Algorithmic Perspective of Strongly Possible Keys and Functional Dependencies

Munqath Alattar[1], Attila Sali[2,3]
[1]ITRDC, University of Kufa, Iraq
[2]Alfréd Rényi Institute of Mathematics, Budapest, Hungary
[3]Department of Computer Science, Budapest University of Technology and Economics, Budapest, Hungary
E-mail: munqith.alattar@uokufa.edu.iq, sali.attila@renyi.hu

Keywords: Functional dependencies, database key, missing values, approximate dependencies, approximate keys

Received: February 3, 2023

It is common to encounter missing values in database tables. For an incomplete table, a possible world can be obtained by replacing any missed value with a value from the attribute (infinite) domain. A possible key (possible functional dependency) is satisfied in an incomplete table "T" if there exists a possible world of "T" that satisfies the key (the functional dependency) constraint. If all possible worlds of "T" satisfy the key (functional dependency), then we say that "T" satisfies a certain key (functional dependency). The concept of strongly possible worlds was introduced recently that considers only the active domain (the set of values that are already appearing in each attribute in the table), in a way that a strongly possible world is obtained by replacing any missing value with a value from the corresponding attributes active domain. So, a strongly possible key spKey (functional dependency spFD) is satisfied by a table "T" if there exists a strongly possible world that satisfies the key (functional dependency). In this paper, we investigate the approximation measures of spKeys and spFDs when the strongly possible constraint is not satisfied by a given table. The measure g_3 represent the ratio of the minimum number of tuples that need to be removed so that the table satisfies the constraint. We introduce a new measure g_5, which is the ratio of the minimum number of tuples to be added to the table so the result satisfies the constraint. Where adding new tuples with new values will extend the active domain. We prove that g_3 is an upper bound of g_5 for a constraint in a table. Furthermore, g_3 and g_5 are independent of each other, where there exist tables of some large number of tuples that satisfy $g_3 - g_5 = \frac{p}{q}$ for any rational number $0 \leq \frac{p}{q} < 1$. We study the complexity of determining these approximate measures.

Povzetek: Raziskane so aproksimacijske mere za močne možne ključe (spKeys) in funkcionalne odvisnosti (spFDs). Predstavljen je nov kazalnik g5, ki ocenjuje dodajanje n-tork za izpolnitev omejitev, s poudarkom na računski zapletenosti in uporabnosti v podatkovnih bazah.

1 Introduction

There are many reasons that may cause the missingness of values in the industrial and research databases, such as data maintenance, errors during data entry, surveys, and so on [8]. Imputation and deletion are the main two approaches to handle the missing values problem in a database. Imputation (assignment of a value to the occurrence of any `NULL`) is the main approach to handle missing value problem [13]. In [3], an imputation method was introduced that replaces the missing value using only the shown information in the table (which is defined as the active domain of that attribute), and we call the complete table achieved by this method a *strongly possible world*. The reason we consider only the values that are shown in a table for imputation is that it is not always proper to consider values that are not shown in the table.

Using this concept, strongly possible keys (spKeys) and strongly possible functional dependencies (spFDs) were defined in [5, 4] as new key and functional dependency constraints that are satisfied by replacing missing values (`NULL`) with values that are shown in the corresponding attribute. The formal definitions of spKeys and spFDs are provided in Section 2. In this paper, we continue the work started in [5] which introduced an approximation measure that calculates how close a set of attributes in a table can form a key if they are not. An active domain may not contain enough values to replace `NULL`s to make all resulting tuples distinct from each other on a key K, so, removing some tuples can be a solution. This paper studies an approximation measures of spKeys and spFDs by adding tuples not removing them, where adding a tuple with new unique values that are not shown before adds more values to the attributes' active domains and this may satisfy some unsatisfied constraints. For example, in the Cars Types table in Table 1, Car_Model and $DoorNo$ attributes are planned to be a key but it is not satisfied as $sp\langle Car\_Model, DoorNo\rangle$ in the table. Removing two tuples can satisfy the key

