Figure 2, then the maximal sets for *Project* are $\{Manager, Room\}$ and $\{Time\}$, for *Time* it is $\{Project, Manager, Room\}$, for *Manager* they are $\{Time\}$ and $\{Room\}$, and for *Room* they are $\{Project, Manager\}$ and $\{Time\}$.
The significance of these concepts is embodied in the following theorem. The second subset relationship actually ensures that r satisfies Σ, while the first subset relationship ensures that r does not satisfy any FD not implied by Σ.

Theorem 1. *[3, 14] A relation r over relation schema R is Armstrong for an FD set Σ over R if and only if $max_\Sigma(R) \subseteq ag(r) \subseteq cl_\Sigma(R)$.*

Based on our examples before and Theorem 1 it follows immediately that the relation r_1 is Armstrong for the set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ from Figure 2. The problem of computing an Armstrong relation for a given FD set is precisely exponential in the input [3]. However, as every maximal attribute set is also closed [14], Theorem 1 can be used to construct an Armstrong relation for a given set Σ of FDs by i) computing the set of maximal sets for R under Σ, and ii) creating a relation that starts with a single tuple t and then inserts for each maximal set X for R under Σ a new tuple t' that has agree set X with the previous tuple. Following this construction for the set $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ from Figure 2, we could introduce tuples which have agree set with the previous tuple in the following sequence of maximal sets: $\{Manager, Room\}$, $\{Time\}$, $\{Project, Manager, Room\}$, $\{Room\}$, and $\{Project, Manager\}$. Up to renaming, this would lead to the relation r_1 shown in Figure 2.

It has been shown that this algorithm produces an Armstrong relation that is always guaranteed to have a number of tuples that is at most quadratic in that of a minimum-sized Armstrong relation [3]. The main complexity concerns the computation of the maximal sets. Mannila and Räihä have established an iterative algorithm MaxFam [14] which takes as input a relation schema R and FD set Σ over R and computes for each $A \in R$ the set $max_\Sigma(A)$. In fact, MaxFam refines the set of maximal sets for R by adding one FD of the input at a time. That is, if R has n attributes, the set $max_\emptyset(R)$ consists of all $n-1$ element subsets of R, and the algorithm then refines these sets by computing $max_{\Sigma' \cup \{\sigma\}}(R)$ in one iteration from $max_{\Sigma'}(R)$ until $\Sigma' \cup \{\sigma\} = \Sigma$. The details are given in [14] but not of importance to the current article.

3 Possibilistic Functional Dependencies

We summarize briefly the definition of the possibilistic model from [12].

Uncertain relations are modeled by assigning to each tuple some degree of possibility with which the tuple occurs in the relation. Formally, we have a *possibility scale*, or p-scale, that is, a strict linear order $\mathcal{S} = (S, <)$ with $k+1$ elements. We write $\mathcal{S} = \{\alpha_1, \ldots, \alpha_{k+1}\}$ 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

tuples that are 'fully possible' while α_{k+1} is reserved for tuples that are 'impossible' to occur in a relation. 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 relations use two p-degrees, i.e., $k = 1$.

A *possibilistic relation schema* $(R, \mathcal{S})$, or p-relation schema, consists of a relation schema R and a p-scale $\mathcal{S}$. A *possibilistic relation*, or p-relation, over $(R, \mathcal{S})$ consists of a relation r over R, and a function *Poss* that assigns to each tuple $t \in r$ a p-degree $Poss(t) \in \mathcal{S} - \{\alpha_{k+1}\}$. We sometimes omit *Poss* when denoting a p-relation. Figure 1 shows a p-relation over $(\textsc{Web}, \mathcal{S} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\})$, where Web consists of the four attributes *Project*, *Time*, *Manager* and *Room*.

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, $r_i = \{t \in r \mid Poss(t) \geq \alpha_i\}$. Indeed, we have $r_1 \subseteq r_2 \subseteq \cdots \subseteq r_k$. If $t \notin r_k$, then $Poss(t) = \alpha_{k+1}$. Every tuple that is 'fully possible' occurs in every possible world, and is therefore also 'fully certain'. Hence, relations are a special case of p-relations. Figure 2 shows the possible worlds $r_1 \subsetneq r_2 \subsetneq r_3 \subsetneq r_4$ of the p-relation of Figure 1.

