Acta Cybernetica **17** (2005) 173–182.

Some Results Related to Dense Families of Database Relations

Vu Duc Thi* and Nguyen Hoang Son†

Abstract

The dense families of database relations were introduced by Järvinen [7]. The aim of this paper is to investigate some new properties of dense families of database relations, and their applications. That is, we characterize functional dependencies and minimal keys in terms of dense families. We give a necessary and sufficient condition for an abitrary family to be $R-$ dense family. We prove that with a given relation R the equality set E_R is an $R-$dense family whose size is at most $\frac{m(m-1)}{2}$, where m is the number of tuples in R. We also prove that the set of all minimal keys of relation R is the transversal hypergraph of the complement of the equality set E_R. We give an effective algorithm finding all minimal keys of a given relation R. We aslo give an algorithm which from a given relation R finds a cover of functional dependencies that holds in R. The complexity of these algorithms is also esimated.

1 Basic definitions

In this section we present briefly the main concepts of the theory of relational databases which will be needed in sequel. The concepts and facts given in this section can be found in [1, 3, 4, 8, 9].

Let U be a finite set of *attributes* (e.g. name, age etc). The elements of U will be denoted by $a, b, c, \ldots, x, y, z$, if an ordering on U is needed, by $a_1, \ldots, a_n$. A map *dom* associates with each $a \in U$ its *domain* $dom(a)$. A *relation* R over U is a subset of Cartesian product $\prod_{a \in U} dom(a)$.

We can think of a relation R over U as being a set of tuples: $R = \{h_1, \ldots, h_m\}$,

$$h_i : U \longrightarrow \bigcup_{a \in U} dom(a), h_i(a) \in dom(a), i = 1, 2, \ldots, m.$$

A *functional dependency* (FD for short) is a statement of form $X \to Y$, where $X, Y \subseteq U$. The FD $X \to Y$ holds in a relation $R = \{h_1, \ldots, h_m\}$ over U if

$$(\forall h_i, h_j \in R)((\forall a \in X)(h_i(a) = h_j(a)) \Rightarrow (\forall b \in Y)(h_i(b) = h_j(b))).$$

*Institute of Information Technology, Vietnamese Academy of Science and Technology, 18 Hoang Quoc Viet, Hanoi, Vietnam.

†Department of Mathematics, College of Sciences, Hue University, Vietnam.

We also say that R satisfies the FD $X \to Y$.

Let F_R be a family of all FDs that holds in R. Then $F = F_R$ satisfies

(F1) $X \to X \in F$,
(F2) $(X \to Y \in F, Y \to Z \in F) \Rightarrow (X \to Z \in F)$,
(F3) $(X \to Y \in F, X \subseteq V, W \subseteq Y) \Rightarrow (V \to W \in F)$,
(F4) $(X \to Y \in F, V \to W \in F) \Rightarrow (X \cup V \to Y \cup W \in F)$.

A family of FDs satisfying (F1) - (F4) is called an $f - family$ over U.

Clearly, F_R is an f-family over U. It is known [1] that if F is an arbitrary$f$-family, then there is a relation R over U such that $F_R = F$.

Given a family F of FDs over U, there exists a unique minimal f-family F^+ that contains F. It can be seen that F^+ contains all FDs which can be derived from F by the rules (F1) - (F4).

A *relation scheme s* is a pair (U, F), where U is a set of attributes and F is a set of FDs over U.

Let U be a nonempty finite set and $\mathcal{P}(U)$ its power set. The mapping $\mathcal{L} : \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ is called a *closure operation* over U if it satisfies the following conditions:

(1) $X \subseteq \mathcal{L}(X)$,
(2) $X \subseteq Y$ implies $\mathcal{L}(X) \subseteq \mathcal{L}(Y)$,
(3) $\mathcal{L}(\mathcal{L}(X)) = \mathcal{L}(X)$.

