Problem Decomposition Method to Compute an Optimal Cover for a Set of Functional Dependencies

Vitalie COTELEA

Academy of Economic Studies of Moldova
61 Banulescu-Bodoni Street, Chisinau, Republic of Moldova
Email: vitalie.cotelea@gmail.com

The paper proposes a problem decomposition method for building optimal cover for a set of functional dependencies to decrease the solving time. At the beginning, the paper includes an overview of the covers of functional dependencies. There are considered definitions and properties of non redundant covers for sets of functional dependencies, reduced and canonical covers as well as equivalence classes of functional dependencies, minimum and optimal covers. Then, a theoretical tool for inference of functional dependencies is proposed, which possesses the uniqueness property. And finally, the set of attributes of the relational schema is divided into equivalence classes of attributes that will serve as the basis for building optimal cover for a set of functional dependencies.
Keywords: *Logical Database Design, Functional Dependencies, Optimal Cover, Problem Decomposition*

1 Introduction

The A set of functional dependencies may have different structures, and several properties of the database are dictated by structure and quality of the functional dependencies set. In addition, a set of functional dependencies can be modified, simplified, while retaining its qualitative aspect of the deduction system.

The notion of equivalent sets of functional dependencies is a central one in database design. This is because the design process starts from a given schema, modifying it to obtain a schema with desirable qualities, but equivalent with former in terms of integrity constraints both structural and behavioral.

These sets of functional dependencies usually are called covers. If the structure of dependencies set is simple, it is more efficient the checking of the database consistency and the possibility of application of integrity constraints is facilitated.

Further, the computing of optimal cover for a set of functional dependencies is dealt with.

2 Covers for functional dependencies

Definition 1. ([1], p.71) Two sets of functional dependencies F and G over scheme R are equivalent, written $F \equiv G$, if $F^+ = G^+$.

It is said that in case if $F \mid\!- G$, then F covers G. If sets F and G are equivalent, they are cover one for another.

If $F \equiv G$, that is if $F^+ = G^+$, then any dependence $X \to Y$ that is implied by F is implied by G. So to check if F and G are equivalent, take any dependence $X \to Y$ in F and check if $G \mid\!- X \to Y$. If some dependence $X \to Y$ does not belong to G^+, then $F^+ \neq G^+$. Then, analog, check if any dependence $V \to W$ in G is derived from F. If all dependencies are derived from these appropriate sets, the sets F and G are equivalent.

Consider by $|F|$ and $\|F\|$ the cardinality of F and number of attributes involved by F (including repeated), respectively. As the complexity of the algorithm to deduct a dependence from a given set of functional dependencies, that is a inference of dependence $X \to Y$ from the set F, is

$O(\|F\|)$, it is not hard to see that the algorithm for determining whether two sets of functional dependencies F are G are equivalent, will consume a polynomial time relative to the size of the input data, $O(\|G\|\cdot|F|+\|F\|\cdot|G|)$.

Example 1. The sets $F=\{AB\to C,$ $AC\to D$, $AD\to B$, $C\to B\}$ and $G=\{AD\to C,$ $AB\to D$, $C\to B\}$ are equivalent, but F is not equivalent to the set $G'=\{AB\to C,$ $AC\to D$, $AD\to B$, $AC\to B\}$.

Definition 2. ([2], p.295) A set F of functional dependencies is non redundant, if $\nexists G$, so that $G\subset F$ (that is, if there is no proper subset G of F) with $G\equiv F$. If such subset exists, then F is redundant. The set F is a non redundant cover for G, if F is cover for G and F is non redundant.

Example 2. Let $G=\{A\to BC$, $B\to C\}$. The set $F=\{A\to B$, $A\to C$, $B\to C\}$ is a cover for the set G, but it is a redundant cover, since $F^1=\{A\to B,\ \ B\to C\}$ is a cover for G and $F^1\subset F$.

There is an alternative definition (procedural) for notion of non redundant cover.

Definition 3. [3] The set F of functional dependencies is non redundant, if there is no functional dependency $X\to Y$ in F, such that $(F-\{X\to Y\})\models X\to Y$. Otherwise, F is redundant.

This definition forms the basis for the algorithm which computes a non redundant cover. It is worth to mention that the result from the application of the algorithm depends on the order of examining the functional dependencies.

It is not hard to see that a set of functional dependencies may have more than one non redundant cover. The result depends on the order in which functional dependencies are examined for removal. In [4], it is considered the representation of all non redundant covers of a set of functional dependencies, and in [5] an efficient algorithm for computing a non redundant cover is presented.

If F is a non redundant set of functional dependencies, then it can not be removed any functional dependency from F, without affecting the equivalence of the obtained set with the previous one.

In contrast, functional dependencies in F can be reduced in size by removing some attributes from them.

Definition 4. ([1], p.74) Let F be a set of functional dependencies over scheme R and let $X\to Y\in F$. Attribute A in R is extraneous in dependency $X\to Y$ with respect to F, if

1. $A\in X$, $F-\{X\to Y\}\cup\{(X-\{A\})\to Y\}\equiv F$

or

2. $A\in Y$, $F-\{X\to Y\}\cup\{X\to(Y-\{A\})\}\equiv F$.

In other words, the attribute A is extraneous in dependency $X\to Y$, if it can be removed from the left or right side of dependency, without changing the closure of F. The elimination process of extraneous attributes is called, respectively, left reduction or right reduction of dependencies.

Definition 5. ([1], p.74) A set F of functional dependencies is left-reduced (right-reduced), if every functional dependency in F contains no extraneous attributes in the left (right) side. If a set of functional dependencies is left-reduced and right-reduced then it is reduced.

