Journal of Computer Science and Cybernetics, V.25, N.2 (2009), 125–134

THE DENSE FAMILIES OF RELATION SCHEMES AND ITS APPLICATIONS

VU DUC THI[1], NGUYEN HOANG SON[2]

[1]*Institute of Information Technology, VAST, Hanoi*

[2]*Department of Mathematics, College of Sciences, Hue University*

Abstract. The dense families of relation schemes were introduced in [11]. The Armstrong relation is an essential concept in investigating the relational data model. Constructing Armstrong relation is a practically importance problem (see, e.g., [8]). The aim of the paper is to continue investigating some new properties of dense families. Applications of the results in studying the time complexity of the problem constructing Armstrong relation are given.

Tóm tắt. Họ trù mật của lược đồ quan hệ được giới thiệu trong [11]. Quan hệ Armstrong là một khái niệm cốt yếu trong nghiên cứu về mô hình dữ liệu quan hệ. Xây dựng quan hệ Armstrong là bài toán có tầm quan trọng trong thực tế (chẳng hạn xem [8]). Mục đích của bài báo này là tiếp tục nghiên cứu một số tính chất của họ trù mật. Ứng dụng các kết quả này vào nghiên cứu độ phức tạp của bài toán xây dựng quan hệ Armstrong trong lớp BCNF.

1. INTRODUCTION

The relational data model introduced by Codd [4] in 1970 is one of the most powerful database models. The basic concept of this model is a relation. It is a table, every row of which corresponds to a record and every column to an attribute. Semantic constraints between sets of attributes play an important role in logical and structural investigations of the relational data model, and in both practice and design theory. Informally, FD means that some attributes values can be unambiguously reconstructed by the others. The concept of Armstrong relation for FD was introduced by Fagin (see, e.g., [2]). An Armstrong relation for a set of FDs is a relation that satisfies each FD implied by the set but no FD that is not implied by it. Hypergraph theory is an important subfield of discrete mathematics with many relevant applications in both theoretical and applied computer science.

The dense families of relation schemes were introduced in [11] (2005). We have characterized minimal keys and antikeys of a relation scheme in terms of dense families.

The paper is organized as follows. In Section 2, some basic concepts and results on the theory of relational databases and hypergraphs are given. In Section 3, we introduce the notion of dense families of relation schemes and investigate some properties and applications of dense families. In Section 4, we prove the time complexity of problem constructing Armstrong relation by dense families and hypergraphs. The paper ends with conclusion.

2. BASIC DEFINITIONS

In this section, we begin by recalling some main concepts of the theory of relational databases that can be found in [1, 5, 6, 8].

Let U be a nonempty finite set of *attributes*. 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 on U is a subset of Cartesian product $\prod_{a\in U} dom(a)$.

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

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

The concept of FD between sets of attributes was introduced by Armstrong [1]. A *FD* is a statement of form $X \to Y$, where $X, Y \subseteq U$. The FD $X \to Y$ *holds* in a relation $R = \{h_1, \ldots, h_m\}$ on U if

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

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

This means that the values of the X component of tuples uniquely determine the values of the Y component.

Let F_R be a family of all FDs that holds in R.

It is obvious that $F = F_R$ 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* on U.

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

Given a family F of FDs on 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* $\mathcal{S}$ is a pair (U, F), where U is a set of attributes and F is a set of FDs on U. Denote $X^+ = \{a \in U : X \to \{a\} \in F^+\}$. X^+ is called the *closure* of X on $\mathcal{S}$. It is obvious that $X \to Y \in F^+$ if and only if $Y \subseteq X^+$.

Subset K of U is called a *key* of $\mathcal{S}$ (resp. R) if $K \to U \in F^+$ (resp. $K \to U \in F_R$). K is a *minimal key* of $\mathcal{S}$ (resp. R) if K is a key of $\mathcal{S}$ (resp. R) and any proper subset of K is not a key of $\mathcal{S}$ (resp. R). Denote by $\mathcal{K}_{\mathcal{S}}$ (resp. $\mathcal{K}_R$) the set of all minimal keys of $\mathcal{S}$ (resp. R).