Remark 1.1. It is clear that, if F is an $f-$ family, and we define $\mathcal{L}_F(X)$ as

$$\mathcal{L}_F(X) = \{a \in U : X \to \{a\} \in F\}$$

then $\mathcal{L}_F$ is a closure operation over U. Conversely, it is known [1, 3] that if $\mathcal{L}$ is a closure operation, then there is exactly one $f-$ family F over U so that $\mathcal{L} = \mathcal{L}_F$, where

$$F = \{X \to Y : X, Y \subseteq U, Y \subseteq \mathcal{L}(X)\}.$$

Thus, there is a one-to-one correspondence between closure operations and $f-$ families over U.

Let R be a relation over U and $K \subseteq U$. Then K is a *key* of R if $K \to U \in F_R$. K is a *minimal key* of R if K is a key of R and any proper subset of K is not a key of R.

Denote K_R the set of all minimal keys of R.

Let $I \subseteq \mathcal{P}(U)$, $U \in I$, and $A, B \in I \Rightarrow A \cap B \in I$. I is called a meet-semilattice over U. Let $M \subseteq \mathcal{P}(U)$. Denote $M^+ = \{\cap M' : M' \subseteq M\}$. We say that M is a generator of I if $M^+ = I$. Note that $U \in M^+$ but not in M, by convention it is the intersection of the empty collection of sets.

Denote $N = \{A \in I : A \neq \cap\{A' \in I : A \subset A'\}\}$. It can be seen that N is the unique minimal generator of I.

2 Hypergraphs and Transversals

Let U be a nonempty finite set and put $\mathcal{P}(U)$ for the family of all subsets of U. The family $\mathcal{H} = \{E_i : E_i \in \mathcal{P}(U), i = 1, 2, \ldots, m\}$ is called a *hypergraph* over U if $E_i \neq \emptyset$ holds for all i (in [2] it is required that the union of $E_i s$ is U, in this paper we do not require this).

The elements of U are called vertices, and the sets $E_1, \ldots, E_m$ the edges of the hypergraph $\mathcal{H}$.

A hypergraph $\mathcal{H}$ is called *simple* if it satisfies $\forall E_i, E_j \in \mathcal{H} : E_i \subseteq E_j \Rightarrow E_i = E_j$. It can be seen that K_R is a simple hypergraph.

Let $\mathcal{H}$ be a hypergraph over U. Then $min(\mathcal{H})$ denotes the set of minimal edges of $\mathcal{H}$ with respect to set inclusion, i.e., $min(\mathcal{H}) = \{E_i \in \mathcal{H} : \nexists E_j \in \mathcal{H} : E_j \subset E_i\}$, and $max(\mathcal{H})$ denotes the set of maximal edges of $\mathcal{H}$ with respect to set inclusion, i.e., $max(\mathcal{H}) = \{E_i \in \mathcal{H} : \nexists E_j \in \mathcal{H} : E_j \supset E_i\}$.

It is clear that, $min(\mathcal{H})$ and $max(\mathcal{H})$ are simple hypergraphs. Furthermore, $min(\mathcal{H})$ and $max(\mathcal{H})$ are uniquely determined by $\mathcal{H}$.

A set $T \subseteq U$ is called a *transversal* of $\mathcal{H}$ (sometimes it is called *hitting set*) if it meets all edges of $\mathcal{H}$, i.e., $\forall E \in \mathcal{H} : T \cap E \neq \emptyset$. Denote by $Trs(\mathcal{H})$ the family of all transversals of $\mathcal{H}$. A transversal T of $\mathcal{H}$ is called *minimal* if no proper subset T' of T is a transversal.

The family of all minimal transversals of $\mathcal{H}$ called the transversal hypergraph of $\mathcal{H}$, and denoted by $Tr(\mathcal{H})$. Clearly, $Tr(\mathcal{H})$ is a simple hypergraph.

Proposition 2.1 ([2]). *Let $\mathcal{H}$ and $\mathcal{G}$ two simple hypergraphs over U. Then*

(1) $\mathcal{H} = Tr(\mathcal{G})$ if and only if $\mathcal{G} = Tr(\mathcal{H})$,
(2) $Tr(\mathcal{H}) = Tr(\mathcal{G})$ if and only if $\mathcal{H} = \mathcal{G}$,
(3) $Tr(Tr(\mathcal{H})) = \mathcal{H}$.