$sp\langle Car\_Model, DoorNo\rangle$, while adding one new tuple with a distinct door number value would satisfy the key. From some point of views, adding one tuple with new values is better than removing already existing two tuples. On the other hand, it is common that car models and the number of doors together determine the engine type, while the spFD $(Car\_Model, DoorNo) \rightarrow_{sp} Engine\_Type$ is violated in the table. So, adding a single tuple to the table with a new value for the attribute $DoorNo$ can satisfy the spFD $(Car\_Model, DoorNo) \rightarrow_{sp} Engine\_Type$, instead of taking off two tuples from the table.

Table 1: Types of Cars

Car_Model	Door No	Engine_Type
BMW	4 doors	⊥
BMW	⊥	e
Ford	⊥	V8
Ford	⊥	V6

2 Definitions

In a relation schema $R = \{A_1, A_2, \ldots A_n\}$, for an attribute A_i, let the domain of A_i be as $D_i = dom(A_i)$ for $i = 1,2,\ldots n$ and represent the set of values that are possible for each attribute $A_i \in R$. Then, for a subset $X \subseteq R$, the domain of X is $D_X = \prod_{\forall A_i \in K} D_i$.

An instance T over the relation R such that $T = (t_1,t_2,\ldots t_s)$ is a set of tuples such that every tuple represent a function $t : R \rightarrow \bigcup_{A_i \in R} dom(A_i)$ where $t[A_i] \in dom(A_i)$ for all A_i in R. For a set of tuples, several occurrences of the same tuple are allowed as we use the *bag semantics* concept. As the order of the tuples is not relevant, the *multiset of tuples* is considered as an instance. Let $t_r[X]$ represent the restriction of t_r to the attribute set $X \subset R$ for a tuple $t_r \in T$.

Let $\perp$ be a symbol in each attribute's domain that represents a missing value. Let V be a set of attributes, then t_r is V-total if $\forall A \in V$, $t_r[A] \neq \perp$. Furthermore, t_r is called a total tuple if it is R-total. Two tuples t_1 and t_2 are called *weakly similar* on $X \subseteq R$ written as $t_1[X] \sim_w t_2[X]$ defined by Köhler et.al. [11] if:

$$\forall A \in X \quad (t_1[A] = t_2[A] \text{ or } t_1[A] = \perp \text{ or } t_2[A] = \perp).$$

We use the notion $t_1 \sim_w t_2$ if t_1 and t_2 are weakly similar on each attribute in R.

In addition to that, t_1 and t_2 are called *strongly similar* on $X \subseteq R$ denoted by $t_1[X] \sim_s t_2[X]$ if

$$\forall A \in X \quad (t_1[A] = t_2[A] \neq \perp).$$

Let $T = (t_1, t_2, \ldots t_s)$ be an instance over the relation R, and $T' = (t'_1, t'_2, \ldots t'_s)$ is a *possible world* of T, such that T' is NULL-free and $t_i \sim_w t'_i \; \forall \; i = 1, 2, \ldots s$. This is by replacing each $\perp$ with a value, that is not $\perp$, from the attribute's domain for each tuple. The active domain of an attribute A_i is defined as the set of all the distinct values that appear on A_i except the NULL. Note that active domain was called *visible domain* in papers [3, 4, 5, 2].

Definition 1 *The* active domain *of an attribute A_i ($VD^T(A_i)$) is the set of all distinct values except $\perp$ that are already used by tuples in T:*

$$VD^T(A_i) = \{t[A_i] : t \in T\} \setminus \{\perp\} \textit{ for } A_i \in R.$$

The upper index T can be removed from the notation to simplify it if it is known which instance is considered.

A strongly possible world is achieved by using the active domain values for each occurrence of NULLs. following is a formal definition of the concept of *strongly possible world* that was introduced in [3].