The set of *antikeys* of $\mathcal{K}_{\mathcal{S}}$ (resp. $\mathcal{K}_R$), denoted by $\mathcal{K}_{\mathcal{S}}^{-1}$ (resp. $\mathcal{K}_R^{-1}$), is defined as follows: a set $A \in \mathcal{K}_{\mathcal{S}}^{-1}$ (resp. $\mathcal{K}_R^{-1}$) iff

(A1) no subset of A is a key of $\mathcal{S}$ (resp. R), and

(A2) A is maximal with respect to this property in the sense that all proper supersets C of A contain at least one key of $\mathcal{S}$ (resp. R).

Hence, it is easy to see that the elements of $\mathcal{K}_{\mathcal{S}}^{-1}$ and $\mathcal{K}_{R}^{-1}$ are maximal non-keys. Moreover, $\mathcal{K}_{\mathcal{S}}^{-1}$ (resp. $\mathcal{K}_{R}^{-1}$) is uniquely determined by $\mathcal{K}_{\mathcal{S}}$ (resp. $\mathcal{K}_{R}$).

$\mathcal{S} = (U, F)$ is in *Boyce-Codd normal form* (BCNF) if $X \to \{a\} \notin F^+$ for $X^+ \neq U$ and $a \notin X$. If a relation scheme is changed to a relation we have the definition of BCNF for relation.

Let $\mathcal{S} = (U, F)$ be a relation scheme. Clearly, if $\mathcal{S} = (U, F)$ is a relation scheme, then there is a relation R on U such that $F_R = F^+$ (see, [1]). Such a relation is called an *Armstrong relation* of $\mathcal{S}$. Evidently, all FDs of $\mathcal{S}$ hold in R.

Now, we introduce some basic concepts about hypergraphs, which will be used in the sequel. The concepts and facts given in this section can be found in [3, 7, 9, 10].

Let U be a nonempty finite set and put $\mathcal{P}(U)$ for the family of all subsets of U. The family $\mathcal{H} = \{E_1, E_2, \ldots, E_m\} \subseteq \mathcal{P}(U)$ is called a *hypergraph* on U if $E_i \neq \emptyset$ holds for all i (in [3] it is required that the union of $E_i s$ is U, in this paper this requirement is removed).

The elements of U are called *vertices*, and the sets $E_1, \ldots, E_m$ are 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 verify that $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_R$ are simple hypergraphs.

In this paper, we always assume that if simple hypergraph $\mathcal{H}$ plays a role of the set of minimal keys (resp. antikeys, i.e., maximal non-keys), then $\mathcal{H} \neq \emptyset$ and $\emptyset \notin \mathcal{H}$ (resp., $\emptyset, U \notin \mathcal{H}$). We consider comparison of two attributes as an elementary step of algorithms. Thus, if we assume that subsets of U are represented as sorted lists of attributes, then a Boolean operation on two subsets requires at most $|U|$ elementary steps.

Let $\mathcal{H}$ be a hypergraph on U. Then $\min(\mathcal{H})$ denotes the set of minimal edges of $\mathcal{H}$ with respect to set inclusion, i.e.,

$$\min(\mathcal{H}) = \{E_i \in \mathcal{H} : \forall E_j \in \mathcal{H} \Rightarrow E_j \not\subseteq E_i\},$$

and $\max(\mathcal{H})$ denotes the set of maximal edges of $\mathcal{H}$ with respect to set inclusion, i.e.,

$$\max(\mathcal{H}) = \{E_i \in \mathcal{H} : \forall E_j \in \mathcal{H} \Rightarrow E_j \not\supseteq E_i\}.$$

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 $\mathcal{H}$ is called the *transversal hypergraph* of $\mathcal{H}$, and denoted by $Tr(\mathcal{H})$. Clearly, $Tr(\mathcal{H})$ is a simple hypergraph.

Proposition 2.1. [3] *Let $\mathcal{H}$ and $\mathcal{G}$ be two simple hypergraphs on 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) $Tr(Tr(\mathcal{H})) = \mathcal{H}$.

The following obvious result will be used in the sequel.