By the definition of minimal transversal, the following proposition is obvious

Proposition 2.2. *Let $\mathcal{H}$ be a hypergraph over U. Then*

$$Tr(\mathcal{H}) = Tr(min(\mathcal{H})).$$

The following algorithm finds the family of all minimal transversals of a given hypergraph (by induction).

Algorithm 2.3 ([5]).

Input: let $\mathcal{H} = \{E_1, \ldots, E_m\}$ be a hypergraph over U.

Output: $Tr(\mathcal{H})$.

Method:

Step 0. We set $L_1 := \{\{a\} : a \in E_1\}$. It is obvious that $L_1 = Tr(\{E_1\})$.

Step q+1. ($q < m$) Assume that

$$L_q = S_q \cup \{B_1, \ldots, B_{t_q}\},$$

where $B_i \cap E_{q+1} = \emptyset, i = 1, \ldots, t_q$ and $S_q = \{A \in L_q : A \cap E_{q+1} \neq \emptyset\}$.

For each i ($i = 1, \ldots, t_q$) constructs the set $\{B_i \cup \{b\} : b \in E_{q+1}\}$. Denote them by $A_1^i, \ldots, A_{r_i}^i (i = 1, \ldots, t_q)$. Let

$$L_{q+1} = S_q \cup \{A_p^i : A \in S_q \Rightarrow A \not\subset A_p^i, 1 \leq i \leq t_q, 1 \leq p \leq r_i\}.$$

Theorem 2.4 ([5]). *For every* $q (1 \leq q \leq m) L_q = Tr(\{E_1, \ldots, E_q\})$*, i.e.,* $L_m = Tr(\mathcal{H})$.

It can be seen that the determination of $Tr(\mathcal{H})$ based on our algorithm does not depend on the order of $E_1, \ldots, E_m$.

Remark 2.5. Denote $L_q = S_q \cup \{B_1, \ldots, B_{t_q}\}$, and $l_q (1 \leq q \leq m-1)$ be the number of elements of L_q. It can be seen that the worst-case time complexity of our algorithm is

$$\mathcal{O}(|U|^2 \sum_{q=0}^{m-1} t_q u_q),$$

where $l_0 = t_0 = 1$ and

$$u_q = \begin{cases} l_q - t_q, & \text{if } l_q > t_q; \\ 1, & \text{if } l_q = t_q. \end{cases}$$

Clearly, in each step of our algorithm L_q is a simple hypergraph. It is known that the size of arbitrary simple hypergraph over U cannot be greater than $C_n^{[n/2]}$, where $n = |U|$. $C_n^{[n/2]}$ is asymptotically equal to $2^{n+1/2}/(\pi.n)^{1/2}$. From this, the worst-case time complexity of our algorithm cannot be more than exponential in the number of attributes. In cases for which $l_q \leq l_m (q = 1, ..., m-1)$, it is easy to see that the time complexity of our algorithm is not greater than $\mathcal{O}(|U|^2 |\mathcal{H}| |Tr(\mathcal{H})|^2)$. Thus, in these cases this algorithm finds $Tr(\mathcal{H})$ in polynomial time in $|U|, |\mathcal{H}|$ and $|Tr(\mathcal{H})|$. Obviously, if the number of elements of $\mathcal{H}$ is small, then this algorithm is very effective. It only requires polynomial time in $|R|$.

The following proposition is obvious

Proposition 2.6 ([5]). *The time complexity of finding* $Tr(\mathcal{H})$ *of a given hypergraph* $\mathcal{H}$ *is (in general) exponential in the number of elements of* U.

Proposition 2.6 is still true for a simple hypergraph.