Similar to the scale $\mathcal{S}$ of p-degrees α_i for tuples, we use a scale $\mathcal{S}^T$ of certainty degrees β_j, or c-degrees, for 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 *marginal certainty* $c_r(\sigma)$ by which the FD $\sigma = X \to Y$ holds on the p-relation r is either the top degree β_1 if σ 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 σ is violated, that is,

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

We can now define the semantics of pFDs. Let $(R, \mathcal{S})$ denote a p-relation schema. A *possibilistic functional dependency* (pFD) over $(R, \mathcal{S})$ is an expression $(X \to Y, \beta)$ where $X \to Y$ denotes an FD over R and $\beta \in \mathcal{S}^T$. A p-relation $(r, Poss)$ over $(R, \mathcal{S})$ satisfies the pFD $(X \to Y, \beta)$ if and only if $c_r(X \to Y) \geq \beta$.

For example, the p-relation r from Figure 1 satisfies the pFDs (σ_i, β_i) from Figure 2 for $i = 1, \ldots, 4$. In fact, it is true for r that $c_r(\sigma_i) = \beta_i$ for $i = 1, \ldots, k$.

4 Possibilistic Armstrong relations

We establish structural and computational properties of Armstrong p-relations for pFDs. In particular, we will first generalize Theorem 1 from classical FDs to pFDs, and then utilize the characterization to develop an algorithm that computes an Armstrong p-relation for any given set of pFDs. While the problem of finding an Armstrong p-relation remains precisely exponential in the input, our algorithm always produces an output of a size that is at most quadratic in that of a minimum-sized Armstrong p-relation for the input.

4.1 Structural characterization

Similar to the elegant classical characterization of Theorem 1 we would like to have sufficient and necessary conditions to decide when a given p-relation is Armstrong for a given set of pFDs.
By definition, a p-relation $(r, Poss)$ over $(R, \{\alpha_1, \ldots, \alpha_{k+1}\})$ is Armstrong for the pFD set Σ if and only if for all $i = 1, \ldots, k$, and for all $X \to Y$ over R,

$$(r, Poss) \text{ satisfies } (X \to Y, \beta_i) \text{ if and only if } \Sigma \text{ implies } (X \to Y, \beta_i).$$

Now, the definition of satisfaction means that $(r, Poss)$ satisfies $(X \to Y, \beta_i)$ if and only if the world r_{k+1-i} satisfies the FD $X \to Y$. For $i = 1, \ldots, k$, and the pFD set Σ, the β_i-cut of Σ is the FD set $\Sigma_i = \{X \to Y \mid \exists j \leq i (X \to Y, \beta_j) \in \Sigma\}$. It has been shown that Σ implies the pFD $(X \to Y, \beta_i)$ if and only if the FD set Σ_i implies the FD $X \to Y$ [12].

Consequently, $(r, Poss)$ is Armstrong for Σ if and only if for all $i = 1, \ldots, k$, the world r_{k+1-i} is an Armstrong relation for the FD set Σ_i. Using the characterization from Theorem 1, we arrive at the following result.

Theorem 2. *Let Σ be a set of pFDs over p-relation schema $(R, \mathcal{S})$ with $|\mathcal{S}| = k + 1$. A p-relation $(r, Poss)$ over $(R, \mathcal{S})$ is Armstrong for Σ if and only if for all $i = 1, \ldots, k$, the world r_{k+1-i} is an Armstrong relation for the β_i-cut Σ_i of Σ. That is, if $max_{\Sigma_i}(R) \subseteq ag(r_{k+1-i}) \subseteq cl_{\Sigma_i}(R)$ holds for all $i = 1, \ldots, k$.* □

For our running example, we can verify with Theorem 2 that the p-relation in Figure 1 is Armstrong for the pFD set Σ shown in Figure 2. We have already seen that r_1 is an Armstrong relation for $\Sigma_4 = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$. The following table shows the maximal sets $\max_{\Sigma_i}(A)$ for all $i = 1, \ldots, 4$. Instead of writing $max_{\Sigma_i}(A)$, we simply write $max_i(A)$ to ease notation, and we use the leading characters to denote our attributes.