Theorem 1. If a set of functional dependencies is reduced, then it is non redundant.

Proof. The statement is true, because if it is assumed that the set of functional dependencies is not non redundant, then from the set may be removed at least one functional dependency and therefore all the attributes of this dependence are extraneous, which contradicts the claim that the set is reduced.

It's obvious that if a dependency is

redundant, then all its attributes are extraneous. To avoid dependencies of the form $X \to \varnothing$, it is assumed that the set which must be reduced is non redundant.

It seems that reduced cover can be calculated by finding and removing at random extraneous attributes. But, considering left and right sides of the dependencies in a different order, it can get different results. When the right sides are examined first, after considering the left, it may appear redundant attributes in right sides. So, if the set of dependencies is not non redundant at algorithms entry, the eliminating of extraneous attributes must begin with the left sides.

Definition 6. ([1], p.77) A set of functional dependencies F is canonical, if F is non redundant, left-reduced and every functional dependency in F is of the form $X \to A$.

Example 3. The set $F = \{A \to B$, $A \to C$, $B \to D\}$ is a canonical cover for $G = \{A \to BC$, $B \to D\}$.

Theorem 2. A canonical set of functional dependencies is a reduced set of functional dependencies.

Proof. The validity of this statement follows directly from Definition 6. Since a canonical set of functional dependencies is non redundant and every functional dependency has a single attribute on the right side, it is right-reduced. Since it is also left reduced, it is reduced.

The reverse statement is not true. This is shown in Example 3, where the set G is reduced, but it is not canonical. However, the relationship between reduced and canonical covers can be characterized by the following theorem.

Theorem 3. ([1], p.77) Let F be a reduced cover. If the set G of functional dependencies is formed by splitting each dependency $X \to A_1 \dots A_n$ in F into $X \to A_1, \dots, X \to A_n$, then G is a canonical cover of F. Conversely, if G is a canonical cover, it is reduced. If the set F is formed by aggregating all dependencies in G with equal left sides into a single functional dependency, then F is also a reduced cover.

Definition 7. ([1], p.78) Let F be a set of functional dependencies over scheme R and let $X, Y \subseteq R$. Two sets of attributes X and Y are equivalent, written $X \leftrightarrow Y$, under the set F, if $F \models X \to Y$ and $F \models Y \to X$.

Definition 7 suggests that the set F can be partitioned into equivalence classes. That is, on F can be defined an equivalence relation: dependencies $X \to Y$ and $V \to W$ in F belong to a class of equivalence, if and only if $X \leftrightarrow V$ under the set F.

Definition 8. Let F be a set of functional dependencies over scheme R and let $X, Y \subseteq R$. It is defined as the set of equivalence classes of functional dependencies for the set X of attributes with respect to F, denoted $E_F(X)$, the set $E_F(X) = \{V \to W \mid \ V \to W \in F \,\&\, X \leftrightarrow V\}$.

So $E_F(X)$ is the set of functional dependencies in F with left sides equivalent to X with respect to F.

Let $\overline{E}_F$ be the set $\overline{E}_F = \{E_F(X) \mid \ X \subseteq R \,\&\, E_F(X) \neq \varnothing\}$. In other words, $\overline{E}_F$ is the set of all nonempty equivalence classes, in which the set F of functional dependencies is partitioned.

The next lemma shows the correlation between structures of two equivalent and non redundant sets of functional dependencies.

Lemma 1. ([1], p.78) Let F and G be equivalent, non redundant sets of functional dependencies over scheme R. Let $X \to Y$ be a functional dependency in F. There is a functional dependency $V \to W$ in G with $X \leftrightarrow V$ under F (hence under G).

Lemma above can be paraphrased as follows. In two non redundant covers F and G, for each dependency in F there is a dependency in G with equivalent left sides. Therefore, the equivalent non redundant sets of functional dependencies have the same number of equivalence classes.

Definition 9. ([1], p.79) A set F of functional dependencies is minimum if F has as few functional dependencies as any equivalent set G of functional dependencies, that is

$$\forall G \quad F \equiv G \quad \Rightarrow \quad |F| \leq |G|.$$

Theorem 4. Let F be a minimum set of functional dependencies. Then F is a non redundant set of functional dependencies.

Proof. The statement is true, because if it is assumed that the set F is not non redundant, then it can be removed from at least one functional dependency and therefore there will be a cover with fewer functional dependencies, which contradicts the assumption that it is minimum.

It is obvious that the reverse statement is not correct.

Consider by $PS_F(X)$ the set of left sides of the dependencies forming equivalence class $E_F(X)$, that is:

$$PS_F(X) = \{V \mid \quad V \to W \in E_F(X)\}.$$

Then there is:

Lemma 2. ([1], p.81) Let F be a non redundant set of functional dependencies. Pick X, a left side of some functional dependency in F and any Y equivalent to X (that is $X \leftrightarrow Y$ under F). There exists a set Z in $PS_F(X)$ such that $(F - E_F(X)) \models Y \to Z$.

Lemma 3. [6] Let F and G be equivalent, non redundant sets of functional dependencies over scheme R. Let X be a left side of some functional dependency in F and any Y such that $X \leftrightarrow Y$ under F. If $Y \to Z \in (F - E_F(X))^+$, then $Y \to Z \in (G - E_G(X))^+$.

Theorem 5. [6] A non redundant set F of functional dependencies is a minimum set, if and only if there are no distinct functional dependencies $X \to Y$ and $V \to W$ in any equivalence class $E_F(X)$ such that $X \to V \in (F - E_F(X))^+$.

