ON THE DENSE FAMILIES IN THE RELATIONAL DATAMODEL

Vu Duc Thi^{*}

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

Nguyen Hoang Son

Nguyen Hoang Son, Department of Mathematics, College of Sciences, Hue University, Vietnam

Riceived 04 November 2005

ABSTRACT

In this paper, dense families of relation schemes are introduced. We characterize minimal keys of relation schemes in terms of dense families. Note that, the dense families of database relations were introduced by Jarvinen [6]. We prove that the set of all minimal keys of a relation scheme s = (U, F) is the transversal hypergraphs of a hypergraph $\overline{D} - \{\emptyset\}$, where *D* is any *s*-dense family. We give a necessary and sufficient condition for an abitrary family to be *s*-dense family. We also present some dense families of relation schemes. Furthermore, in this paper, we also study antikeys by means of dense families. We present connections between antikeys and a dense family of relation schemes. Finally, we study the time complexity of the problem finding antikeys.

1. INTRODUCTION

In this section we briefly present 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, 7, 8].

Let *U* be a finite set of *attributes* (e.g. name, age etc) and $R = \{h_1, ..., h_m\}$ be a relation over *U*. A *functional dependency* (FD for short) over *U* is a statement of form $X \rightarrow Y$, where $X, Y \subseteq U$. The FD $X \rightarrow Y$ holds in a relation *R* if

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

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

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

(F1) $X \rightarrow X \in F$,

- (F2) $(X \rightarrow Y \in F, Y \rightarrow Z \in F) \Longrightarrow (X \rightarrow Z \in F),$
- (F3) $(X \to Y \in F, X \subseteq V, W \subseteq Y) \Longrightarrow (V \to W \in F),$

^{*}Corresponding author e-mail: vdthi@ioit.ac.vn

(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 a *f*-family over U.

Clearly, F_R is a *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 P(U) its power set. The mapping L: $P(U) \rightarrow P(U)$ is called a *closure operation* over U if it satisfies the following conditions:

(1)
$$X \subseteq L(X)$$

- (2) $X \subseteq Y$ implies $L(X) \subseteq L(Y)$,
- (3) L(L(X)) = L(X).

Remark 1.1. It is clear that, if F is a f-family, and we define $L_F(X)$ as

$$L_F(X) = \{a \in U : X \to \{a\} \in F\}$$

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

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

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

Let s = (U, F) be a relation over U and $K \subseteq U$. Then K is a key of s if $K \to U \in F^+$. K is a minimal key of s if K is a key of s and any proper subset of K is not a key of s.

Denote K_s the set of all minimal keys of s. Evidently, K_s is a Sperner system over U (i.e. $A, B \in K_s$ implies $A \not\subset B$).

Let K be a Sperner system over R. We define the set of antikeys of K, denoted by K^{-1} , as follows:

$$K^{-1} = \{A \in \mathbb{P}(\mathbb{U}) : (B \in K) \Longrightarrow (B \not\subseteq A) \text{ and } (A \subset C) \Longrightarrow (\exists B \in K) (B \subseteq C)\}.$$

It is easy to see that K^{-1} is also a Sperner system over U.

2. HYPERGRAPHS AND TRANSVERSALS

Let *U* be a nonempty finite set and put *P*(*U*) for the family of all subsets of *U* (its power set). The family $H = \{E_i : E_i \in \mathbb{P}(U), i = 1, ..., m\}$ is called a *hypergraph* over *U* if $E_i \neq \emptyset$ holds for all *i* (in [3] it is required that the union of E_i s is *R*, in this paper we do not require this).