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

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

Algorithm 2.1. ([7])
Input: let $\mathcal{H} = \{E_1, \ldots, E_m\}$ be a hypergraph on U.
Output: $Tr(\mathcal{H})$.
Method:
Step 0. We set $\mathcal{L}_1 = \{\{a\} : a \in E_1\}$. It is obvious that $\mathcal{L}_1 = Tr(\{E_1\})$.
Step q+1. $(q < m)$ Assume that

$$\mathcal{L}_q = \mathcal{S}_q \cup \{B_1, \ldots, B_{t_q}\},$$

where $B_i \cap E_{q+1} = \emptyset, i = 1, \ldots, t_q$ and $\mathcal{S}_q = \{A \in \mathcal{L}_q : A \cap E_{q+1} \neq \emptyset\}$.

For each i $(i = 1, \ldots, t_q)$ constructs the set $\{B_i \cup \{b\} : b \in E_{q+1}\}$. Denote these sets by $A^i_1, \ldots, A^i_{r_i} (i = 1, \ldots, t_q)$. Let

$$\mathcal{L}_{q+1} = \mathcal{S}_q \cup \{A^i_p : A \in \mathcal{S}_q \Rightarrow A \not\subset A^i_p, 1 \leqslant i \leqslant t_q, 1 \leqslant p \leqslant r_i\}.$$

Theorem 2.3. ([7]) *For every q $(1 \leqslant q \leqslant m)$ $\mathcal{L}_q = Tr(\{E_1, \ldots, E_q\})$, i.e., $\mathcal{L}_m = Tr(\mathcal{H})$.*

We can see that the determinant of $Tr(\mathcal{H})$ based on our algorithm does not depend on the order of $E_1, \ldots, E_m$.

Remark 2.1. ([7]) Let $\mathcal{L}_q = \mathcal{S}_q \cup \{B_1, \ldots, B_{t_q}\}$, and let l_q $(1 \leqslant q \leqslant m-1)$ be the number of elements of $\mathcal{L}_q$. It can verify that the worst-case time complexity of our algorithm is

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

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, at each step of the above algorithm, $\mathcal{L}_q$ is a simple hypergraph. It is known that the size of arbitrary simple hypergraph on U is less than $C_n^{[\frac{n}{2}]}$, where $n = |U|$. $C_n^{[\frac{n}{2}]}$ is asymptotically equal to

$$\sqrt{\frac{2}{\pi n}} 2^n.$$

From this, the worst-case time complexity of the above algorithm is less than exponential in the number of attributes. In cases for which $l_q \leqslant l_m$ $(q = 1, \ldots, m-1)$, it is easy to see that the time complexity of above algorithm is not greater than $\mathcal{O}(|U|^2 \cdot |\mathcal{H}| \cdot |Tr(\mathcal{H})|^2)$. 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 $|U|$.

Proposition 2.4. ([7]) *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.*

Definition 2.1. Let $\mathcal{H}$ be a simple hypergraph on U. We define a family $\mathcal{H}^{-1}$ as follows: a set $A \in \mathcal{H}^{-1}$ iff

(i) $A \in \mathcal{P}(U)$ and $A \not\supseteq B, \forall B \in \mathcal{H}$,

(ii) A is maximal with respect to this property in the sense that all proper supersets C of A contain at least one $B \in \mathcal{H}$.

Note that we have only one case, when $\mathcal{H}^{-1}$ is not a hypergraph. That is, when $\mathcal{H}$ consists of the singletons $\{A_i\}$, where $A_i \in U$.

Hence, it is easy to see that if $\mathcal{H}^{-1}$ is a hypergraph on U, then $\mathcal{H}^{-1}$ is a simple hypergraph.

Proposition 2.5. ([10]) *Let $\mathcal{H}$ be a simple hypergraph on U. Then*

$$\mathcal{H}^{-1} = \overline{Tr(\mathcal{H})}.$$

Remark 2.2. We assume that $U = \{a_1, a_2, \ldots, a_n\}$ $(n > 1)$. Then, let us take a partition $U = X_1 \cup X_2 \cup \cdots \cup X_m \cup W$, where $m = [\frac{n}{3}]$ and $|X_i| = 3$ $(1 \leqslant i \leqslant m)$.

We set

$\mathcal{H} = \{B : |B| = 2, B \subseteq X_i \text{ for some } i\}$ if $|W| = 0$.