A	$\max_1(A)$	$\max_2(A)$	$\max_3(A)$	$\max_4(A)$
Project	$\{MTR\}$	$\{MR, T\}$	$\{MR, T\}$	$\{MR, T\}$
Time	$\{MRP\}$	$\{MRP\}$	$\{MRP\}$	$\{MRP\}$
Manager	$\{PTR\}$	$\{PTR\}$	$\{PR, T\}$	$\{R, T\}$
Room	$\{MP, PT\}$	$\{MP, PT\}$	$\{MP, T\}$	$\{MP, T\}$

Knowing these maximal sets, we can verify by Theorem 2 that r_2 is Armstrong for $\Sigma_3 = \{\sigma_1, \sigma_2, \sigma_3\}$, r_3 is Armstrong for $\Sigma_2 = \{\sigma_1, \sigma_2\}$, and r_4 is Armstrong for $\Sigma_1 = \{\sigma_1\}$. Note that only the maximal sets in red font are realized by the relations. This is a consequence of the chain of possible worlds, since the maximal sets realized in some world are also realized in every world that contains it. In fact, the tuple in $r_2 - r_1$ realizes the maximal set PR for Σ_3 together with the previous tuple, the two tuples in $r_3 - r_2$ realize the maximal sets PTR and PT for Σ_2 together with their corresponding previous tuples, and the tuple in $r_4 - r_3$ realizes the maximal set MR for Σ_1.

Algorithm 1 Armstrong p-relation

Require: Set Σ of pFDs over p-schema $(R, \{\beta_1, \ldots, \beta_k\})$
Ensure: Armstrong p-relation for Σ
1: $\Sigma_0 \leftarrow \emptyset$;
2: **for all** $A \in R$ **do** $\max_0(A) \leftarrow \{R - \{A\}\}$;▷ Maximal set families for empty FD set
3: **for** $i = 1$ to k **do** $\{\max_i(A)\}_{A \in R} \leftarrow$ MaxFam$(R, \Sigma_i - \Sigma_{i-1}, \{\max_{i-1}(A)\}_{A \in R})$ ▷ Max set families for next β-cut
4: **for all** $A \in R$ **do** $t_0(A) \leftarrow c_{A,0}$; ▷ Initial tuple
5: $j \leftarrow 0$; $r \leftarrow \{t_0\}$; $Poss_r(t_0) = \alpha_1$; $\max(R) \leftarrow \emptyset$; ▷ Some initializations
6: **for** $i = k$ downto 1 **do**
7: $\max_i(R) \leftarrow \max_i(R) - \max(R)$ ▷ Remove already realized max sets
8: **for all** $W \in \max_i(R)$ **do** ▷ Realize next max set as agree set
9: $j \leftarrow j + 1$;
10: **for all** $A \in R$ **do**
11: **if** $A \in W$ **then** $t_j(A) \leftarrow t_{j-1}(A)$; ▷ t_j and t_{j-1} agree on A
12: **else** $t_j(A) \leftarrow c_{A,j}$; ▷ Unique value for t_j on A
13: $Poss_r(t_j) \leftarrow \alpha_{k+1-i}$ ▷ t_j gets possibility α_{k+1-i}
14: $r \leftarrow r \cup \{t_j\}$;
15: $\max(R) \leftarrow \max(R) \cup \max_i(R)$ ▷ Mark elements from $\max_i(R)$ as realized
16: **return**(r);

4.2 Computational characterization

We establish an algorithm that computes an Armstrong p-relation for any given set of pFDs. By Theorem 2 we compute the maximal set families $\{\max_i(A)\}_{A \in R}$, and realize them with tuples of p-degrees α_{k+1-i}, for $i = 1, \ldots, k$. Algorithm 1 is a high-level description of this strategy.

We start with the maximal set families under the empty FD set in line (2). Since the β_i-cuts form a chain $\Sigma_1 \subseteq \Sigma_2 \cdots \subseteq \Sigma_k$ of classical FD sets, and the classical algorithm for computing maximal sets is iterative, we can compute the maximal set families $\{\max_i(A)\}_{A \in R}$ by refining the maximal set families $\{\max_{i-1}(A)\}_{A \in R}$ based on the "new" FDs in $\Sigma_i - \Sigma_{i-1}$. This is achieved by line (3). The call MaxFam$(R, \Sigma_i - \Sigma_{i-1}, \{\max_{i-1}(A)\}_{A \in R})$ invokes the classical procedure [14], but fetches the maximal set families for each p-degree.

