Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Computer Science and Information Theory

Key and Functional Dependency Constraints for Incomplete Databases with Limited Domains

Ph.D. Dissertation Booklet

Munqath Hamid Al-Atar

Scientific Supervisor:
Dr. Sali Attila

February 26, 2021

Introduction

A relation is a set is an unordered collection of non-duplicated rows. While in an SQL table, the same row can appear more than once. In this dissertation, we use the bag semantics by taking list of tuples that allows several occurrences of the same tuple.

Keys have always been fundamental for database management, in particular for understanding the structure and semantics of data. A key is a set of attributes whose values will uniquely determine a tuple in the relation. It was Codd who formulated the principle of a key uniqueness and totality, that for a key K of any relation schema R, any relation with nulls over R, K must be null-free [4, 3]. In a given relation R, two attributes sets X and Y are said that X determines Y as a functional dependency denoted as $X \rightarrow Y$ if and only if, each value of X is associated with at most one value of Y for all the relation tuples.

Data used to represent practical information may not be complete and accurate. Although some methods of data analysis may overcome the missing value problem, many others require complete databases. Missing data may cause some anomalous results of the different database operations, such as selection, update and deletion. Data incompleteness issue complicates data analysis for the analysts. It may cause loss of data efficiency and effectiveness [2]. Serious problems can also result from missing data as most statistical methods automatically remove cases with missing data. Then, in the end, there would not be enough data to perform the analysis. Furthermore, when analyzing population-related data, if a specific class of the target population did not respond to the data collection questionnaire, they become underrepresented in the data. Hence, the measuring process is not going as intended. Missed data values issue is an essential challenge in the satisfaction of key and functional dependency constraints. Data imputation, or the filling in of missing values, for partially missing data can be used to complete such incompleteness.

In this dissertation, we considered only the existing attribute values in the table when the attribute's domain is not known, to provide a more meaningful and semantically acceptable possible (strongly possible) world. The main motivation for the study of strongly possible keys and functional dependencies lies in the identification of tuples of an incomplete dataset by filling up occurrences of nulls using the already present data only if it is possible. My theses are grouped around two main parts.

The first part focuses on handling the keys in incomplete databases when the domains of the attributes are limited to a specific set of values. For this purpose, a set of properties and conditions for keys in incomplete databases was introduced. An algorithm to discover whether a given incomplete data table satisfies a specific key was developed and the algorithmic solutions complexity was provided. For the cases that may not exist any strongly possible keys, an approximation approach was provided to measure how close a key can hold when it is violated in a data table with missing data values.

The second part is related to handle functional dependencies in incomplete databases with limited domains. We provided an algorithmic way to discover the satisfaction of an spFD in a given incomplete data table. A set of properties and conditions were provided. We introduced a set of axiom rules for FD over incomplete databases with limited domain for the axiomatization of these FD's. In addition to that, we analyzed the interaction between keys and FD's in such databases. And studied the special case when the FD's are restricted to singular attributes in databases with missing data values.

Theses

This dissertation mainly propose a novel solution on handling missing data values in cases where the attribute's domain may not be known and limited in their nature. The research solution is applied to examine the keys and FD's satisfaction and conclude rules, properties, theorems, and algorithms for that.

1 Keys in Incomplete Data With Limited Unknown Domain (sp-Keys)

1.1 Thesis 1: Strongly Possible Keys For SQL ([1 4])

A new version of keys for incomplete databases (spKeys) has been introduced. This key uses the attributes' *visible domain* (all the values already shown for each attribute) to overcome the missing values. A *strongly possible world* (spWorld) T' of incomplete table T is a complete table that obtained by replacing any null in T with a value from the corresponding attribute's visible domain. A set K of attributes is an spKey $sp\langle K\rangle$ in an incomplete table T if there exist an spWorld T' of T such that K is a key in T'. The results show that the concept of strongly possible keys lies in between the two concepts of possible and certain keys introduced in [6]. The properties of spKeys show that a single attribute with a null value cannot be a strongly possible key. Furthermore, two tuples shouldn't be strongly similar in the spKey attribute set (two tuples are *strongly similar* if they both have the same non-null value under every attribute, and they are *weakly similar* if both have the same value or one of them (or both) has null.

The characterization of the implication problem shows that for a set of strongly possible key constraints Σ and as a single spKey θ over a relation schema R, $\Sigma \models \theta$, if for every instance T over R satisfying every spKey in Σ, we have that T satisfies θ. The results show that systems of strongly possible key constraints enjoy Armstrong instances provided they satisfy a natural necessary condition.