$\mathcal{H} = \{B : |B| = 2, B \subseteq X_i \text{ for some } i\ (1 \leqslant i \leqslant m-1) \text{ or } B \subseteq X_m \cup W\}$ if $|W| = 1$.

$\mathcal{H} = \{B : |B| = 2, B \subseteq X_i \text{ for some } i\ (1 \leqslant i \leqslant m) \text{ or } B = W\}$ if $|W| = 2$.

Clearly, $\mathcal{H}$ is a simple hypergraph on U and $n - 1 \leqslant |\mathcal{H}| \leqslant n + 2$.

By Proposition 2.5, we have

$\overline{Tr(\mathcal{H})} = \{A : |A \cap X_i| = 1 \text{ for all i}\}$ if $|W| = 0$.

$\overline{Tr(\mathcal{H})} = \{A : |A \cap X_i| = 1\ (1 \leqslant i \leqslant m-1) \text{ and } |A \cap (X_m \cup W)| = 1\}$ if $|W| = 1$.

$\overline{Tr(\mathcal{H})} = \{A : |A \cap X_i| = 1\ (1 \leqslant i \leqslant m)$ and $|A \cap W| = 1\}$ if $|W| = 2$.

It is easy to see that $|\overline{Tr(\mathcal{H})}| > 3^{[\frac{n}{4}]}$.

Now let $\mathcal{K}$ be a Sperner system on U (i.e. $A, B \in \mathcal{K}$ implies $A \not\subseteq B$). Denote

$$s(\mathcal{K}) = \min\{m : |R| = m, \mathcal{K}_R = \mathcal{K}\}.$$

Theorem 2.6. ([6]) $\sqrt{2|\mathcal{K}^{-1}|} \leqslant s(\mathcal{K}) \leqslant |\mathcal{K}^{-1}| + 1$.

Because a simple hypergraph is also a Sperner system, by Theorem **??** and Proposition , we have the following corollary.

Corollary 2.1. *Let $\mathcal{H}$ be a simple hypergraph on U. Then*

$$\sqrt{2|\overline{Tr(\mathcal{H})}|} \leqslant s(\mathcal{H}) \leqslant |\overline{Tr(\mathcal{H})}| + 1.$$

3. DENSE FAMILIES OF RELATION SCHEME

In this section, we first introduce the notion of dense family of relation scheme $\mathcal{S} = (U, F)$ [11], that is, a collection subsets of U, which by applying certain condition introduces the set F^+. Some results were given in [10, 11], however, some the results shall be proved again and better. Next, we give an application of dense family, which provides an algorithm for finding a BCNF relation R from a given BNCF relation scheme $\mathcal{S}$ such that R is an Armstrong relation of $\mathcal{S}$.

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

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

Proposition 3.1. ([9]) *$F_{\mathcal{D}}$ is an f-family on U.*

In addition, the following proposition is obvious.

Proposition 3.2. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two families of subsets of U. If $\mathcal{D}_1 \subseteq \mathcal{D}_2$ then $F_{\mathcal{D}_2} \subseteq F_{\mathcal{D}_1}$.*

The notion of dense family of a relation scheme is defined in [11] as follows.

Definition 3.1. Let $\mathcal{S} = (U, F)$ be a relation scheme and let $\mathcal{D}$ be a family of subsets of U. We say that a family $\mathcal{D}$ is *$\mathcal{S}$-dense* (or *dense in $\mathcal{S}$*) if $F^+ = F_{\mathcal{D}}$.

We set $\mathcal{L}_{\mathcal{S}} = \{X^+ : X \subseteq U\}$, i.e., $\mathcal{L}_{\mathcal{S}}$ is the set of all closures of $\mathcal{S}$. The problem is how to find dense families. Our next proposition guarantees the existence of at least one dense family, that is $\mathcal{L}_{\mathcal{S}}$, and $\mathcal{L}_{\mathcal{S}}$ is also the greatest $\mathcal{S}$-dense family.

Proposition 3.3. ([11])

(1) *The family $\mathcal{L}_{\mathcal{S}}$ is $\mathcal{S}$-dense.*

(2) *$\mathcal{L}_{\mathcal{S}}$ is the greatest $\mathcal{S}$-dense family.*

The following lemma is obvious.