3 Dense Families

Let $\mathcal{D} \subseteq \mathcal{P}(U)$ be a family of subsets of a U. We define a set $F_{\mathcal{D}}$ over $\mathcal{D}$ as follows

$$F_{\mathcal{D}} = \{X \to Y : (\forall A \in \mathcal{D}) X \subseteq A \Rightarrow Y \subseteq A\}.$$

Proposition 3.1 ([7]). *If $\mathcal{D}$ is a family of subsets of a finite set U, then $F_{\mathcal{D}}$ is an $f-$ family over U.*

The notion of dense family of a database relation is defined in [7], as follows:

Let R be a relation over U. We say that a family $\mathcal{D} \subseteq \mathcal{P}(U)$ of attribute sets is $R-$*dense* (or *dense in* R) if $F_R = F_{\mathcal{D}}$.

The following proposition guarantees the existence of at least one dense family. In the sequel we denote $\mathcal{L}_{F_R}$ simply by $\mathcal{L}_R$.

Proposition 3.2 ([7]). *The family $\mathcal{L}_R$ is $R-$dense.*

Proposition 3.3 ([7]). *If $\mathcal{D}$ is $R-$dense, then $\mathcal{D} \subseteq \mathcal{L}_R$.*

Note that by Proposition 3.2 and Proposition 3.3, $\mathcal{L}_R$ is the greatest $R-$dense family.

For any $A \subseteq U$, we denote by $\overline{A}$ the *complement* of A with respect to the set U, that is, $\overline{A} = \{a \in U : a \notin A\}$.

Theorem 3.4 ([7]). *Let R be a relation over U. If $\mathcal{D} \subseteq \mathcal{P}(U)$ is $R-$dense, then the following conditions hold*

(1) K is a key of R if and only if it contains an element from each set in $\{\overline{A} : A \in \mathcal{D}, A \neq U\}$.

(2) K is a minimal key of R if and only if it minimal with respect to the property of containing an element from each set in $\{\overline{A} : A \in \mathcal{D}, A \neq U\}$.

Let U be a finite set and $\mathcal{P}(U)$ its power set. For every family $\mathcal{D} \subseteq \mathcal{P}(U)$, the complement family of $\mathcal{D}$ is the family $\overline{\mathcal{D}} = \{\overline{A} : A \in \mathcal{D}\}$ over U.

Let $R = \{h_1, \ldots, h_m\}$ be a relation over U, and E_R the *equality set* of R, i.e.,

$$E_R = \{E_{ij} : 1 \leq i < j \leq m\}$$

where $E_{ij} = \{a \in U : h_i(a) = h_j(a)\}$.

Proposition 3.5. *The equality set E_R is $R-$dense.*

Proof. Assume that $X \to Y \in F_R$. Let $E_{ij} \in E_R$ such that $X \subseteq E_{ij}$. This means that $h_i(X) = h_j(X)$. From this, and according to the definition of FDs, we have $h_i(Y) = h_j(Y)$. Thus, $Y \subseteq E_{ij}$. By the definition of F_{E_R}, that is,

$$F_{E_R} = \{X \to Y : (\forall E_{ij} \in E_R) X \subseteq E_{ij} \Rightarrow Y \subseteq E_{ij}\},$$

we obtain $X \to Y \in F_{E_R}$.

Conversely, let $X \to Y \in F_{E_R}$. Suppose that there are $h_i, h_j \in R$ such that $h_i(X) = h_j(X)$, $1 \leq i < j \leq m$. Which means that $X \subseteq E_{ij}$. By $X \to Y \in F_{E_R}$, $Y \subseteq E_{ij}$. Hence, we also obtain $h_i(Y) = h_j(Y)$. Consequently, $X \to Y \in F_R$.

The proposition is proved. □

It is easy to see that the dense family E_R has at most $\frac{m(m-1)}{2}$ elements. By Proposition 3.3, we also have $E_R \subseteq \mathcal{L}_R$.

Theorem 3.6. *Let R be a relation over U. Then*

$$K_R = Tr(min(\overline{E_R})).$$

Proof. By the definition of relation R, we have $U \notin E_R$. From this, Proposition 2.2, Proposition 3.5 and Theorem 3.4, the theorem is obvious.