Corollary 1. If F and G are equivalent, minimum sets of functional dependencies, then the corresponding equivalence classes contain the same number of functional dependencies.

Corollary 2. If F and G are equivalent and minimum sets of functional dependencies, then for each left side $X_j \in PS_F(X)$ there is a single left side V_k in $PS_G(X)$ such that $X_j \to V_k \in (F - E_F(X))^+$ and $V_k \to X_j \in (F - E_F(X))^+$.

Proposition 1. The existence of the bijection indicated in Corollary 2, allows substitution of some left sides of a minimum set of functional dependencies by the corresponding left sides of another minimum cover, which does not affect the equivalence of minimal sets. In addition, the new set of functional dependencies will continue to be minimal.

The above theorem states that if a non redundant set G has two dependencies $X \to Y$ and $V \to W$, such that $X \leftrightarrow V$ and $(G - E_G(X)) \models X \to V$, then G is not minimum set of functional dependencies. These two dependencies can be substituted with other functional dependency $V \to YW$. Consequently, it is obtained an equivalent set of functional dependencies with one dependency less.

The algorithm to minimize a set of functional dependencies is based on this process.

A set F of functional dependencies can be evaluated taking into account the number of attribute symbols (including repeated) involved by the functional dependencies in F. For example, the set $F=\{AB \to C,\ \ C \to B\}$ consists of five attribute symbols, that is $\|F\|=5$.

Definition 10. ([1], p.86) A set F of functional dependencies is optimal if there is no equivalent set G of functional dependencies with fewer attribute symbols than F, that is

$$\forall G \quad F \equiv G \quad \Rightarrow \quad \|F\| \leq \|G\|$$

Theorem 6. ([1], p.86) An optimal set of functional dependencies is reduced and minimum.

Example 4. The set $F=\{ABC \to E$, $BC \to D$, $D \to BC\}$ is not an optimal set of functional dependencies, because the set $G=\{AD \to E$, $BC \to D$, $D \to BC\}$ consists of fewer symbols than F and $F \equiv G$. It should be noted that the set G is optimal.

Unfortunately, there is not known any algorithm of polynomial complexity that would build an optimal cover for a given set of functional dependencies. This problem belongs to the class of NP-complete problems.

A size reduction technique for solving this problem is proposed below.

Definition 11. Let $X \to Y \in F$ be a functional dependency. The set X of attributes is a determinant for the set Y of attributes, if no proper subset X' of set X exists such that $X' \to Y \in F^+$.

3 An inference model of functional dependencies

To prove several assertions about functional dependency inference, a model called Maximal derivation is proposed. This model deducts in a linear mode the set of attributes functionally dependent (under a set of functional dependencies) for a given set of attributes.

It has the uniqueness property and it is very easy to use in demonstrating claims about functional dependencies structures. In general, this model is not something else than a sequence of sets of attributes, which are built iteratively, involving for their construction groups of functional dependencies with left sides included in the previous set.

Since the maximal derivation is a sequence of sets of attributes, there can be built its reduced version, called simply – derivation, which effectively applies in the inference of functional dependencies. Some properties of the proposed model and its inference ability have been proven. It is equivalent to applying the inference model of dependencies with Armstrong axioms.

In [7] is presented a model called *maximal derivation* (the name is taken from [8]). The construction concept is based on the algorithm which computes the closure of the set of attributes under the set of dependencies, as described in [5].

Definition 12. Let F be a set of functional dependencies over set R of attributes and let $X \subseteq R$. *Maximal derivation* of the set of attributes X under the set F of dependencies is a sequence of sets of attributes $< X_0, X_1, ..., X_n >$, so that:

(1). $X_0 = X$;

(2). $X_i = X_{i-1} \bigcup Z$, $i=\overline{1n}$, where $Z=\bigcup_j W_j$ for all dependencies $V_j \to W_j \in F$ which satisfies $V_j \subseteq X_{i-1}$ and $W_j \not\subset X_{i-1}$;

(3). Nothing else from R is a member of X_i.

Before we show that maximal derivation is a powerful derivation tool for functional dependencies, two of its properties are considered.

Lemma 4. [7] If $X \subseteq Y$ and sequences $< X_0, X_1, ..., X_n >$, $< Y_0, Y_1, ..., Y_m >$ are maximal derivations of the sets X and Y, respectively, under F, then for any X_i

exists a set Y_j such that $X_i \subseteq Y_j$ and $j \le i$.

This property tells us that if the set of attributes is larger, then the terms of maximal derivation converge faster and they are closer to the beginning of the maximal derivation.

Lemma 5. [7] If $< X_0, X_1, ..., X_n >$ is the maximal derivation of the set X under the set F of functional dependencies, then $X \to X_i \in F^+$, $i = \overline{0n}$.

The property represented by this lemma states that any term of maximal derivation is functionally determined by the set of attributes on which this derivation is built.

Based on these two properties the next theorem will be proven:

Theorem 7. [7] Let $< X_0, X_1, ..., X_n >$ be the maximal derivation of the set $\boldsymbol{X}$ under the set $\boldsymbol{F}$ of functional dependencies. Then $X \to Y \in F^+$ if and only if $Y \subseteq X_n$.

This theorem actually proves that applying the maximal derivation for the deduction of functional dependencies from a given set of dependencies is equivalent to applying Armstrong's axioms for the dependencies deduction process, because this theorem's proof is based only on the inference of these rules. But unlike other derivation instruments, the deduction using maximal derivation is unique, i.e. there are no two different maximal derivations for the deduction of a functional dependency from a given set of dependencies.