The elements of *U* are called vertices, and the sets $E_1, ..., E_m$ the edges of the hypergraph *H*. A hypergraph *H* is called *simple* if it satisfies $\forall E_i, E_j \in H: E_i \subseteq E_j \Rightarrow E_i = E_j$. It can be seen that simple hypergraphs are Sperner-systems. Clearly, K_S , K_S^{-1} are simple hypergraphs. Let *H* be a hypergraph over *U*. Then min(H) denotes the set of minimal edges of H with respect to set inclusion, i.e., min(H) = $\{E_i \in H: \exists E_j \in H : E_j \subset E_i\}$, and max(H) denotes the set of maximal edges of H with respect to set inclusion, i.e., max(H) = $\{E_i \in H: \exists E_j \in H : E_j \subset E_i\}$. It is clear that, min(H) and max(H) are simple hypergraphs. Futhermore, min(H) and max(H) are uniquely determined by H. A set $T \subseteq U$ is called a *transversal* of H (sometimes it is called *hitting set*) if it meets all edges of H, i.e., $\forall E \in H: T \cap E \neq \emptyset$. Denote by Trs(H) the family of all transversals of H. A transversal T of H is called *minimal* if no proper subset T' of T is a transversal.

The family of all minimal transversals of *H* called the transversal hypergraph of *H*, and denoted by Tr(H). Clearly, Tr(H) is a simple hypergraph.

Proposition 2.1 [2] Let H and G two simple hypergraphs over U. Then

- (1) H = Tr(G) if and only if G = Tr(H),
- (2) Tr(H) = Tr(G) if and only if H = G,
- $(3) \quad Tr(Tr(H)) = H.$

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

Proposition 2.2 *Let H be a hypergraph over U. Then*

$$Tr(H) = Tr(\min(H)).$$

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

Algorithm 2.3 [5]

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

Output: Tr(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, ..., B_{t_q}\},\$$

where $B_i \cap E_{q+1} = \emptyset$, $i = 1, ..., t_q$ and $S_q = \{A \in L_q: A \cap E_{q+1} \neq \emptyset\}$, for each i $(i = 1, ..., t_q)$ construct the set $\{B_i \cup \{b\}: b \in E_{q+1}\}$. Denote them by $A_1^i, ..., A_n^i$ $(i = 1, ..., t_q)$. Let

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

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

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

Remark 2.5 [5]. Denote $L_q = S_q \cup \{B_1, ..., B_{t_q}\}$, and $l_q (1 \le q \le m-1)$ be the number of elements of L_q . It can be seen that the worst-case time complexity of our algorithm is

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

where $l_0 = t_0 = 1$ and

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

243

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 $2^{n+1/2}/(\prod n)^{1/2}$. 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 $O(|U/^2| H ||Tr(H)|^2)$. Thus, in these cases this algorithm finds Tr(H) in polynomial time in |U|, |H| and |Tr(H)|. Obviously, if the number of elements of H is small, then this algorithm is very effective. It only requires polynomial time in |U|.

The following proposition is obvious

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

Proposition 2.1 is still true for a simple hypergraph.

3. DENSE FAMILIES

In this section, we introduce the notion of dense families of relation scheme s = (U, F). A *s*-dense family is a collection of subsets of *U*, which by applying certain condition introduces the set F^+ . Then we characterize minimal keys of relation schemes in terms of dense families.

Let $D \subseteq P$ (U) be a family of subsets of a finite set U. We define a set F_D over D as follows:

 $F_{\mathsf{D}} = \{ X \to Y : (A \in \mathsf{D}) X \subseteq A \Longrightarrow Y \subseteq A \}.$

Proposition 3.1 F_{D} is an *f*-family over *U*.

Definition 3.2. Let s = (U, F) be a relation scheme and let D be a family of subsets of U. We say that a family D is *s*-dense (or dense in s) if $F^+ = F_D$.

We set $L_s = \{X^+ : X \subseteq U\}$, i.e., L_s is the set of all closures of *s*. The problem is how to find dense families. Our next proposition guarantees the existence of at least one dense family.

Proposition 3.3 *The family* L_s *is s-dense.*

Proof. Assume that $X \to Y \in F^+$. Let $A \in L_s$ such that $X \subseteq A$. This implies that $A \to X \in F^+$. By (F2) we have $A \to Y \in F^+$. From this and $A \in L_s$, we obtain $Y \subseteq A$. Consequently, $X \to Y \in F_{L_s}$.