We then begin to realize the maximal sets as agree sets, starting with a base tuple of p-degree α_1 in lines (4,5). The for-loop between lines (6) and (15) realizes the maximal sets from lower to higher c-degrees β_i (line 6) by inserting a single new tuple (line 14) for each each unrealized maximal set, lines (7,8). The new tuple has agree set with its predecessor tuple on the current maximal agree set, lines (8-12) and is assigned p-degree α_{k+1-i} in line (13). Line (15) marks the maximal set as realized.

As every maximal set $X \in max_i(R)$ is also closed under Σ_i [14], Theorem 2 shows that Algorithm 1 is correct.

Theorem 3. *On input* $((R, \{\beta_1, \ldots, \beta_k\}), \Sigma)$*, Algorithm 1 computes a p-relation that is Armstrong for* Σ*.* □

If we apply Algorithm 1 to our p-relation schema (WEB, $\{\alpha_1, \ldots, \alpha_5\}$) and the pFD set Σ from Figure 2, it will compute an Armstrong p-relation such as the one shown in Figure 1, up to renaming.

4.3 Complexity results

We recall what we mean by precisely exponential [3]. Firstly, it means that there is an algorithm for computing an Armstrong p-relation, given a set Σ of pFDs, where the running time of the algorithm is exponential in Σ. Secondly, it means that there is a set Σ of pFDs 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. The exponential lower bound is retained from the special case where $k = 1$, and the input family would be $exp_n := ((R_n = \{A_1, \ldots, A_{2n}, B\}, \{\alpha_1, \alpha_2\}), \Sigma_n^{exp})$ where $\Sigma_n^{exp} := \bigcup_{i=1}^{n} \{\{(A_{2i-1}, A_{2i}\} \to B, \beta_1)\}$. The upper bound follows immediately from the fact that we are able to apply the classical exponential-time algorithm for the maximal set computation.

Theorem 4. *The complexity of finding an Armstrong p-relation, given a set Σ of pFDs, is precisely exponential in Σ.* □

The case exp_n shows a negative extreme case in which the number of tuples in the output is exponential in the input size. However, there are also positive extreme cases in which the number of tuples in the output is logarithmic in the input size. Such a case is given by $log_n := ((R_n, \{\alpha_1, \alpha_2\}), \Sigma_n^{log})$ where $\Sigma_n^{log} := \bigcup_{i=1}^{n} \{(\{X_1, \ldots, X_n\} \to B, \beta_1) \mid \forall i = 1, \ldots, n, X_i \in \{A_{2i-1}, A_{2i}\}\}$.

Despite the worst-case exponential time complexity, our algorithm makes conservative use of resources. An Armstrong p-relation for Σ is minimum-sized if there is no Armstrong p-relation for Σ with a fewer number of tuples. Let $max_\Sigma(R)$ denote the maximal sets that need to be realized in any Armstrong p-relation r for a pFD set Σ. Due to Theorem 2 it follows that $|max_\Sigma(R)|$ is bounded by $|ag(r) = ag(r_k)|$, and that $|ag(r)|$ is bounded by $\binom{|r|}{2}$. From $|ag(r)| \leq \binom{|r|}{2}$ follows that $\sqrt{(1 + 8 \cdot |max_\Sigma(R)|)/2} \leq |r|$, and Algorithm 1 shows that $|r = r_k| \leq |max_\Sigma(R)| + 1$. If the size of a p-relation is the number of its tuples, we thus obtain the following result.

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

5 The Tool

We have transferred our findings into a prototype implementation[1] that design teams can use to compute Armstrong p-relations for any given set of pFDs.