Due to the fact that Armstrong rules are sound and complete, the maximal derivation has the same properties. In addition, the derivation is a deterministic process and not a nondeterministic one as is the case of deduction using rules of inference.

Definition 13. [7] Let $X \to Y \in F^+$ and $< X_0, X_1, ..., X_n >$ be the maximal derivation of the set X under F. Let X_i be the first element which contains the set Y. Then the subsequence $< X_0, X_1, ..., X_i >$ is considered to be the *derivation* (not necessarily the maximal one) of the functional dependency $X \to Y$ under F.

From Theorem7 and Definition 13 follows

Corollary 3. [7] $X \to Y \in F^+$ then and only then when the derivation of $X \to Y$ under F exists.

Corollary 4. [7] If $X \to Y \in F^+$ and the dependency $V \to W \in F$ is used for computing the derivation of the $X \to Y$ under F, then $X \to V \in F^+$.

The correctness of this statement logically follows from the Lemma5 and the reflexivity and transitivity rules.

It is obvious that the last element, X_n, in maximal derivation is nothing else but X^+. And Theorem 7 says that $X \to Y$ follows logically from F, if $Y \subseteq X^+$. So, the maximal derivation serves as a theoretical model for algorithm to building the closure of a set X of attributes under a set F of functional dependencies.

The uniqueness of derivation is explained by the fact that every step of the algorithm to create the next term of maximal derivation involves all dependencies that satisfy condition (2), respectively.

4 Redundant and non redundant equivalence classes of attributes

In this section, we introduce the notion of contribution graph for a set of functional dependencies and condensed graph of the contribution graph. Also, it is presented that strongly connected components of a contribution graph divide the set of attributes of relational schema into equivalence classes of attributes and a strict partial order can be defined over the nodes of condensed graph.

Mapping of functional dependencies inference in contribution graph is examined and there are introduced concepts of redundant equivalence class and non redundant equivalence class of attributes.

Given a set F of functional dependencies on the set R of attributes, that are part of the relation scheme $Sch(R,F)$, a contribution graph is drawn, in order to represent F.

Definition 14. Contribution graph $G=(S,E)$ of set F is a graph that:

- for $\forall A \in R$ there exists in S a vertex labeled with attribute A;
- for $\forall X \to Y \in F$ and for $\forall A \in X$ and $\forall B \in Y$ there exists in E an edge $a=(A,B)$, that is directed from vertex A to vertex B.

Example 5. If $F=\{ABC \to E$, $BC \to D$, $D \to BC\}$ and $R=\{A,B,C,D,E\}$, then the contribution graph of set F of dependencies is presented in Figure 1.

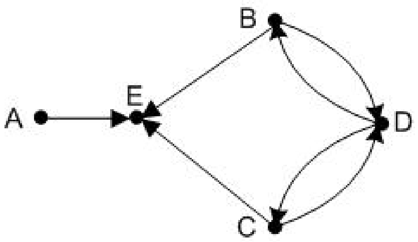

Fig. 1. A contribution graph for set F

Two vertices $A,B \in S$ are strongly connected, if and only if there exists in graph G a path from A to B and backwards, from B to A. It is obvious that the relation of strong connectivity is an equivalence relation. So, there is a partition of set of vertices S into pairwise disjoint subsets. That is, $S=\bigcup_{i=1}^{n} S_i$ and all vertices in S_i, $i=\overline{1,n}$, are strongly connected, and every two vertices from different subsets are not strongly connected.

In accordance with this partition, sub-graphs $G_i=(S_i,E_i)$, $i=\overline{1,n}$ are called strongly connected components [9] of the graph G, where E_i represents the set of edges that connect pairs of vertices in S_i.

Example 6. The set of vertices of the graph represented in Figure 1 are split into three equivalence classes $S_1=\{A\}$, $S_2=\{B,C,D\}$ and $S_3=\{E\}$.

The concept of the condensed graph of a contribution graph is introduced:

Definition 15. Let G^* be the condensed graph of the graph G. Set of vertices of graph G^* represents set $\{G_1,...,G_n\}$ of all strongly connected components of graph G and there is an edge from vertex G_i to vertex G_j of graph G^*, if there exists in G at least one edge that connects one vertex from component G_i to one vertex from component G_j.

Obviously the graph G^* is an acyclic one.

Example 7. The condensed graph of graph from Figure 1 has three vertices and two edges, as shown in Figure 2.

Fig. 2. Condensed graph of the graph from Figure 1

Over the set of vertices of graph G^* a strict partial order is defined. Vertex G_i precedes vertex G_j, if G_j is accessible from G_i. Now, the equivalence classes $S_1,...,S_n$ will be sorted based on the corresponding order graph's G^* vertices.

Lemma 6. If $X \to Y \in F^+$ and X is a determinant of set Y under F, then for every attribute $A \in (X-Y)$ there is an attribute $B \in Y$ so that in the contribution graph G there exists a path from vertex A to vertex B and for every attribute $B \in (Y-X)$ there exists in X an attribute A, from which the vertex B can be reached.

Proof. Let attribute $B \in (Y-X)$ and let the subset X' of set X be determinant for

B under F. Because $X' \to B \in F^+$, according to Definition 13, there is a derivation $H = < X'_0, X'_1, ..., X'_m >$ for dependency $X' \to B$ under F. Then, based on Corollary 4, there exists a sequence of dependencies $V_1 \to W_1, ..., V_q \to W_q$ in F, where $A \in V_1$, $B \in W_q$ and $W_i \cap V_{i-1} \neq \varnothing$, for $i = \overline{1, q-1}$. Contribution graph has a structure, such that for every dependency $V_j \to W_j$ in F, from each vertex labeled with an attribute in V_j an edge leaves to every vertex labeled with an attribute in W_j. So, there exists a path from every vertex $A \in X'$ to vertex B.