1.2 Thesis 2: Determining Strongly Possible Keys ([1 6 7]

An spKey holds if a matching covering the table T exists in the bipartite graph $G = (T, T'; E)$, where T' is that set of all the possible combinations of the visible domains of T, so Hall's condition is naturally applied. We introduced an algorithm for validating if there is a strongly possible world that satisfies the key. The running time of the algorithm is $O(|K| \cdot |T| \log |T| + |T|^5)$, so it runs in a polynomial time on the size of the input as the number of tuples in the table. It was applied to real-world datasets to determine all two element strongly possible key sets.

The existence of a system of strongly possible keys is equivalent to the existence of a given sized common independent set of three or more matroids. This matroid intersection problem is NP-complete, however, this reduction is not one-to-one, so it does not prove, just hint the NP-completeness of spKey problem. We could prove the spKey problem is NP-complete in general by a Karp-reduction of 3SAT to our problem. But, if in the system of strongly possible keys, every key is disjoint from all other keys, then it can be decided in polynomial time . In case of a single strongly possible key constraint, it requires computing the largest common independent set of two matroids, which can be solved in polynomial time [7]. However, we can solve that case by reducing to the somewhat simpler problem of matchings in bipartite graphs.

1.3 Thesis 3: spKeys Approximation and Analytical Compression ([1 5])

We have introduced spKeys approximation to measure how close a strongly possible key holds in incomplete data tables using g_3 measure. g_3 based on the idea that the degree to which the spkey $sp\langle K \rangle$ is approximate is determined by the minimum number of tuples needed to be removed from the table so that K becomes a strongly possible key. We derived the measure g_4 from g_3 that consider the effects of the set of tuples that doesn't need to be removed on the measure results. An analytical comparison was applied on several tables represent different cases to compare the approximation measures of g_3 and g_4. The analytical comparison gave bounds of the two measures such that either $g_3(K) = g_4(K)$ or $1 < g_3(K)/g_4(K) < 2$. Furthermore, for any rational number $1 < \frac{p}{q} < 2$, there exist tables of arbitrarily large number of tuples with $g_3(K)/g_4(K) = \frac{p}{q}$.

2 FDs in incomplete data with limited Unknown domain (spFD)

2.1 Thesis 4: Strongly Possible Functional Dependencies For SQL ([5])

An important application of functional dependencies (spFD's) is the lossless decomposition of database tables to eliminate redundancy and the possibilities of inconsistent updates. Also, functional dependencies are important in maintaining data quality, enforce data consistency, and to guide repairs over a database [1]. An spFD $X \rightarrow_{sp} Y$ holds in an incomplete table T if there exist an spWorld T' of T such that $X \rightarrow Y$ holds in T'. We did a brief comparison of the different types of functional dependencies for incomplete databases with our proposed spFD.

A graph-theoretical characterization was introduced to determine when a table satisfies a given spFD. The characterization uses the weak similarity graphs (where the vertices represent the tuples and there is an edge between two vertices if their tuples are weakly similar). We employed the list coloring approach to characterize when $T \models X \rightarrow_{sp} Y$ holds using weak similarity graphs. The characterization shows that the existence of proper coloring of the complement of the weak similarity graph of Y ($\overline{G_Y}$) using the lists determined by the strongly possible extensions on X (strongly possible extension of a tuple t is a total tuple t' that obtain by replacing any null in t with a value from the corresponding visible domain) is a necessary and sufficient condition for $T \models X \rightarrow_{sp} Y$ to hold. We proved that it is always enough to generate at most $\Delta(\overline{G_Y}) + 1$ X-extensions for each tuple t. We showed that the spFD satisfaction problem is NP-complete. However, this problem can be solved in polynomial time when $\overline{G_Y}$ is a complete graph.

For $T \models X \rightarrow_{sp} Y$, fixing the set X of the spFD, makes the right-hand sides form a down-set. And for a fixed set Y, the left-hand side forms an up-set. And this charactirazes the spFD's with fixed left-hand side or right-hand side.

2.2 Thesis 5: spFD's Axiomatisation ([2])

We collected axioms and rules that are sound for strongly possible functional dependencies. We studied and analyzed the axioms of weak/strong [8] and possible/certain [9, 5] FD's in the context of strongly possible functional dependencies. The main goal is to find a good basis for the axiomatization of strongly possible functional dependencies. However, the inherent finiteness of the domains gives such restrictions that traditional completeness proofs are unusable. For example, we show that union rule is only satisfied if there is a single strongly possible world satisfying both dependencies in the premise. This makes it hard to give a good definition of strongly possible closure of an attribute set. One of the main further research directions is a resolution of this problem, that is a reasonable definition of sp-closure. Reflexivity, Augmentation, and Decomposition axioms are sound for spFD's. While Union, Transitivity, Pseudo-transitivity, and Composition axioms are not sound.