On the other hand, let $X \to Y \in F_{L_s}$. Because $X^+ \in L_s$ and $X \subseteq X^+$, according to the definition of

 $F_{L_{\rm s}}$, that is,

$$F_{I_{-}} = \{ X \to Y : \forall A \in L_s \} X \subseteq A \Longrightarrow Y \subseteq A \},$$

we obtain $Y \subseteq X^+$. Hence $X^+ \to Y \in F^+$. On the other hand, we have $X \to X^+ \in F^+$. Thus, by (F2) we obtain $X \to Y \in F^+$.

The proposition is proved.

The following lemma is obvious.

Lemma 3.4 If the family D is s-dense then $F^+ = \{X \to Y : (\forall A \in D) X \subseteq A \Rightarrow Y \subseteq A\}.$

Next we show that L_s is the greatest *s*-dense family.

Proposition 3.5 *If* D *is s-dense, then* $D \subseteq L_s$.

Proof. Suppose that $A \in D$. Because $A \to A^+ \in F^+$ and $A \subseteq A$, according to Lemma 3.4 we have $A^+ \subseteq A$. On the other hand, we have $A \subseteq A^+$. Hence $A^+ = A$. This means that $A \in L_s$. The proposition is proved.

Now we give a necessary and sufficient condition for an arbitrary family D to be *s*-dense.

Theorem 3.6 Let s = (U, F) be a relation scheme and let D be a family of subsets of a U. Then D is s-dense iff for every $X \subseteq U$

$$L_{s}(X) = \begin{cases} 4 & A, \text{ if } \exists A \in \mathsf{D} : X \subseteq A, \\ U, \text{ otherwise,} \end{cases}$$

where $L_s(X) = \{a \in U : X \rightarrow \{a\} \in F^+\}.$

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

$$L_{F_{\mathsf{D}}}(X) = \begin{cases} 4 & A, & \text{if } \exists A \in \mathsf{D} : X \subseteq A, \\ U, & otherwise. \end{cases}$$

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

Since $\emptyset \subseteq 4_{A \in D} A \subseteq A$, according to the definition of F_D and L_{F_D} we obtain

$$L_{F_{\mathsf{D}}}(\emptyset) = \underset{A \in \mathsf{D}}{4}A.$$

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

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

$$B = \underset{A \in \mathbf{G}}{4} A.$$

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

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

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

$$L_{F_{\mathsf{D}}}(X) = \frac{4}{A \in \mathsf{D}}(A).$$

By Remark 1.1 it is easy to see that $F^+ = F_D$ holds iff $L_s = L_{FD}$ does.

The theorem is proved.

4. MINIMAL KEYS

In this section, we investigate minimal keys of a relation scheme by means of dense families. We show that generating all minimal keys of a relation scheme s = (U, F) can be reduced to generating all minimal hypergraphs of a hypergraph $\overline{D} - \{\emptyset\}$, where *D* is any *s*-dense family.

Theorem 4.1 Let s = (U, F) be a relation scheme over U and D family of subsets of a U. Then if D is an s-dense, then

$$K_s = Tr(\mathsf{D} - \{\emptyset\}).$$

Proof. Assume *K* is a minimal key of *s*. This implies that $K \to U \in F^+$. Let $T \in D - \{U\}$. By Lemma 3.4, it is easy to see that if $K \subseteq T$ then T = U. This means that, if $T \neq U$ then $K \cap \overline{T} \neq \emptyset$. Hence, $K \in Trs(\overline{D} - \{\emptyset\})$ \$. It is obvious that if there exists a $K' \subset K$ such that $K' \in Trs(\overline{D} - \{\emptyset\})$ then $K' \to U \in F^+$. This contracticts the hypothesis $K \in K_s$. Consequently, *K* is a minimal transversal of $\overline{D} - \{\emptyset\}$.

On the other hand, let K be a minimal transversal of \overline{D} - { \emptyset }. Therefore, we have

$$\forall \mathbf{T} \in \mathbf{D} - \{U\}: \mathbf{K} \cap \overline{T} \neq \emptyset. \tag{1}$$

Note that $T \neq U$. By (1), it is clear that if $K \subseteq T$ then T = U. Thus, according to Lemma 3.4, we have $K \to U \in F^+$. It can be seen that, if there exists a minimal key K' such that $K' \subset K$, then $K' \cap \overline{T} \neq \emptyset$ for all $T \in D$. This contracdicts the fact that $K \in Tr(\overline{D} - \{\emptyset\})$. Consequently, K is a minimal key of s.