The proof is complete. □

Let $R = \{h_1, \ldots, h_m\}$ be a relation over U, and N_R the *nonequality set* of R, i.e.,

$$N_R = \{N_{ij} : 1 \leq i < j \leq m\}$$

where $N_{ij} = \{a \in U : h_i(a) \neq h_j(a)\}$.

Note that, because R is a relation, $\emptyset \notin N_R$ and $U \notin E_R$. Moreover, $N_R = \overline{E_R}$. From this, and Theorem 3.6, the following corollary is immediate

Corollary 3.7. *Let R be a relation over U. Then*

$$K_R = Tr(min(N_R)).$$

Corollary 3.7 was shown in [5].

Proposition 3.8. *If $\mathcal{D}$ is $R-$ dense, then*

$$min(\overline{\mathcal{D}} - \{\emptyset\}) = \overline{max(E_R)}.$$

Proof. According to Theorem 3.6, we have $K_R = Tr(\overline{E_R})$. By Proposition 2.2, it is clear that

$$K_R = Tr(\overline{max(E_R)}). \tag{1}$$

Because $\mathcal{D}$ is $R-$ dense, and by Theorem 3.4, we have $K_R = Tr(\overline{\mathcal{D}} - \{\emptyset\})$. Furthermore, we have

$$Tr(\overline{\mathcal{D}} - \{\emptyset\}) = Tr(min(\overline{\mathcal{D}} - \{\emptyset\})).$$

Hence

$$K_R = Tr(min(\overline{\mathcal{D}} - \{\emptyset\})). \tag{2}$$

From (1) and (2), we give

$$Tr(min(\overline{\mathcal{D}} - \{\emptyset\})) = Tr(\overline{max(E_R)}).$$

By $min(\overline{\mathcal{D}} - \{\emptyset\})$ and $\overline{max(E_R)}$ are simple hypergraphs, thus according to Proposition 2.1 we have

$$min(\overline{\mathcal{D}} - \{\emptyset\}) = \overline{max(E_R)}.$$

The proposition is proved. □

From Proposition 3.8, the following corollary is clear

Corollary 3.9. *If $\mathcal{D}$ is $R-$ dense, then*

$$min(\overline{\mathcal{D}} - \{\emptyset\}) = min(N_R).$$

Now we give a necessary and sufficient condition for an arbitrary family $\mathcal{D}$ is $R-$ dense.

Theorem 3.10. *Let R be a relation, $\mathcal{D} \subseteq \mathcal{P}(U)$ a family of subsets of a U. Then $\mathcal{D}$ is $R-$ dense iff for every $X \subseteq U$*

$$\mathcal{L}_R(X) = \begin{cases} \bigcap\limits_{X \subseteq A} A & \text{if } \exists A \in \mathcal{D} : X \subseteq A, \\ U & \text{otherwise,} \end{cases}$$

where $\mathcal{L}_R(X) = \{a \in U : X \to \{a\} \in F_R\}$.

Proof. First we prove that in an arbitrary family $\mathcal{D} \subseteq \mathcal{P}(U)$ for all $X \subseteq U$

$$\mathcal{L}_{F_{\mathcal{D}}}(X) = \begin{cases} \bigcap\limits_{X \subseteq A} A & \text{if } \exists A \in \mathcal{D} : X \subseteq A, \\ U & \text{otherwise.} \end{cases}$$

Suppose that X is a set such that there is no $A \in \mathcal{D}$ with $X \subseteq A$. By the definition of $F_{\mathcal{D}}$, it is easy to see that $X \to U \in F_{\mathcal{D}}$. Hence, $\mathcal{L}_{F_{\mathcal{D}}}(X) = U$.

Since $\emptyset \subseteq \bigcap_{A \in \mathcal{D}} A \subseteq A$, according to the definition of $F_{\mathcal{D}}$ and $\mathcal{L}_{F_{\mathcal{D}}}$ we obtain

$$\mathcal{L}_{F_{\mathcal{D}}}(\emptyset) = \bigcap_{A \in \mathcal{D}} A.$$

If $X \neq \emptyset$ and there is an $A \in \mathcal{D}$ such that $X \subseteq A$ then we set

$$\mathcal{G} = \{A : X \subseteq A, A \in \mathcal{D}\},$$

$$B = \bigcap_{A \in \mathcal{G}} A.$$

It is easy to see that $X \subseteq B$ holds. If $\mathcal{G} = \mathcal{D}$ or $\mathcal{G} \neq \mathcal{D}$, then we also obtain $X \to B \in F_{\mathcal{D}}$.