[1] https://www.dropbox.com/s/fciy01597tgxnfu/Possibilistic-Armstrong-Calculator.exe

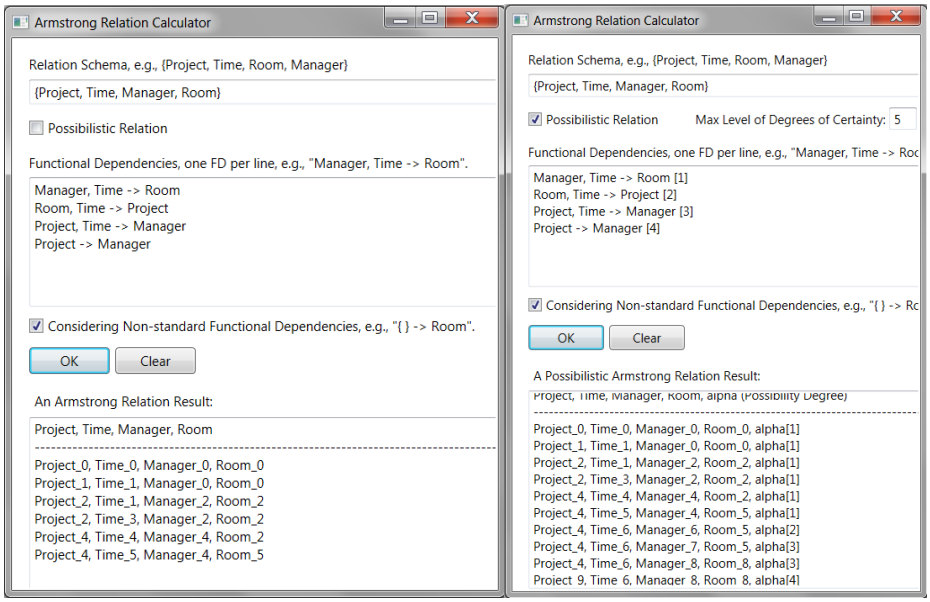


Fig. 3. GUI for classical and possibilistic case of running example

Figure 3 shows the graphical user interface for the tool. Users can declare the input in the form of an attribute set, and choose whether they want to compute a classical or a possibilistic Armstrong database. In the former case, they can simply enter an FD set. In the latter case, they can define how many p-degrees are available, i.e., specify k, and then enter a pFD set. Finally, users can choose whether to consider non-standard (p)FDs, whose left-hand attribute set is empty. The screenshots in Figure 3 show the GUI for our running example (right), and for the running example after "forgetting" the possibilistic information (left).

6 Experiments

We report on some experiments that illustrate the extreme cases exp_n and log_n, as well as the average case performance of Algorithm 1.

Extreme Cases. Figures 4 and 5 illustrate the experiments for the extreme cases of exponential and logarithmic output size, respectively. It shows, in particular, that even under extreme circumstances the algorithm performs well. For example, Figure 4 shows that even an Armstrong p-relation with 65,000 tuples can be computed in less than 90 seconds, while Figure 5 shows that even with huge input sizes, the algorithm still returns a result within 2hrs. However, applying the algorithm to instances of exp_n beyond $n = 25$ is not feasible.

Average Cases. We studied average behavior by applying Algorithm 1 to random input. For each fixed number $n = 5, \ldots, 15$ of attributes, and each fixed number $k = 1, \ldots, 10$ of c-degrees, we generated 250 input sets Σ, each of which