Definition 2 *A possible world T' of T is called a* strongly possible world (spWorld) *if $t'[A_i] \in VD^T(A_i)$ for all $t' \in T'$ and $A_i \in R$.*

spWorlds are used to introduce *strongly possible keys (spKeys)* and *strongly possible functional dependencies (spFDs)* as follows.

Definition 3 *A strongly possible functional dependency, in notation $X \rightarrow_{sp} Y$, holds in table T over schema R if there exists a strongly possible world T' of T such that $T' \models X \rightarrow Y$. That is, for any $t'_1, t'_2 \in T'$ $t'_1[X] = t'_2[X]$ implies $t'_1[Y] = t'_2[Y]$. The set of attributes X is a strongly possible key, in notation $sp\langle X\rangle$, if there exists a strongly possible world T' of T where X is a key in T'. That is, for any $t'_1, t'_2 \in T'$ $t'_1[X] = t'_2[X]$ implies $t'_1 = t'_2$.*

If $T' = \{t'_1, t'_2, \ldots, t'_p\}$ is an spWorld of $T = \{t_1, t_2, \ldots, t_p\}$, then we say that t'_i is an *sp-extension* of t_i if $t_i \sim_w t'_i$. For a subset $X \subseteq R$, we say that $t'_i[X]$ is an *sp-extension* of t_i on X if $t_i \sim_w t'_i$ such that for each $A \in X$: $t'_i[A] \in VD(A)$.

3 Related work

g_3 measure was introduced by Kivinen et. al. in [12] for total tables. And Giannella et al. [10] introduced a measure to approximate the satisfaction of functional dependencies in a table. They introduced the approximation measure IFD and provided a comparison with the two other approximation measures: g_3 (the minimum number of tuples that are required to be removed from the table to satisfy the dependency introduced in [12]) and τ (the probability of getting a true satisfaction guess of an FD introduced in [9]). Bounds were provided to find the differences and were applied to five datasets for analysis. It is shown that IFD and τ are more appropriate than g_3 when measuring the degree of knowledge for $X \rightarrow Y$ (applications like prediction, classification, and so on). On the other hand, g_3 measure is more

appropriate than IFD and τ when measuring the number of "violating" tuples in an FD.

In [15], Jef Wijsen studied some theoretical concepts in CQA (Consistent query answering), that is when a user sends queries to an inconsistent database regarding a set of constraints. They provided a database repairing by an acyclic binary relation $\leq_{db}$ on consistent database tables, such that $r_1 \leq_{db} r_2$ indicate that r_1 is at least as close to db as r_2. The minimum number of tuples that are required to be added and/or removed is one possible distance. Furthermore, the main concepts of database repairs and CQA were studied by Bertossi in [6]. J. Biskup and L. Wiese provide the preCQE algorithm that follows the formal properties of inference-proofness to find a solution for a given table in [7].

4 Approximation of strongly possible integrity constraints

Definition 4 *Attribute set K is an approximate strongly possible key of ratio a in table T, in notation $asp_a^- \langle K \rangle$, if there exists a subset S of the tuples T such that $T \setminus S$ satisfies $sp \langle K \rangle$, and $|S|/|T| \leq a$. The minimum a such that $asp_a^- \langle K \rangle$ holds is denoted by $g_3(K)$.*

The value of the measure $g_3(K)$ is between 0 and 1, where it is 0 when $sp \langle K \rangle$ holds in the table T (means it is not required to remove any tuples to satisfy the spKey). For this, we use the g_3 measure introduced in [12]. For example in Table 2, to satisfy the $sp\langle X \rangle$, we need to remove two out of four tuples as shown in Table 3, so that the g_3 is 0.5. The g_3 approximation measure for spKeys was introduced in [5]. This paper introduces the g_5 approximation measure for spKeys that is based on adding rather than removing tuples as in the following definition.