It must be mentioned that, if $\overline{X}$ is considered the union of all determinant of attributes in $Y - X$, then $\overline{X} \cup X \cap Y = X$. Indeed, if we suppose that the set $\overline{X} \cup X \cap Y$ is a proper subset of set X, this will contradict the supposition that X is a determinant for Y under F.

Corollary 5. If reduced dependency $V \to W$ is used non redundantly in building the derivation H for dependency $X \to Y$ under F, then in contribution graph G there exists a path from every vertex labeled with an attribute in V to every vertex labeled with an attribute in Y.

The following theorem shows a correlation between non redundant equivalence classes of attributes and the left and right sides of a left-reduced dependency.

Theorem 8. Let $X \to Y \in F^+$, where X is a determinant for Y under F and $X, Y \subseteq T_1 \cup ... \cup T_m$. For a T_j, where $j = \overline{1, m}$, the following takes place: if $Y \cap T_j \neq \varnothing$, then $X \cap T_j \neq \varnothing$.

Proof. The soundness of this statement is proven by contradiction: let $Y \cap T_j \neq \varnothing$, but $X \cap T_j = \varnothing$. Evidently that $X \subseteq T_1 \cup ... \cup T_{j-1} \cup T_{j+1} \cup ... \cup T_m$ and $X \to (Y \cap T_j) \in F^+$. Let X', where $X' \subseteq X$, is determinant for $Y \cap T_j$ under F. According to Lemma 6, on the contribution graph of the set F of dependencies, from every vertex labeled with an attribute in X' there exists a path to a vertex labeled with an attribute in $Y \cap T_j$. Thereby, $X' \subseteq T_1 \cup ... \cup T_{j-1}$. But, in this case, T_j is redundant. Therefore, $X' \cap T_j \neq \varnothing$. Then, more so $X \cap T_j \neq \varnothing$ takes place. A contradiction has been reached.

Below, there are shown a series of features related to the determinants and the sets of redundant and non redundant equivalence classes of attributes.

Theorem 9. If X is a determinant under F of set $S_1 \cup ... \cup S_j$, where $j = \overline{1, n}$, then $X \subseteq S_1 \cup ... \cup S_j$.

Proof. Let $X \not\subset S_1 \cup ... \cup S_j$. Then there exists an equivalence class S_t, where $t = \overline{j, n}$, such that $X \cap S_t \neq \varnothing$. By Lemma 6, in the contribution graph G, from every attribute $A \in X \cap S_t$ there is a path towards B, where $B \in S_1 \cup ... \cup S_j$. But this fact contradicts the supposition that sets $S_1, ..., S_j$ precede the set S_t.

Corollary 6. If X is a determinant of set $S_1 \cup ... \cup S_n$ under F, then $X \cap S_1 \neq \varnothing$.

Proof. Indeed, for every attribute B in S_1 or $B \in X$, or, according to Lemma 6, there is in X an attribute A from which vertex B is accessible in contribution graph G. But then A is also a member of equivalence class S_1.

Definition 16. Equivalence class S_j is called non redundant, if and only if for every attribute A in S_j, the expression $(\bigcup_{i=1}^{n} S_i - S_j) \to A \notin F^+$ holds.

Considering Lemma 6, it can be concluded that set S_j is non redundant, if and only if for every attribute A in S_j, the expression $(\bigcup_{i=1}^{j-1} S_i) \to A \notin F^+$ holds.

From the ordered sequence of sets $S_1,...,S_n$ a sequence of ordered non redundant sets can be built $T_1,...,T_n$, where $T_1 = S_1$ and $T_j = S_j - (\bigcup_{i=1}^{j-1} T_i)_F^+$ for $j = \overline{2,n}$. As a result of this process, some sets T_j can become empty. These empty sets can be excluded from the sequence and a sequence of nonempty sets $T_1,...,T_m$ will be obtained, keeping the precedence of prior sets.

Proposition 2. $T_1 = S_1$.

Proposition 3. $(T_1 \cup ... \cup T_m) \to (S_1 \cup ... \cup S_n) \in F^+$.

Example 8. Sequence of equivalence classes of attributes $S_1 = \{A\}$, $S_2 = \{B,C,D\}$ and $S_3 = \{E\}$ turns into the following sequence of non redundant equivalence classes of attributes: $T_1 = \{A\}$, $T_2 = \{B,C,D\}$.

Lemma 7. If X is a determinant under F of set $S_1 \cup ... \cup S_n$, then Z, where $Z = X \cap (S_1 \cup ... \cup S_j)$ and $j = \overline{1,n}$, is a determinant for $S_1 \cup ... \cup S_j$ under F.

Proof. According to Theorem 9, the expression $X \subseteq S_1 \cup ... \cup S_n$ takes place. First it will be shown that $Z \to (S_1 \cup ... \cup S_j) \in F^+$. Lets suppose the contrary: $Z \to (S_1 \cup ... \cup S_j) \notin F^+$. Then there exists a set Z', where $Z' \subseteq X$, which is a determinant of set $S_1 \cup ... \cup S_j$ and $Z' \cap (\bigcup_{i=j+1}^{n} S_i) \neq \varnothing$. Considering Lemma 6, there is a path from every vertex labeled with A in $Z' \cap (\bigcup_{i=j+1}^{n} S_i)$ that leads to a vertex B in $\bigcup_{i=1}^{j} S_i$. A contradiction has been encountered. Therefore, $Z \to (S_1 \cup ... \cup S_j) \in F^+$.