The theorem is proved.

Let s = (U, F) be a relation scheme over U. We define the family M_s as follows:

 $M_s = L_s - \{U\}.$

Proposition 4.2 *The family* M_s *is s-dense.*

Proof. Assume that $X \to Y \in F^+$. Let $A \in M_s$ such that $X \subseteq A$. This implies that $A \to X \in F^+$. By (F2), we have $A \to Y \in F^+$. From this and according to the fact that $A \in M_s$, we get $Y \subseteq A$. From definition of F_{M_s} , that is,

$$F_{M_{*}} = \{ V \to W : (\forall A \in M_{s}) \ V \subseteq A \Longrightarrow W \subseteq A \},\$$

it is obvious that $X \to Y \in F_{M_n}$.

On the other hand, let $X \to Y \in F_{M_s}$. Clearly, if X is a key of s then $X \to Y \in F^+$. Suppose that X is not key of s. This implies that $X^+ \in M_s$. Furthermore, $X \subseteq X^+$. Therefore, according to definition of F_{M_s} , we have $Y \subseteq X^+$. Consequently, $X \to Y \in F^+$.

The proposition is proved.

Theorem 4.3 Let s = (U, F) be a relation scheme over U. Then

$$K_s = Tr(\min(M_s)).$$

Proof. According to Proposition 2.2, Proposition 4.2 and Theorem 4.1, the proof is straightforward.

By Proposition 2.2, Proposition 3.3 and Theorem 4.1, the following corollary is also obvious.

Corollary 4.4 *Let* s = (U, F) *be a relation scheme over U. Then*

246

$$K_{\rm s} = Tr(\min(\overline{L_{\rm s}} - \{\emptyset\})).$$

Let s = (U, F) be a relation scheme over U. For every $X \subseteq U$, we set

$$I(X) = \{a \in U : X \to \{a\} \notin F^+\}.$$

Then I(X) is called an *independent set* of s. Denote by I_s the family of all independent sets of s.

Set $\min(I_s) = \{B \in I_s : B \neq \emptyset, \nexists C \in I_s : C \subset B\}$. $\min(I_s)$ is called the *family of all minimal independent sets* of *s*. Clearly, $\min(I_s)$ is a simple hypergraph over *s*.

It can be seen that *X* is a key of *s* if and only if $I(X) = \emptyset$.

Theorem 4.5 Let s = (U, F) be a relation scheme over U. Then

 $K_s = Tr(\min(I_s)).$

Proof. By definition of M_s and min(I_s), we have

$$\min(M_s) = \min(I_s).$$

From this and Theorem 4.3, it is clear that

$$K_s = Tr(\min(I_s)).$$

The theorem is proved.

5. ANTIKEYS

In this section, firstly, we study antikeys by means of dense families. We present connections between antikeys and a dense family of relation schemes.

Proposition 5.1

$$\min(\overline{M_s}) = \max(M_s).$$

Proof. Suppose $A \in \overline{\min(\overline{M_s})}$. Therefore, $\overline{A} \in \min(\overline{M_s})$. This means that

$$\forall B \in \overline{M_s} : B \not\subset \overline{A}$$

or

$$\forall B \in M_{s} : B \not \supset A.$$

Consequently, we obtain $A \in \max(M_s)$.

On the other hand, let $A \in \max(M_s)$. By an argument analogous to the previous one, we get $A \in \overline{\min(\overline{M_s})}$. The proposition is proved.

Remark 5.2. Because min($\overline{M_s}$) is a simple hypergraph over U, according to Proposition 2.1(1) and Theorem 4.3, we obtain

$$Tr(K_s) = \min(M_s).$$

From this and Proposition 5.1, we have

$$\overline{Tr(K_s)} = \max(M_s).$$

The following proposition is known [8].