We provided several possible weakenings and restrictions that keep soundness for those that are not. We deduced that if a rule contains only one spFD in its premise, then it remains sound. Because more than one spFD's in the premise cause that spFD's may not hold in the same spWorld by the limitations of visible domains. For a complete axiomatization, this problem must be handled. In particular, the fact that composition does not hold in general makes usual proof methods of completeness virtually unusable. Our experience shows that if no null-free subschema is defined, then spFD's are "rather independent" of each other, that is they satisfy only inference rules that have a single spFD as a premise, except for spFD's between one element attribute sets.

We believe that for attribute sets of sizes larger than one, *Reflexivity, Augmentation, Decomposition, Disjoint composition* form a complete system of axiomatization. The proof of this may be based on that for a collection Σ of spFD's and singleton spFD $X \to_{sp} Y$ the theorem could be used by constructing a table instance that satisfies every spFD in Σ but $\overline{G_Y}$ cannot be list-colored using the strongly possible extensions on X. The interaction between sp-keys and sp-FD/c-FDs was introduced and the weakening and transitivity interaction rules were obtained.

2.3 Thesis 6: spFDs for singular attributes ([3])

Substituting a value from the visible domain in place of a $\perp$ produces a duplication in that attribute. Because for a singular attribute, the visible domain represents all the possible replacements for any $\perp$ occurrence. spFD's between singular attributes have special properties. For example, we showed a bidirectional property for singular attributes spFD. There is a natural correspondence between directed graphs and single attribute spFD's. We gave a characterization of those directed graphs that may occur in this context.

List of Publications

[1] Alattar, Munqath, & Attila Sali. ”Strongly Possible Keys for SQL.” Journal on Data Semantics 9.2 (2020): 85-99.

[2] Alattar, M., & Sali, A. ”Toward an Axiomatization of Strongly Possible Functional Dependencies.” Vietnam Journal of Computer Science 8.1 (2021): 1-19.

[3] Alattar, M., & Sali, A. ”Strongly Possible Functional Dependencies for SQL.” Acta Cybernetica Journal: Accepted

[4] Alattar, M., & Sali, A. (2019, May). ”Strongly possible keys.” In Pannonian Conference on Advances in Information Technology (PCIT 2019) (p. 23).

[5] Alattar, M., & Sali, A. (2019, June). ”Strongly Possible Keys in Incomplete Databases with Limited Domains.” The Eleventh International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2019).

[6] Alattar, M., & Sali, A. (2019, September). ”Keys in relational databases with nulls and bounded domains.” In European Conference on Advances in Databases and Information Systems, Part of the Lecture Notes in Computer Science book series (LNCS, volume 12245), (pp. 33-50). Springer, Cham.

[7] Alattar, M., & Sali, A. (2020, February). ”Functional Dependencies in Incomplete Databases with Limited Domains.” In International Symposium on Foundations of Information and Knowledge Systems, Part of the Lecture Notes in Computer Science book series (LNCS, volume 12012), (pp. 1-21). Springer, Cham.

[8] Alattar, M., & Sali, A. (2019, October). ”Keys and Functional dependencies in Incomplete databases with Limited Domains.” 15th International Miklos Ivanyi PhD & DLA Symposium (p. 86)

[9] Alattar, M., & Sali, A. (2020, October). ”Multivalued dependencies in incomplete databases with limited domain: Properties and rules.” 16th International Miklos Ivanyi PhD & DLA Symposium.

References

[1] Laure Berti-Equille et al. “Discovery of genuine functional dependencies from relational data with missing values”. In: 2018.

[2] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. “A novel framework for imputation of missing values in databases”. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 37.5 (2007), pp. 692–709.

[3] Sven Hartmann, Uwe Leck, and Sebastian Link. “On Codd families of keys over incomplete relations”. In: *The Computer Journal* 54.7 (2011), pp. 1166–1180.

[4] Sven Hartmann, Uwe Leck, and Sebastian Link. “Strong keys and functional dependencies in partial database relations”. In: *ICDT*. Citeseer. 2010.

[5] Henning Köhler and Sebastian Link. “SQL schema design: Foundations, normal forms, and normalization”. In: *Information Systems* 76 (2018), pp. 88–113.

[6] Henning Köhler et al. “Possible and certain keys for SQL”. In: *The VLDB Journal* 25.4 (2016), pp. 571–596.

[7] Eugene L Lawler. “Matroid intersection algorithms”. In: *Mathematical programming* 9.1 (1975), pp. 31–56.

[8] Mark Levene and George Loizou. “Axiomatisation of functional dependencies in incomplete relations”. In: *Theoretical Computer Science* 206.1 (1998), pp. 283–300.

[9] Y Edmund Lien. “On the equivalence of database models”. In: *Journal of the ACM (JACM)* 29.2 (1982), pp. 333–362.