To complete the proof of this lemma, it will be shown that Z is a determinant under F of set $S_1 \cup ... \cup S_j$. Indeed, if it is considered that Z is not a determinant of F under F, then there must exist in Z an attribute A, such that $(Z - \{A\}) \to (S_1 \cup ... \cup S_j) \in F^+$. But then $(Z - \{A\}) \to Z \in F^+$ takes place, fact that implies $(X - \{A\}) \to X \in F^+$. So, a contradiction has been encountered, that X is a determinant of set $S_1 \cup ... \cup S_n$ under X.

Theorem 10. If set X of attributes is a determinant of set $S_1 \cup ... \cup S_n$, then $X \subseteq T_1 \cup ... \cup T_m$.

Proof. Let S_j be the first set of attributes that doesn't coincide with T_j and assume that there is an attribute A in X, such that $A \in S_j$ and $A \notin T_j$. Lemma 7 implies that $(X \cap (S_1 \cup ... \cup S_j)) \to (S_1 \cup ... \cup S_j) \in F^+$. Since $A \notin T_j$, then $(X \cap (S_1 \cup ... \cup S_j)) \to A \in F^+$. So $(X - \{A\}) \to X \in F^+$, thus X is not a determinant of set $S_1 \cup ... \cup S_n$ under F.

Appealing to Theorem 10, Lemma 7 can be paraphrased for non redundant equivalence classes of attributes.

Lemma 8. If X is a determinant under F of set $T_1 \cup ... \cup T_m$, then Z, where $Z = X \cap (T_1 \cup ... \cup T_j)$ and $j = \overline{1,m}$, is a determinant for $T_1 \cup ... \cup T_j$ under F.

5 Calculation of determinants using a scheme decomposition method

In this section, there are proposed theoretical tools that can be the basis of relational schemes decomposition algorithm for computing determinants related to the scheme, rather all independent components from which all determinants of scheme can be built. The scheme is partitioned in subschema to solve the determinants searching problem for each subschema separately. Then, for each subschema will be found determinants with the fewest attributes (including repeated). The groups of attributes from the set of functional dependencies that are determinants in some subschema are substituted with the shortest determinants. This happens only for equivalence classes of functional dependencies containing the determinants in the left or right sides as subsets.

Below there are considered the functional dependencies and non redundant equivalence classes of attributes. It is examined how the determinants of a non redundant equivalence class of attributes in relation to a set of functional dependencies in the projection of this set of dependencies on the attributes of non redundant equivalence class of attributes are reflected.

The next theorem presents a property of non redundant equivalence classes of attributes if the set of functional dependencies, on which these classes are built, is reduced.

Theorem 11. If the dependency $V \to W \in F$ is reduced and $W \cap T_j \neq \varnothing$, then $V \cap T_j \neq \varnothing$.

Proof. Appealing to the definition of contribution graph, from each vertex labeled with an attribute in V an edge leaves to every vertex labeled with an attribute in W. Then $V \cap T_i = \varnothing$ for $i = \overline{j+1, m}$. Assuming that $V \cap T_j = \varnothing$, then there is $(T_1 \cup ... \cup T_{j-1}) \to V \in F^+$. From this it follows that $(T_1 \cup ... \cup T_{j-1}) \to W \in F^+$. But in this case, T_j is redundant, fact that contradicts the nature of this set of attributes.

Proposition 4. Let $X \to Y \in F^+$ be a functional dependency. If $Y \subseteq T_1 \cup ... \cup T_j$ and dependency $V \to W \in F$ is non redundantly used in derivation of dependency $X \to Y$ under F, then $V \cap T_i = \varnothing$ for $i = \overline{j+1, m}$.

Veracity of this statement is based directly on the Corollary 5.

Definition 17. Let F be a set of functional dependencies over the set R of attributes. Projection of the set F of dependencies, labeled $\pi_Z(F)$, on a set Z of attributes, where $Z \subseteq R$, is the set of functional dependencies defined by the expression

$$\pi_Z(F) = \{(X \cap Z) \to (Y \cap Z) \mid X \to Y \in F \,\&\, (X \cap Z) \neq \varnothing \,\&\, (Y \cap Z) \neq \varnothing\}.$$

Then the following statement is true:

Lemma 9. If X, where $X \subseteq T_1 \cup ... \cup T_j$, is a determinant of the set T_j under F, then $(X \cap T_j) \to T_j \in \pi^+_{T_j}(F)$.

Proof. According to Theorem 8, $X \cap T_j \neq \varnothing$. Since $X \to T_j \in F^+$, following Corollary 3, there is a derivation $H = < X_0, X_1, ..., X_n >$ for dependency $X \to T_j$ under F.

Let $H' = < Z_0, Z_1, ..., Z_m >$ be the maximal derivation for $X \cap T_j$ under $\pi_{T_j}(F)$. Given the Lemma 7, to prove the lemma, it suffices to show that $T_j \subseteq Z_m$.

Indeed, given that $T_j \subseteq X_n$, then either $X = T_j$, or T_j is formed in H from dependencies which contain attributes of T_j in their right side. In the first case,

$X \cap T_j = T_j$ and the dependency $(X \cap T_j) \to T_j$ is deducted from any set of dependencies. In the second case, taking account of Theorem 11, the dependencies used in H, that have some attributes of T_j in their right sides, have also attributes of T_j in the left sides. Thus, if H has used the dependency $V \to W$ in F and $W \cap T_j \neq \varnothing$, then H' has used the dependency $(V \cap T_j) \to (W \cap T_j)$ in $\pi_{T_j}(F)$. Therefore, $T_j \subseteq Z_m$.