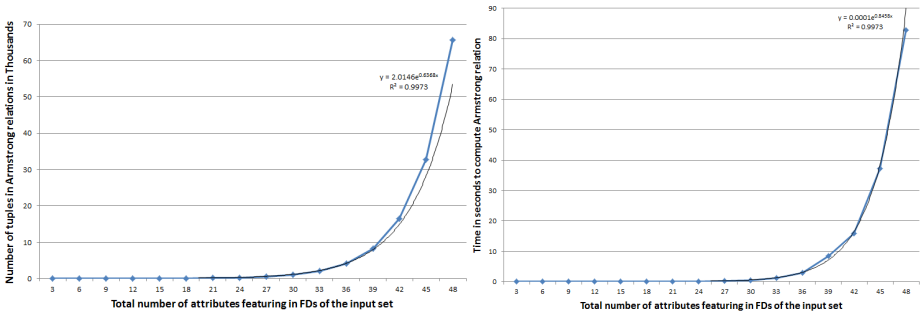


Fig. 4. Output sizes & times for exponential case

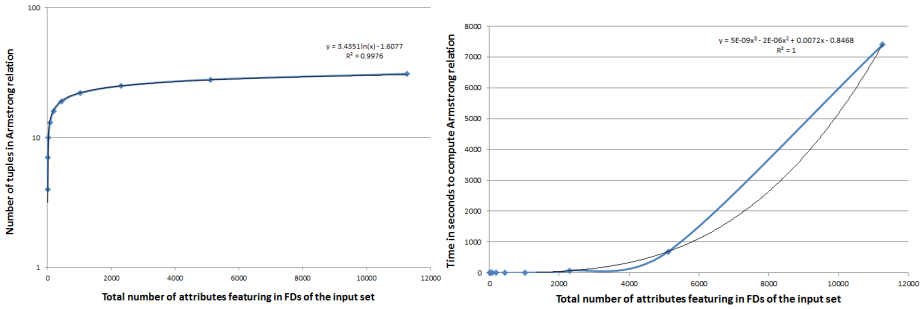


Fig. 5. Output sizes & times for logarithmic case

had between n and $n^2/2$ pFDs with three attributes on average, and a randomly assigned p-degree between 1 and k. With n and k as the x- and y-axes, respectively, the z-axis was then either the average number of tuples in the output, or the time it took to compute it.

Figure 6 shows the average output size and time in the number k of p-degrees, parameterized by the schema size n. For fixed n, there is logarithmic growth of the output size and constant time in k. The size growth results from a significant number of maximal sets being realized by a small k. The computation time is agnostic to k because each FD is visited once, irrespective of its p-degree.

Figure 7 shows the average output size and time in the schema size n, parameterized by k. For fixed k, the output size and times are both low-degree polynomial in n. Extreme cases are therefore considered to be the exception.

7 Conclusion and Future Work

We have established structural and computational properties of Armstrong databases for a class of pFDs that generalize classical FDs. The class has important applications in database schema design, where the input requires a set of meaningful pFDs. Our theoretical and experimental analysis suggests that the compu-

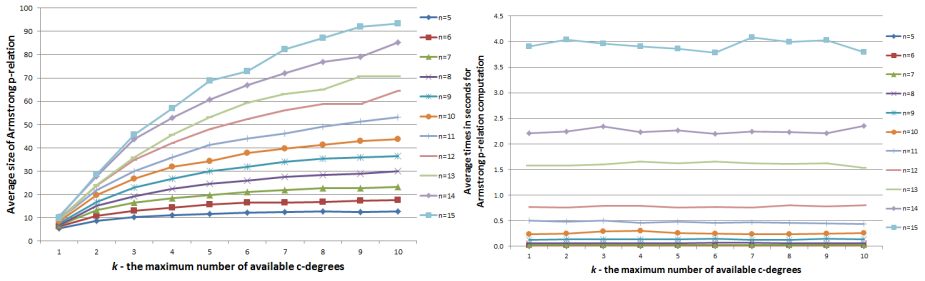


Fig. 6. Average output sizes & times for fixed schema sizes n in number k of p-degrees

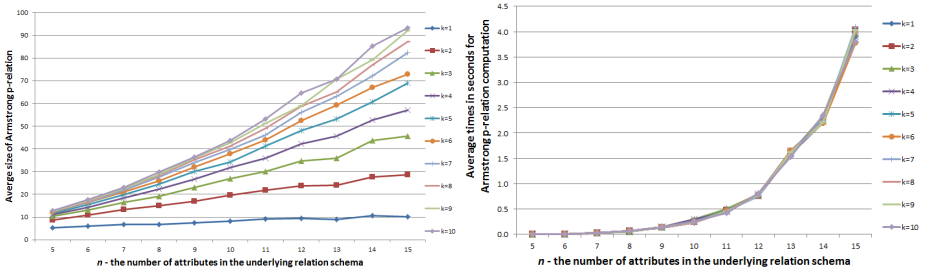


Fig. 7. Average output sizes & times for fixed number k of p-degrees in schema size n

tational properties of Armstrong databases are supportive for their target use in the acquisition process of the pFDs. Due to the equivalence between pFDs and possibilistic Horn clauses [12], our results also transfer to Armstrong models of possibilistic Horn clauses, which have important applications in abductive and deductive reasoning, knowledge approximation and compilation [20]. Note that the results of this article have been extended to the combined class of pFDs and possibilistic cardinality constraints [19].