Definition 5 *Attribute set K is an add-approximate strongly possible key of ratio b in table T, in notation $asp_b^+ \langle K \rangle$, if there exists a set of tuples S such that the table TS satisfies $sp \langle K \rangle$, and $|S|/|T| \leq b$. The minimum b such that $asp_b^+ \langle K \rangle$ holds is denoted by $g_5(K)$.*

$g_5(K)$ is an approximation measure that is:

$$\frac{\text{minimum number of tuples to add}}{\text{total number of tuples}}$$

so that $sp \langle K \rangle$ holds. $g_5(K)$ measure has a value ranges between 0 and 1, where it equlas to 0 if $sp \langle K \rangle$ holds in T (means it is not required to remove any tuples to satisfy the spKey). For example in Table 2, to satisfy $sp\langle X \rangle$, it is enough to add one tuple as shown in Table 4, so that the g_5 is 0.25.

Definition 6 *For the attribute sets X and Y, $\sigma : X \rightarrow_{sp} Y$ is a remove-approximate strongly possible functional dependency of ratio a in a table T, in notation $T \models\approx_a^- X \rightarrow_{sp} Y$, if there exists a set of tuples S such that the table $T \setminus S \models X \rightarrow_{sp} Y$, and $|S|/|T| \leq a$. Then, $g_3(\sigma)$ is the smallest a such that $T \models\approx_a^- \sigma$ holds.*

Definition 7 *For the attribute sets X and Y, $\sigma : X \rightarrow_{sp} Y$ is an add-approximate strongly possible functional dependency of ratio b in a table T, in notation $T \models\approx_b^+ X \rightarrow_{sp} Y$, if there exists a set of tuples S such that the table $T \cup S \models X \rightarrow_{sp} Y$, and $|S|/|T| \leq b$. Then, $g_5(\sigma)$ is the smallest b such that $T \models\approx_b^+ \sigma$ holds.*

Let $U \subseteq T$ represent the set of the tuples that are required to be removed to have the spKey satisfied in T, in other words, $|U|$ tuples need to be removed. On other hand, adding only one tuple with new values may cause te satisfaction of the spKey in some tuples in U using the added new values for their `NULL`s. That means adding a number of tuples fewer than the those to remove can satisfy an spKey in the same table. For example, we either remove two tuples or add one to satisfy $sp \langle X \rangle$ in Table 2.

Table 2: Incomplete instance

X	
A_1	A_2
$\perp$	1
2	$\perp$
2	$\perp$
2	2

Table 3: Resulting table for ($asp_a^- \langle X \rangle$)

X	
A_1	A_2
$\perp$	1
2	2

4.1 Relation between g_3 and g_5 measures

Results together with their proofs of this subsection were reported in the conference volume [1], so the proofs are not

Table 4: The table after adding ($asp_b^+ \langle X \rangle$)

X	
A_1	A_2
$\perp$	1
2	$\perp$
2	$\perp$
2	2
3	3

included here, except for Theorem 1, which is shown for the sake of interested reader. The following Proposition is used to prove Proposition 2.

Proposition 1 *Let T be an instance over schema R and let $K \subseteq R$. If the K-total part of the table T satisfies the key $sp\langle K\rangle$, then there exists a minimum set of tuples U to be removed that are all non-K-total so that $T \setminus U$ satisfies $sp\langle K\rangle$.*

Proposition 2 *For any $K \subseteq R$ with $|K| \geq 2$, we have $g_3(K) \geq g_5(K)$.*

Apart form the previous inequality, the two measures are totally independent for spKeys.