Further, it is established the relationship between determinants of scheme and its subschema determinants obtained via projections.

Theorem 12. Let $Sch(\bigcup_{i=1}^{n} S_i, F)$ be a database schema. The set X, where $X \subseteq T_1 \cup ... \cup T_m$, is a determinant for $T_1 \cup ... \cup T_m$ under F, if and only if $X \cap T_1, ..., X \cap T_m$ are determinants for $T_1, ..., T_m$ under $\pi_{T_1}(F), ..., \pi_{T_m}(F)$, respectively.

Proof. Necessity. Let the set X be a determinant for $T_1 \cup ... \cup T_m$ under F. It will be shown that $X \cap T_1, ..., X \cap T_m$ are determinants for $T_1, ..., T_m$ under $\pi_{T_1}(F), ..., \pi_{T_m}(F)$, respectively.

Will be proved this by applying mathematical induction on the number of non redundant equivalence classes, i, where $i = \overline{1, m}$. Let $i = 1$. Taking into account Lemma 8, $X \cap T_1$ is a determinant for T_1 under F. According to Lemma 9, $X \cap T_1$ is a determinant for T_1 in relation to $\pi_{T_1}(F)$.

It is assumed now that the assertion is fair for $T_1 \cup ... \cup T_{k-1}$, namely, for a number of classes less than k and will demonstrate that the affirmation is also true for a number of classes equal to k.

Since $X \cap (T_1 \cup ... \cup T_k) \to (T_1 \cup ... \cup T_k) \in F^+$, where, according to Lemma 8, $X \cap (T_1 \cup ... \cup T_k)$ is a determinant for $(T_1 \cup ... \cup T_k)$ under F, then $X \cap (T_1 \cup ... \cup T_k) \to T_k \in F^+$. If it is assumed that for an attribute A, where $A \in X \cap T_k$, the expression $(X \cap (T_1 \cup ... \cup T_k) \setminus \{A\}) \to T_k \in F^+$ holds, then, on the basis that $X \cap (T_1 \cup ... \cup T_i)$ is determinant for $(T_1 \cup ... \cup T_i)$, where $i = \overline{1, k-1}$, it follows that $(X \cap (T_1 \cup ... \cup T_k) \setminus \{A\}) \to (X \cap (T_1 \cup ... \cup T_k)) \in F^+$ takes place. But in this case, $X \cap (T_1 \cup ... \cup T_k)$ will not be determinant for $(T_1 \cup ... \cup T_k)$. Thus, there is $X' \subseteq X$, that $X \cap T_k \subseteq X'$ and X' is determinant for T_k under F. Whence, $X \cap T_k = X' \cap T_k$, is determinant for T_k under F and, according to Lemma 9, $X \cap T_k$ is determinant for T_k under $\pi_{T_k}(F)$.

Sufficiency. Now it will be proven that if the $X \cap T_1, ..., X \cap T_m$ are determinants for $T_1, ..., T_m$ under $\pi_{T_1}(F), ..., \pi_{T_m}(F)$, respectively, then the set Z, where $Z = X \cap T_1 \cup ... \cup X \cap T_m$, is determinant for $T_1 \cup ... \cup T_m$ under F.

It is obvious that $Z \to (T_1 \cup ... \cup T_m) \in F^+$, where $Z = X \cap T_1 \cup ... \cup X \cap T_m$.

Assuming that for at least one attribute A in $X \cap T_i$ the expression $(Z - \{A\}) \to (T_1 \cup ... \cup T_m) \in F^+$ holds, then, by virtue of Lemma 9, $X \cap T_i$ will not be determinant for T_i under $\pi_{T_i}(F)$. Consequently, $Z = X \cap T_1 \cup ... \cup X \cap T_m$, is determinant for $T_1 \cup ... \cup T_m$ under F. The theorem is proved.

Let $Sch(\bigcup_{i=1}^{n} S_i, F)$ be a relational schema, and let $T_1 \cup ... \cup T_m$ be the set of non redundant equivalence classes of attributes built on the set F of dependencies. Thus, each determinant X of the set $T_1 \cup ... \cup T_m$ under F consists of the union of determinants for the set T_i (one from each non redundant equivalence class of attributes) under $\pi_{T_i}(F)$, where $i = \overline{1,m}$. The problem of calculating the sets of attributes that can be substituted with other equivalent sets of attributes of smallest cardinality for each equivalence class in which the set F of dependencies of the scheme $Sch(\bigcup_{i=1}^{n} S_i, F)$ is partitioned, consists in finding all the determinants for each T_i.

It should be noted that the set of dependencies $\pi_{T_i}(F)$ may not be minimum, even if F is minimum. Moreover, it may be neither non redundant.

Example 9. If $F = \{C \to B$, $B \to C$, $AB \to D$, $AD \to B\}$, then there are two non redundant classes of attributes $T_1 = \{A\}$ and $T_2 = \{B, C, D\}$. Projection on class T_2 of the set F will be $\pi_{T_2}(F) = \{C \to B$, $B \to C$, $B \to D$, $D \to B\}$. Although the set F of functional dependencies is minimum, the set $\pi_{T_2}(F)$ is not minimum, because there is an equivalent set of dependencies $F' = \{C \to B$, $B \to CD$, $D \to B\}$ with fewer dependencies.

Example 10. If $F = \{C \to B$, $B \to C$, $AB \to D$, $AD \to C\}$, also there are two non redundant classes $T_1 = \{A\}$ and $T_2 = \{B, C, D\}$. In this case the projection $\pi_{T_2}(F) \equiv \{C \to B$, $B \to C$, $B \to D$, $D \to C\}$ is not non redundant, because the dependency $B \to C$ is redundant in $\pi_{T_2}(F)$. In other words, $\pi_{T_2}(F) \equiv \{C \to B$, $B \to D$, $D \to C\}$.