For future work it would be interesting to conduct empirical studies to confirm the effectiveness of our tool for the acquisition process. Here, we may use our tool to compute Armstrong databases on the fly, and measure the impact of their use on recognizing those pFDs that are meaningful for a given application domain. It is also interesting to combine the possibilistic approach of this paper with the recently introduced embedded uniqueness constraints [24] and embedded functional dependencies [25, 23]. Since these embedded dependencies address data with missing values, combining them with our possibilistic approach would mean that uncertain data with missing values can be addressed.

References

1. Arenas, M.: Normalization theory for XML. SIGMOD Record **35**(4), 57–64 (2006)
2. Balamuralikrishna, N., Jiang, Y., Koehler, H., Leck, U., Link, S., Prade, H.: Possibilistic keys. Fuzzy Sets Syst. **376**, 1–36 (2019)

3. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. J. ACM **31**(1), 30–46 (1984)
4. Brown, P., Link, S.: Probabilistic keys. IEEE Trans. Knowl. Data Eng. **29**(3), 670–682 (2017)
5. Calì, A., Calvanese, D., Lenzerini, M.: Data integration under integrity constraints. In: Seminal Contributions to Information Systems Engineering, 25 Years of CAiSE, pp. 335–352 (2013)
6. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
7. Dubois, D., Prade, H.: Possibility theory and its applications: Where do we stand? In: Springer Handbook of Computational Intelligence, pp. 31–60 (2015)
8. Fagin, R.: Horn clauses and database dependencies. J. ACM **29**(4), 952–985 (1982)
9. Johnson, D.S., Klug, A.C.: Testing containment of conjunctive queries under functional and inclusion dependencies. J. Comput. Syst. Sci. **28**(1), 167–189 (1984)
10. Köhler, H., Link, S.: SQL schema design: foundations, normal forms, and normalization. Inf. Syst. **76**, 88–113 (2018)
11. Langeveldt, W.D., Link, S.: Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies. Inf. Syst. **35**(3), 352–374 (2010)
12. Link, S., Prade, H.: Possibilistic functional dependencies and their relationship to possibility theory. IEEE Trans. Fuzzy Systems **24**, 1–7 (2016)
13. Link, S., Prade, H.: Relational database schema design for uncertain data. Inf. Syst. **84**, 88–110 (2019)
14. Mannila, H., Räihä, K.J.: Design by example: An application of Armstrong relations. J. Comput. Syst. Sci. **33**(2), 126–141 (1986)
15. Marnette, B., Mecca, G., Papotti, P.: Scalable data exchange with functional dependencies. Proc. VLDB Endow. **3**(1), 105–116 (2010)
16. Prokoshyna, N., Szlichta, J., Chiang, F., Miller, R.J., Srivastava, D.: Combining quantitative and logical data cleaning. Proc. VLDB Endow. **9**(4), 300–311 (2015)
17. Ram, S.: Deriving functional dependencies from the entity-relationship model. Commun. ACM **38**(9), 95–107 (1995)
18. Roblot, T., Hannula, M., Link, S.: Probabilistic cardinality constraints - validation, reasoning, and semantic summaries. VLDB J. **27**(6), 771–795 (2018)
19. Roblot, T., Link, S.: Cardinality constraints and functional dependencies over possibilistic data. Data Knowl. Eng. **117**, 339–358 (2018)
20. Selman, B., Kautz, H.A.: Knowledge compilation and theory approximation. J. ACM **43**(2), 193–224 (1996)
21. Tan, H.B.K., Zhao, Y.: Automated elicitation of functional dependencies from source codes of database transactions. Information & Software Technology **46**(2), 109–117 (2004)
22. Vincent, M.: Semantic foundations of 4NF in relational database design. Acta Inf. **36**(3), 173–213 (1999)
23. Wei, Z., Hartmann, S., Link, S.: Discovery algorithms for embedded functional dependencies. In: Maier, D., Pottinger, R., Doan, A., Tan, W., Alawini, A., Ngo, H.Q. (eds.) Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020. pp. 833–843. ACM (2020)
24. Wei, Z., Leck, U., Link, S.: Discovery and ranking of embedded uniqueness constraints. Proc. VLDB Endow. **12**(13), 2339–2352 (2019)
25. Wei, Z., Link, S.: Embedded functional dependencies and data-completeness tailored database design. Proc. VLDB Endow. **12**(11), 1458–1470 (2019)