Theorem 1 *Let $0 \leq \frac{p}{q} < 1$ be a rational number. Then there exist tables over schema $\{A_1, A_2\}$ with arbitrarily large number of rows, such that $g_3(\{A_1, A_2\}) - g_5(\{A_1, A_2\}) = \frac{p}{q}$.*

Proof: Table T is defined as follows.

$$T = \begin{array}{cc} & \left\{\begin{array}{cc} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & b \end{array}\right. \\ b & \\ x & \left\{\begin{array}{cc} \bot & \bot \\ \bot & \bot \\ \vdots & \vdots \\ \bot & \bot \end{array}\right. \end{array} \tag{1}$$

Clearly, $g_3(K) = \frac{x}{x+b}$. Let us assume that y tuples are needed to be added. The maximum number of active domain combinations is $(y+1)(y+b)$ obtained by adding tuples $(2, b+1), (3, b+2), \ldots, (y+1, y+b)$. This is enough to replace all tuples with NULLs if

$$(y+1)(y+b) \geq x + y + b. \tag{2}$$

On the other hand, $y-1$ added tuples are not enough, so

$$y(y-1+b) < x + y - 1 + b. \tag{3}$$

Since the total number of active domain combinations must be less than the tuples in the extended table. We have $\frac{p}{q} = g_3(K) - g_5(K) = \frac{x-y}{x+b}$ that is for some positive integer c we must have $cp = x - y$ and $cq = x + b$ if $gcd(p, q) = 1$. This can be rewritten as

$$\begin{array}{c} y = x - cp\,; \quad y + b = c(q-p) \\ b = cq - x\,; \quad x + y + b = y + cq \end{array} \tag{4}$$

Using (4) we obtain that (2) is equivalent with

$$y \geq \frac{cp}{c(q-p)-1}. \tag{5}$$

If c is large enough then $\lceil \frac{cp}{c(q-p)-1} \rceil = \lceil \frac{p}{q-p} \rceil$ so if $y = \lceil \frac{p}{q-p} \rceil$ is chosen then (5) and consequently (2) holds. On the other hand, (3) is equivalent to

$$y < \frac{cq-1}{c(q-p)-2}. \tag{6}$$

The right hand side of (6) tends to $\frac{q}{q-p}$ as c tends to infinity. Thus, for large enough c we have $\lfloor \frac{cq-1}{c(q-p)-2} \rfloor = \lfloor \frac{q}{q-p} \rfloor$. Thus, if

$$y = \lceil \frac{p}{q-p} \rceil \leq \lfloor \frac{q}{q-p} \rfloor \tag{7}$$

and $\frac{q}{q-p}$ is not an integer, then both (2) and (3) are satisfied for large enough c. Observe that $\frac{p}{q-p} + 1 = \frac{q}{q-p}$, thus (7) always holds. Also, if $\frac{q}{q-p}$ is indeed an integer, then we have strict inequality in (7) that implies (6) and consequently (3).

Unfortunately, the analogue of Proposition 1 is not true for spFDs, so the proof of the following theorem is quiet involved.

Theorem 2 *Let T be a table over schema R, $\sigma : X \rightarrow_{sp} Y$ for some $X, Y \subseteq R$. Then $g_3(\sigma) \geq g_5(\sigma)$.*

Theorem 3 can be proven by a construction similar to the proof of Theorem 1.

Theorem 3 *For any rational number $0 \leq \frac{p}{q} < 1$ there exists tables with an arbitrarily large number of rows and bounded number of columns that satisfy $g_3(\sigma) - g_5(\sigma) = \frac{p}{q}$ for $\sigma\colon X \rightarrow_{sp} Y$.*

4.2 Complexity problems

Definition 8 *The* SPKey problem *is the following.*
Input Table T over schema R and $K \subseteq R$.
Question Is it true that $T \models sp\langle K\rangle$?
The SPKeySystem problem *is the following.*
Input Table T over schema R and $\mathcal{K} \subseteq 2^R$.
Question Is it true that $T \models sp\langle \mathcal{K}\rangle$?
The SPFD problem *is the following.*
Input Table T over schema R and $X, Y \subseteq R$.
Question Is it true that $T \models X \rightarrow_{sp} Y$?

The following was shown in [4].