Lemma 3.1. *Let $\mathcal{D}$ be an $\mathcal{S}$-dense family. Then*

(1) $F^+ = \{X \to Y : (\forall A \in \mathcal{D}) X \subseteq A \Rightarrow Y \subseteq A\}$.

(2) $K \to U \in F^+$ *if and only if* $K \in Trs(\overline{\mathcal{D}} \setminus \{\emptyset\})$.

Next, we prove some the following fundamental results.

Proposition 3.4.

(1) *The family* $\mathcal{M}_{\mathcal{S}} = \mathcal{L}_{\mathcal{S}} \setminus \{U\}$ *is* $\mathcal{S}$*-dense.*

(2) $\mathcal{K}_{\mathcal{S}} = Tr(\overline{\mathcal{D}} \setminus \{\emptyset\})$, *where* $\mathcal{D}$ *is an* $\mathcal{S}$*-dense family.*

Proof.

(1) Suppose $X \to Y \in F^+$. By Proposition 3.2 and Proposition 3.3 (2), we have $X \to Y \in F_{\mathcal{M}_{\mathcal{S}}}$.

Let $X \to Y \in F_{\mathcal{M}_{\mathcal{S}}}$. It is clear that if X is a key of $\mathcal{S}$ then $X \to Y \in F^+$. In case if X is not key of $\mathcal{S}$, then $X^+ \in \mathcal{M}_{\mathcal{S}}$. We then have $X \subseteq X^+$, and hence according to definition of $F_{\mathcal{M}_{\mathcal{S}}}$,

$$F_{\mathcal{M}_{\mathcal{S}}} = \{V \to W : (\forall A \in \mathcal{M}_{\mathcal{S}}) V \subseteq A \Rightarrow W \subseteq A\},$$

we immediately obtain $Y \subseteq X^+$. Consequently, $X \to Y \in F^+$.

(2) Suppose $K \in \mathcal{K}_{\mathcal{S}}$. Then $K \to U \in F^+$. By Lemma 3.1 (2), we have $K \in Trs(\overline{\mathcal{D}} \setminus \{\emptyset\})$. Moreover, it is easy to see that if there exists a $K' \subset K$ such that $K' \in Trs(\overline{\mathcal{D}} \setminus \{\emptyset\})$ then $K' \to U \in F^+$. This contradicts the hypothesis that K is a minimal key of $\mathcal{S}$. Consequently, $K \in Tr(\overline{\mathcal{D}} \setminus \{\emptyset\})$.

Let $K \in Tr(\overline{\mathcal{D}} \setminus \{\emptyset\})$. Also by Lemma 3.1 (2), we get $K \to U \in F^+$. It can be seen that, if there exists a $K' \in \mathcal{K}_{\mathcal{S}}$ such that $K' \subset K$, then $K' \cap \overline{T} \neq \emptyset$ for all $T \in \mathcal{D} \setminus \{U\}$. This contradicts the fact that $K \in Tr(\overline{\mathcal{D}} \setminus \{\emptyset\})$. Hence $K \in \mathcal{K}_{\mathcal{S}}$.

The proposition is proved. ■

Note that Proposition 3.4 (1) guarantees the existence of another dense family. Proposition 3.4 (2) characterizes a set of all minimal keys of relation schemes by the dense families.

Based on Proposition 3.4, we obtain the following result, which was shown in [10] by hypergraphs.

Theorem 3.5. *Let* $\mathcal{S} = (U, F)$ *be a relation scheme. Then*

(1) $\mathcal{K}_{\mathcal{S}} = Tr(\min(\overline{\mathcal{M}_{\mathcal{S}}}))$.

(2) $\mathcal{K}_{\mathcal{S}}^{-1} = \max(\mathcal{M}_{\mathcal{S}})$.

Proof.

(1) By Proposition 2.2, Proposition 3.4 (1), and Proposition 3.4 (2), the proof is straightforward.