By the definition of $\mathcal{L}_{F_{\mathcal{D}}}$, we have $B \subseteq \mathcal{L}_{F_{\mathcal{D}}}(X)$. Using $X \subseteq B \subseteq \mathcal{L}_{F_{\mathcal{D}}}(X)$, we obtain $B \to \mathcal{L}_{F_{\mathcal{D}}}(X) \in F_{\mathcal{D}}$.

Now we suppose that b is an attribute such that $b \notin B$. Then, there is $A \in \mathcal{G}$ so that $b \notin A$. Hence, by the definition of $F_{\mathcal{D}}$ we have $B \to B \cup \{b\} \notin F_{\mathcal{D}}$. Consequently,

$$\mathcal{L}_{F_{\mathcal{D}}}(X) = \bigcap_{A \in \mathcal{D}} (A).$$

By Remark 1.1 it is easy to see that $F_R = F_{\mathcal{D}}$ holds iff $\mathcal{L}_R = \mathcal{L}_{F_{\mathcal{D}}}$ does.

The Theorem is proved. □

From Theorem 3.10 and Proposition 3.5, the following proposition is obvious

Proposition 3.11. *Let $R = \{h_1, \dots, h_m\}$ be a relation over $U = \{a_1, \dots, a_n\}$. Then*

(1) If $\mathcal{D}$ is $R-$ dense, then $\mathcal{D} \cup \{U\}$ also is $R-$dense, and thus $E_R \cup \{U\}$ is $R-$dense.

(2) If $m = 1$ or $F_R = \{\{a_1\} \to U, \dots, \{a_n\} \to U\}$, then families $\mathcal{D}_1 = \emptyset$, $\mathcal{D}_2 = \{\emptyset\}$ and $\mathcal{D}_3 = \{U\}$ are $R-$denses.

4 Finding the set of all minimal keys of a relation

In this section, we give the following algorithm finding all minimal keys of a given relation R. Remember that this problem is inherently exponential in the size of R [4].

Algorithm 4.1.

Input: a relation $R = \{h_1, \dots, h_m\}$ over U.

Output: K_R.

Method:

Step 1. Construct the equality set

$$E_R = \{E_{ij} : 1 \leq i < j \leq m\}$$

where $E_{ij} = \{a \in U : h_i(a) = h_j(a)\}$.

Step 2. Compute the complement of E_R as follows

$$\overline{E_R} = \{\overline{E_{ij}} : E_{ij} \in E_R\}.$$

Denote elements of $\overline{E_R}$ by $N_1, \dots, N_k$

Step 3. From $\overline{E_R}$ compute the family $min(\overline{E_R}) = \{N_i \in \overline{E_R} : \not\exists N_j \in \overline{E_R} : N_i \subseteq N_j\}$.

Step 4. By Algorithm 2.3 we construct the set $Tr(min(\overline{E_R}))$.

Based on Proposition 2.2, Algorithm 2.3 and Theorem 3.6, we have $K_R = Tr(min(\overline{E_R}))$. It can be seen that the time complexity of this algorithm is the time complexity of Algorithm 2.3. In many cases this algorithm is very effective (see Remark 2.5).

It can be seen that, if the number of elements of the equality set E_R is constant, i.e. $|E_R| \leq k$ for some constant k, then the time complexity of finding K_R of a given relation R is polynomial time [9].

The following example shows that for a given relation R, Algorithm 4.1 can be applied to find all minimal keys of a given relation R.