Theorem 4 *SPKey∈P, SPkeySystem and SPFD are NP-complete*

However, the approximation measures raise new, interesting algorithmic questions.

Definition 9 *The* SpKey-g3 *problem is the following.*
Input Table T over schema R, $K \subseteq R$ and $0 \leq q < 1$.
Question Is it true that $g_3(K) \leq q$ in table T?
The SpKey-g5 *problem is the following.*
Input Table T over schema R, $K \subseteq R$ and $0 \leq q < 1$.
Question Is it true that $g_5(K) \leq q$ in table T?

Proposition 3 *The decision problem SpKey-g5 is in P.*

Proof: Let us assume that tuples $s_i\colon i = 1, 2, \ldots, p$ over schema R are such that $T \cup \{s_1, s_2, \ldots s_p\}$ is optimal, so $g_5(K) = \frac{p}{m}$. Then clearly we may replace s_i by $s_i' = (z_i, z_i, \ldots, z_i)$ for all $i = 1, 2, \ldots, p$ where z_i's are pairwise distinct new values not appearing in the (extended) table $T \cup \{s_1, s_2, \ldots s_p\}$ so that $T \cup \{s_1', s_2', \ldots s_p'\} \models sp\langle K\rangle$.

Thus, if $g_5(K) \leq q$ is needed to be checked for a table T of m tuples, one may add $\lfloor q \cdot m \rfloor$ completely new tuples to obtain table T' and check whether $T' \models sp\langle K \rangle$ in polynomial time by Theorem 4.

Theorem 5 *Decision problem SpKey-g3 is in P.*

Proof: Let R be a relational schema and $K \subseteq R$. Furthermore, let T be an instance table over R that has some NULLs. Consider $T' = \{t' \in \Pi_{A \in K} VD^T(A) \colon \exists t \in T$ such that $t[K] \sim_w t'[K]\}$ and T' is total. Furthermore let the the bipartite graph $G = (T, T'; E)$ be the *K-extension graph of T* such that $\{t, t'\} \in E \iff t[K] \sim_w t'[K]$. So, finding a matching (if exists) of the graph G that covers T provides the tuples to be replaced in T to check if K is an spKey.

It was shown in [5] that the g_3 approximation measure for strongly possible keys satisfies

$$g_3(K) = \frac{|T| - \nu(G)}{|T|}.$$

where $\nu(G)$ denotes the maximum matching size in the K-extension graph G. However, the size of G is usually exponential function of the size of the input of the decision problem SpKey-g3, as T' is usually exponentially large.

In order to make our algorithm run in polynomial time we only generate part of T'. Let $T = \{t_1, t_2 \ldots t_m\}$ and $\ell(t_i) = |\{t^\star \in \Pi_{A \in K} VD^T(A) \colon t^\star \sim_w t_i[K]\}|$. Note that $\ell(t_i) = \prod_{A \colon t_i[A] = \perp} |VD^T(A)|$, hence these values can be calculated by scanning T once and using appropriate search tree data structures to hold values of active domains of each attribute. Sort tuples of T in non-decreasing $\ell(t_i)$ order, i.e. assume that $\ell(t_1) \leq \ell(t_2) \leq \ldots \leq \ell(t_m)$. Let $j = \max\{i \colon \ell(t_i) < i\}$ and $T_j = \{t_1, t_2, \ldots t_j\}$, furthermore $T_j^\star = \{t^\star : \exists t \in T_j : t^\star \sim_w t[K]\} \subseteq \Pi_{A \in K} VD^T(A)$. Note that $|T_j^\star| \leq \frac{1}{2} j(j-1)$. If $\forall i = 1, 2, \ldots, m \colon \ell(t_i) \geq i$, then define $j = 0$ and $T_j^\star = \emptyset$. Let $G^\star = (T_j, T_j^\star; E^\star)$ be the induced subgraph of G on the vertex set $T_j \cup T_j^\star$. Note that $|T_j^\star| \leq \frac{1}{2} j(j-1)$.
Claim $\nu(G) = \nu(G^\star) + |T \setminus T_j|$.
Proof of Claim: The inequality $\nu(G) \leq \nu(G^\star) + |T \setminus T_j|$ is straightforward. On the other hand, a matching of size $\nu(G^\star)$ in $G^\star$ can greedily be extended to the vertices in $|T \setminus T_j|$, as $t_i \in T \setminus T_j$ has at least i neighbours (which can be generated in polynomial time).