(2) Because $\min(\overline{\mathcal{M}_{\mathcal{S}}})$ is a simple hypergraph on U, according to (1) and Proposition 2.1 (1), we obtain

$$Tr(\mathcal{K}_{\mathcal{S}}) = \min(\overline{\mathcal{M}_{\mathcal{S}}}),$$

and hence

$$\overline{Tr(\mathcal{K}_{\mathcal{S}})} = \max(\mathcal{M}_{\mathcal{S}}).$$

On the other hand, by Proposition 2.5 we have

$$\mathcal{K}_{\mathcal{S}}^{-1} = \overline{Tr(\mathcal{K}_{\mathcal{S}})}.$$

which immediately gives

$$\mathcal{K}_{\mathcal{S}}^{-1} = \max(\mathcal{M}_{\mathcal{S}}).$$

The theorem is proved. ■

Theorem 3.5 gives the following application, which is the algorithm for finding a BCNF relation R from a given BCNF relation scheme $\mathcal{S}$ such that R is an Armstrong relation of $\mathcal{S}$. This algorithm is applied to investigate the time complexity of CONSTRUCTING ARMSTRONG RELATION.

Algorithm 3.1. (CAR-BCNF)
Input: Let $\mathcal{S} = (U, F)$ be a BCNF relation scheme.
Output: A BCNF relation R such that R is an Armstrong relation of $\mathcal{S}$.
Method:
Step 1. From $\mathcal{S}$ compute dense family $\mathcal{M}_{\mathcal{S}}$.
Step 2. From the dense family $\mathcal{M}_{\mathcal{S}}$ compute $\max(\mathcal{M}_{\mathcal{S}})$.
Denote by $A_1, \ldots, A_t$ the elements of $\max(\mathcal{M}_{\mathcal{S}})$.
Step 3. Set $\mathcal{Q}_{\mathcal{S}} = \{B : B \neq \emptyset, B = A_i \setminus \{a\}, a \in U, i = 1, 2, \ldots, t\}$. Denote by $B_1, \ldots, B_l$ elements of $\mathcal{Q}_{\mathcal{S}}$.
Step 4. Construct a relation $R = \{h_0, h_1, \ldots, h_l\}$ as follows

$$\text{for all } a \in U, h_0(a) = 0, \forall i = 1, \ldots, l,$$

$$h_i(a) = \begin{cases} 0, & \text{if } a \in B_i, \\ i, & \text{otherwise.} \end{cases}$$

Theorem 3.6.

(1) *The output R of the Algorithm CAR-BCNF is BCNF and an Armstrong relation of $\mathcal{S}$.*

(2) *The time complexity of Algorithm CAR-BCNF is exponential in the size of $\mathcal{S}$.*

Proof. (1) According to Gottlob and Libkin [8], R is BCNF relation.

We set

$$\mathcal{E}_R = \{E_{ij} : 1 \leq i < j \leq l + 1\},$$

where

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

Then we construct the family

$$\max(\mathcal{E}_R) = \{E_{ij} \in \mathcal{E}_R : \forall E_{pq} \in \mathcal{E}_R \Rightarrow E_{pq} \not\supseteq E_{ij}\}.$$

From Algorithm CAR-BCNF and Theorem 3.5 (2), it follows that $\max(\mathcal{E}_R) = \mathcal{K}_{\mathcal{S}}^{-1}$. It is well-known that (Theorem 3.5, [5]) $\mathcal{K}_R^{-1} = \max(\mathcal{E}_R)$. From these and the definition of antikeys, we obtain $\mathcal{K}_R = \mathcal{K}_{\mathcal{S}}$. Moreover, it is known that in BCNF class $\mathcal{K}_R = \mathcal{K}_{\mathcal{S}}$ iff R is an Armstrong relation of $\mathcal{S}$.

(2) Obviously, the family $\mathcal{Q}_{\mathcal{S}}$ and relation R are constructed in polynomial time in the size of $\max(\mathcal{E}_R)$. The family $\max(\mathcal{M}_{\mathcal{S}})$ is also constructed in polynomial time in the size of $\mathcal{M}_{\mathcal{S}}$. Therefore, the time complexity of Algorithm CAR-BCNF is complexity of step 1. By Proposition 2.5, Theorem 3.5 (2) and Algorithm 2.1, we see that the time complexity of Algorithm CAR-BCNF is exponential in the size of $\mathcal{S}$.