Therefore, before computing the determinants of each set T_i it is useful to minimize the set $\pi_{T_i}(F)$ of functional dependencies.

It is to mention that partitioning the set of attributes in classes of equivalence, essentially reduces dimensions of problem of computing the optimal cover. For this, it suffices to consider the set T_i, for example, which may include the determinant with minimum cardinality. The set T_i contains all attributes involved in the determinants for T_i under $\pi_{T_i}(F)$. These will form all the determinants for $T_1 \cup ... \cup T_m$ under F. In addition, left side of every dependency in $\pi_{T_i}(F)$, in essence, represents a determinant for T_i. In case $\pi_{T_i}(F) = \varnothing$, there is only one determinant, T_i itself.

The following is an integrator example.

Example 11. Let $F = \{ABC \to E,$ $BC \to D,$ $D \to BC\}$ be a minimum and reduced set of functional dependencies. Any set of functional dependencies can be minimized and reduced in polynomial time [1]. It is necessary to build an optimal cover of this set of dependencies.

The set F of functional dependencies is divided into equivalence classes of dependencies. Obviously, that on the equivalence classes of functional dependencies can be defined a strict partial order. Let $Attr(F_i)$ denote the set of attributes involved by dependencies of equivalence class F_i. The equivalence class

F_i precedes the equivalence class F_j if $Attr(F_j)^+ \subset Attr(F_i)^+$. In the considered example, the set of functional dependencies is divided into two equivalence classes $F = F_1 \cup F_2$, where $F_1 = \{ABC \to E\}$, and $F_2 = \{BC \to D, \quad D \to BC\}$.

The contribution graph for the set of dependencies F has the form represented in Figure 1. As noted already above, the set of vertices of the graph in Figure 1 is divided into three equivalence classes of attributes $S_1 = \{A\}$, $S_2 = \{B, C, D\}$ and $S_3 = \{E\}$, and are reduced to the following sequence of non redundant equivalent classes of attributes $T_1 = \{A\}$, $T_2 = \{B, C, D\}$.

The set F of functional dependencies, below, is projected on the sets of attributes T_1 and T_2, resulting in the following sets of functional dependencies $\pi_{T_1}(F) = \varnothing$, $\pi_{T_{21}}(F) = \{BC \to D, \quad D \to BC\}$. Thus, for the non redundant classes of attributes there were obtained the following sets of determinants $\{A\}$, $\{D, BC\}$, respectively.

Now the groups of attributes that are determinants and part of dependencies in F are substituted by those with the smallest length. Substitutions occur in the equivalence classes of dependencies which precede corresponding class that has generated the determinant. Therefore, the set of attributes BC of dependencies that are part of the equivalence class F_1 (there is only one dependency) is substituted by determinant D. Thus optimal cover is obtained as $F = \{AD \to E, \quad BC \to D, \quad D \to BC\}$.

6 Conclusions

It is known that various types of covers provide specific properties to database scheme. Referring to the problem of building optimal covers, it was found that it is the strictest structure of functional dependencies regarding the constituent elements.

Because the task of obtaining the optimal cover is classified as NP-complete problem, a way of achieving a solution, in acceptable time, is to apply a decomposition method to dive the original in smaller problems that could be solved, and then to combine particular solutions in order to construct the initial problem solution.

It should be mentioned that the proposed method does not change the complexity of the problem. Its nature continues to be NP-complete. However, it can be solved in such a way that it will reduce the time needed to impose constraints on database content and to reduce the time required to execute the algorithm for computing the closures.

References

[1] D. Maier, "The theory of relational database", Computer Science Press, 1983, 637 p.

[2] S. Sumathi, S. Esakkirajan, "Fundamentals of Relational Database Management Systems", 2007, Springer-Verlag Berlin and Heidelberg GmbH & Co. K., 776 p.

[3] J. Paredaens, "About Functional Dependencies in a Database Structure and their Coverings", Phillips MBLE Lab. Report 342, 1977.

[4] E. A. Lewis, L.C. Sekino, P. D. Ting, "A Canonical Representation for the Relational Schema and Logical Data Independence", *IEEE Computer Software and Applications Conf. (COMPSAC '77)*, 1977, pp.276-280.

[5] C. Beeri, Bernstein, A. Philip, "Computational problems related to the design of normal form relational schemas", *ACM Trans. Database Syst.*, 1979, V.4, N 1, pp.30-59.

[6] В. Котеля, Караниколов Аурел, “Минимальное покрытие в схеме базы данных”, *Мат.исслед., вып 87*, Штиинца, Кишинев, 1986, с.49-61.
[7] V. Cotelea, “An inference model for functional dependencies in database schemas”, *Meridian Ingineresc*, N.3, 2009, pp.89-92.
[8] J. D. Ullman, “On Kent’s “Consequences of assuming a universal relation””, *ACM Trans. Database Syst.*, 1983, V.8, N.4, p.637-643.
[9] S. Even, “Graph Algorithms”, Computer Science press, 1979, 250 p.

Vitalie COTELEA is Associate Professor at Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Moldova. He is the author and co-author over 90 scientific works, including one monograph and more than 10 books. His work focuses on Databases and Information Systems Design and Declarative Programming. He has graduated the Faculty of Mathematics and Cybernetics in 1974 of State University of Moldova, Chisinau. He holds a PhD diploma in Computer Science from 1988 of Kiev State University, Ukraine.