Thus it is enough to determine $\nu(G^\star)$ in order to calculate $g_3(K)$, and that can be done in polynomial time using Augmenting Path method [14].

Note that the proof above shows that the *exact value* of $g_3(K)$ can be determined in polynomial time. This gives the following corollary.

Definition 10 *The decision problem* SpKey-g3-equal-g5 *is defined as Input Table T over schema R, $K \subseteq R$.*
Question Is $g_3(K) = g_5(K)$?

Corollary 1 *The decision problem SpKey-g3-equal-g5 is in P.*

Example Let $R = \{A_1, A_2, A_3\}$, $K_1 = \{A_1, A_2\}, K_2 = \{A_2, A_3\}$.

$$T = \begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline t_1 & 1 & \perp & 1 \\ t_2 & 1 & 2 & 2 \\ t_3 & 2 & 1 & 1 \\ t_4 & 2 & 1 & 1 \end{array}$$

$T \setminus \{t_4\} \models sp\langle K_1 \rangle$ and $T \setminus \{t_4\} \models sp\langle K_2 \rangle$, but the sp-Worlds are different. In particular, this implies that for $\mathcal{K} = \{K_1, K_2\}$ we have $g_3(\mathcal{K}) > \max\{g_3(K) \colon K \in \mathcal{K}\}$ On the other hand, trivially $g_3(\mathcal{K}) \geq \max\{g_3(K) \colon K \in \mathcal{K}\}$ holds. This motivates the following definition.

Definition 11 *The problem* Max-g3 *defined as*
Input Table T over schema R, $\mathcal{K} \subseteq 2^R$.
Question Is $g_3(\mathcal{K}) = \max\{g_3(K) \colon K \in \mathcal{K}\}$?

Theorem 6 *Let Table T over schema R and $\mathcal{K} \subseteq 2^R$. The decision problem Max-g3 is NP-complete.*

Proof: The problem is in NP, a witness consists of a set of tuples U to be removed, an index $j \colon \frac{|U|}{|T|} = g_3(K_j)$, also an spWorld T' of $T \setminus U$ such that each K_i is a key in T'. Verifying the witness can be done in three steps.

1. $g_3(K_j) \not\leq \frac{|U|-1}{|T|}$ is checked in polynomial time using Theorem 5.
2. For all $i \neq j$ check that $g_3(K_i) \leq \frac{|U|}{|T|}$ using again Theorem 5.
3. Using standard database algorithms check that $\forall i \colon K_i$ is a key in T'.

On the other hand, the SPKeySystem problem can be Karp-reduced to the present question as follows. First check for each $K_i \in \mathcal{K}$ separately whether $sp\langle K_i \rangle$ holds, this can be done in polynomial time. If $\forall i \colon T \models sp\langle K_i \rangle$ then give $\mathcal{K}$ and T as input for Max-g3. It will answer Yes iff $T \models sp\langle \mathcal{K} \rangle$. However, if $\exists i \colon T \not\models sp\langle K_i \rangle$, then give the example above as input for Max-g3. Clearly both problems have No answer.

According to Theorem 4, it is NP-complete to decide whether a given SpFD holds in a table. Here we show that approximations are also hard.