The theorem is proved. ■

4. THE COMPLEXITY OF PROBLEM CONSTRUCTING ARMSTRONG RELATION

The following problem plays an important role in the theory of relational database design.

Problem 4.1. *Let $\mathcal{S} = (U, F)$ be a BCNF relation scheme. Construct a BCNF relation R on U such that R is the Armstrong relation of $\mathcal{S}$.*

We prove the following result by hypergraphs and dense families.

Theorem 4.1. *The time complexity of Problem 4.1 is exponential in the size of $\mathcal{S}$.*

Proof. We prove that:

(1) There is an algorithm finding a BCNF relation R from a given BNCF relation scheme $\mathcal{S} = (U, F)$ such that R is an Armstrong relation of $\mathcal{S}$, and the time complexity of this algorithm is exponential in the size of $\mathcal{S}$.

(2) There exists a BCNF relation scheme $\mathcal{S} = (U, F)$ such that the number of rows of any BCNF relation R so that R is an Armstrong relation of $\mathcal{S}$ is exponential in the size of $\mathcal{S}$.

For (1): we have Algorithm CAR-BCNF.

For (2): we consider a simple hypergraph $\mathcal{H}$ as in Remark 2.2. Then, by Corollary 2.1 we have

$$s(\mathcal{H}) \geqslant \sqrt{2|\overline{Tr(\mathcal{H})}|}.$$

Now, we construct a k-relation scheme $\mathcal{S} = (U, F)$, where $F = \{B \to U : B \in \mathcal{H}\}$. It it clear that $\mathcal{S}$ is BCNF and $\overline{Tr(\mathcal{K}_{\mathcal{S}})} = \overline{Tr(\mathcal{H})}$. From this we get

$$s(\mathcal{K}_{\mathcal{S}}) > \sqrt{2}3^{[\frac{n}{8}]}.$$

On the other hand, the number of rows of BCNF relation R which is constructed in Algorithm CAR-BCNF satisfies

$$|R| \leq |U||\overline{Tr(\mathcal{H})}| + 1.$$

Hence, we always can construct a BCNF relation scheme $\mathcal{S}$ such that the number of rows of any BCNF relation R so that R is an Armstrong relation of $\mathcal{S}$ is exponential in the size of $\mathcal{S}$.

The theorem is proved. ■

5. CONCLUSION

In this paper, we have introduced the notion of dense families of relation schemes and investigated some new properties of dense families, and their applications. We have presented an algorithm finding a BCNF relation R from a given BCNF relation scheme $\mathcal{S}$ such that R is an Armstrong relation of $\mathcal{S}$. Based on the obtained, we have shown that in BCNF class the time complexity of problem constructing Armstrong relation is exponential in the size of $\mathcal{S}$.

REFERENCES

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

[2] C. Beeri, M. Dowd, R. Fagin, R. Staman, On the structure of armstrong relations for functional dependencies, *J. ACM* **31** (1984) 30–46.

[3] C. Berge, *Hypergraphs: combinatorics of finite sets*, North - Holland Publishing Co., Amsterdam, 1989 (p. 255).

[4] E. F. Codd, A relational model for large shared data banks, *Communications of the ACM* **13** (1970) 377–387.

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

[6] J. Demetrovics, V. D. Thi, Some observations on the minimal Armstrong relations for normalized relation schemes, *Computers and Artificial Intelligence* **14** (1995) 455–467.

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

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

[9] Järvinen, Dense families and key functions of database relation instances, Freivalds R. (ed.), Fundamentals of Computation Theory, *Proceedings of the 13$^{\text{th}}$ International Symposium, Lecture Notes in Computer Science* **2138**, Springer-Verlag, Heidelberg, 2001 (184–192).

[10] N. H. Son, Investigation on antikeys and minimal keys of relation schemes by hypergraphs, *Annales Univ. Sci. Budapest., Sect. Comp.* **26** (2006) 79–89.

[11] V. D. Thi, N. H. Son, On the dense families in the relational datamodel, *Asean Journal on Science and Technology for Development* **22** (3) (2005) 241–249.

Received on April 27 - 2008