Example 4.2. Let us consider the relation R over $U = \{a, b, c, d\}$ as follows

$$R = \begin{array}{cccc} a & b & c & d \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 4 & 0 & 4 & 4 \\ 5 & 5 & 5 & 0 \end{array}$$

It can be seen that the equality set E_R is the following

$E_R = \{\emptyset, \{b\}, \{c\}, \{d\}, \{b, c\}, \{c, d\}, \{a, b, c\}, \{b, c, d\}\}$.

Hence

$\overline{E_R} = \{\{a\}, \{d\}, \{a, d\}, \{a, b\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, U\}$,

$min(\overline{E_R}) = \{\{a\}, \{d\}\}$.

From this, we obtain

$K_R = \{\{a, d\}\}$.

5 Finding the cover of a relation

From Proposition 3.5 and Theorem 3.10 we have an application, which is the following algorithm finding a cover of FDs of a given relation R. Recall that this problem is inherently exponential in the size of R [6].

Algorithm 5.1.

Input: a relation $R = \{h_1, \ldots, h_m\}$ over U.

Output: F_R.

Method:

Step 1. Construct the equality set

$$E_R = \{E_{ij} : 1 \leq i < j \leq m\}$$

where $E_{ij} = \{a \in U : h_i(a) = h_j(a)\}$.

Step 2. Compute the family $E_R^+ = \{\cap \mathcal{A} : \mathcal{A} \subseteq E_R\}$. Denote the elements of E_R^+ by $X_1, \ldots, X_t$.

Step 3. Construct set of FDs as follows

$$F = \{K_1 \to X_1 : K_1 \in Key(X_1)\} \cup \cdots \cup \{K_t \to X_t : K_t \in Key(X_t)\}$$

where $Key(X_i)$ is a set of all minimal keys of $\Pi_{X_i}(R)$ (the projection of R onto the attributes set X_i).

Obviously, $F = F_R$. Note that $\mathcal{L}_R = E_R^+$. It is easy to see that the time complexity of this algorithm is exponential in the number of attributes.

The following example shows that for a given relation R, Algorithm 5.1 can be applied to find a cover of a given relation R.

Example 5.2. R is the following relation over $U = \{a, b, c, d\}$

$$R = \begin{matrix} a & b & c \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{matrix}$$

It can be seen that the equality set E_R is the following
$E_R = \{\{c\}, \{a, c\}, \{b, c\}\}$.
Therefore
$E_R^+ = \{\{c\}, \{a, c\}, \{b, c\}, U\}$.
From this, we have
$F = \{\{a\} \to \{c\}, \{b\} \to \{c\}, \{a, b\} \to \{c\}\}$.
It is obvious that $F = F_R$.

References

[1] Armstrong W. W., *Dependency structure of database relationship*, Information Processing 74, North-Holland Pub. Co. , (1974) 580-583.

[2] Berge C., *Hypergraphs: combinatorics of finite sets*, North - Holland, Amsterdam (1989).

[3] Demetrocis J., *On the equivalence of candidate keys with Sperner systems*, Acta Cybernetica **4**, (1979), 247-252.

[4] Demetrovics J., Thi V.D., *Keys, antikeys and prime attributes*, Annales Univ. Sci. Budapest Sect. Comp. **8**, (1987), 35-52.

[5] Demetrovics J., Thi V. D., *Describing candidate keys by hypergraphs*, Computers and Artificial Intelligence **18, 2** (1999), 191-207.

[6] Gottlob G., Libkin L., *Investigations on Armstrong relations, denpendency inference, and excluded functional dependencies*, Acta Cybernetica Hungary **9, 4** (1990), 385-402.

[7] Järvinen J., *Dense families and key functions of database relation instances*, in: Freivalds R. (ed.), Fundamentals of Computation Theory, Proceedings of the 13th International Symposium, Lecture Notes in Computer Science **2138** (Springer-Verlag, Heidelberg, 2001), 184-192.

[8] Thi V. D., *Minimal keys and antikeys*, Acta Cybernetica **7** (1986), 361-371.

[9] Thi V. D., Son N. H., *Some problems related to keys and the Boyce-Codd normal form*, Acta Cybernetica **16, 3** (2004), 473-483.

Received December, 2004