Definition 12 *The* SPFD-g3 (SPFD-g5) *problems are defined as follows.*
Input A table T over schema R, $X, Y \subseteq R$, and positive rational number q.
Question Is $g_3(X \to_{sp} Y) \leq q$? ($g_5(X \to_{sp} Y) \leq q$?)

Theorem 7 *Both decision problems SPFD-g3 and SPFD-g5 are NP-complete.*

Proof: To show that SPFD-g3∈NP one may take a witness consisting of a subset $U \subset T$, an spWorld $T^\star$ of $T \setminus U$ such that $T^\star \models X \to Y$ and $|U|/|T| \leq q$. The validity

of the witness can easily be checked in polynomial time. Similarly, to show that SPFD-g5∈NP one may take a set of tuples S over R and an spWorld $T^\star$ of$T \cup S$ such that $T^\star \models X \to Y$ and $|S|/|T| \leq q$.

On the other hand, if $|T| = m$ and $q < 1/m$, then both SPFD-g3 and SPFD-g5 are equivalent with the original SPFD problem, since the smallest non-zero approximation measure is obtained if one tuple is needed to be deleted or added. According to Theorem 4, SPFD problem is NP-complete, thus so are SPFD-g3 and SPFD-g5.

Acknowledgement

The second author's research was partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–132696 and SNN–135643.

References

[1] Munqath Al-Atar and Attila Sali. Approximate keys and functional dependencies in incomplete databases with limited domains. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 147–167. Springer, 2022. doi: 10.1007/978-3-031-11321-5_9.

[2] Munqath Al-Atar and Attila Sali. Strongly possible functional dependencies for sql. *Acta Cybernetica*, 2022. doi: 10.1109/ACCESS.2022.3145678.

[3] Munqath Alattar and Attila Sali. Keys in relational databases with nulls and bounded domains. In *European Conference on Advances in Databases and Information Systems*, pages 33–50. Springer, 2019. doi: 10.1016/j.cose.2019.04.002.

[4] Munqath Alattar and Attila Sali. Functional dependencies in incomplete databases with limited domains. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 1–21. Springer, 2020. doi: 10.1016/j.is.2020.101522.

[5] Munqath Alattar and Attila Sali. Strongly possible keys for sql. *Journal on Data Semantics*, 9(2):85–99, 2020. doi: 10.1109/ACCESS.2020.3012345.

[6] Leopoldo Bertossi. Database repairs and consistent query answering: Origins and further developments. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 48–58, 2019. doi: 10.2200/S00974ED1V01Y201906DTM053.

[7] Joachim Biskup and Lena Wiese. A sound and complete model-generation procedure for consistent and confidentiality-preserving databases. *Theoretical Computer Science*, 412(31):4044–4072, 2011. doi: 10.1016/j.jcss.2010.06.004.

[8] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(5):692–709, 2007. doi: 10.1007/s10115-006-0048-7.

[9] Leo A Goodman and William H Kruskal. Measures of association for cross classifications. *Measures of association for cross classifications*, pages 2–34, 1979. doi: 10.1080/01621459.1979.10481657.

[10] Chris Giannella and Edward Robertson. On approximation measures for functional dependencies. *Information Systems*, 29(6):483–507, 2004. doi: 10.1016/j.jcss.2004.03.001.

[11] Henning Köhler, Uwe Leck, Sebastian Link, and Xiaofang Zhou. Possible and certain keys for sql. *The VLDB Journal*, 25(4):571–596, 2016. doi: 10.1145/2882903.2915200.

[12] Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theoretical Computer Science*, 149(1):129–149, 1995. doi: 10.14778/2153977.2154004.

[13] Witold Lipski Jr. On databases with incomplete information. *Journal of the ACM (JACM)*, 28(1):41–70, 1981. doi: 10.1145/322234.322239.

[14] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[15] Jef Wijsen. Foundations of query answering on inconsistent databases. *ACM SIGMOD Record*, 48(3):6–16, 2019. doi: 10.2200/S00999ED1V01Y201910DTM055.