Proposition 5.3 Let s = (U, F) be a relation scheme over U. Then

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

Theorem 5.4 *Let* s = (U, F) *be a relation scheme over U. Then*

- (1) $K_s^{-1} = \max(M_s).$
- (2) $K_s^{-1} = \overline{\min(I_s)}$.

Proof. (1) The proof is immediate from Remark 5.2 and Proposition 5.3.

(2) Since $\min(I_s)$ is a simple hypergraph over U, according to Proposition 2.1 and Theorem 4.5, we get

$$Tr(K_s) = \min(I_s).$$

Therefore

$$\overline{Tr(K_s)} = \overline{\min(I_s)}.$$

From this and according to Proposition 5.3, we obtain

$$K_s^{-1} = \min(I_s).$$

The theorem is proved.

Note that a set of minimal keys and set of antikeys form a Sperner system. Let K be a Sperner system over U. The time complexity of finding K^{-1} is (in general) exponential in the number of elements of U [7]. However, if we restrict the number of elements of K, then the time complexity of finding K^{-1} is polynomial time.

Proposition 5.5 [8]. Let K be a Sperner system over U. Then

$$K^{-1} = \overline{Tr(\mathbf{K})}$$
.

Lemma 5.6. Let K be a Sperner system over U. If $|K| \le c$ (c is a constant) then K^{-1} is computable in polynomial time.

Proof. Assume that $\kappa = \{K_1, ..., K_c\}$ where $c \le 1$. Certainly, $K_i \ne \emptyset$ for all i = 1, 2, ..., c. We construct the set

$$L = \{\{a_1\} \cup \ldots \cup \{a_k\} : a_i \in K_i, 1 \le i \le c\}.$$

Denote elements of L by L_i , ..., L_t . Clearly, $L_i \in Trs(K)$ for all i = 1, 2, ..., t. Then, we compute

$$\min(\mathbf{L}) = \{L_i \in \mathbf{L} : \nexists \ L_j \in \mathbf{L} : \mathcal{L}_j \subset L_i\}.$$

According to the denifition of transversal hypergraphs, we have

$$Tr(K) = \min(L).$$

By Proposition 5.5, we obtain

$$\kappa^{-1} = \overline{\min(L)}.$$

248

Obviously, $|L| < |U|^c$. Consequently, $\overline{\min(L)}$ is computable in polynomial time.

The lemma is proved.

Algorithm 5.7. (Finding K⁻¹)

Input: let $K = \{K_1, ..., K_c\}$ be a Sperner system over U, where c is a constant.

Output: K⁻¹.

Method:

Step 1. We construct the set

 $L = \{\{a_1\} \cup \ldots \cup \{a_k\} : a_i \in K_i, 1 \le i \le c\}.$

Step 2. Compute

 $\min(\mathbf{L}) = \{L_i \in \mathbf{L} : \nexists \ L_j \in \mathbf{L} : L_j \subset L_i\}.$

Step 3. Let $K^{-1} = \overline{\min(L)}$.

By Lemma 5.6, it is clear that Algorithm 5.7 computes K^{-1} . Futhermore, the time complexity of Algorithm 5.7 is polynomial time in the size of *U*. Obviously, if *c* is small then our algorithm is very effective.

REFERENCES

- 1. Armstrong, W.W. (1974), Dependency structure of database relationship, Information Processing 74, North-Holland Pub. Co., pp. 580-583.
- 2. Berge, C. (1989), Hypergraphs: combinatorics of finite sets, North Holland, Amsterdam.
- 3. Demetrovis, J. (1979), On the equivalence of candidate keys with Sperner systems, Acta Cybernetica vol 4, pp. 247-252.
- 4. Demetrovics, J. and Thi, V.D. (1987), Keys, antikeys and prime attributes, Annales Univ. Sci. Budapest Sect. Comp., vol 8, pp. 35-52.
- 5. Demetrovics, J. and Thi, V.D. (1999), Describing candidate keys by hypergraphs, Computers and Artificial Intelligence, vol 18, pp. 191-207.
- Jarvinen, J. (2001), 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, pp. 184-192.
- 7. Thi, V.D. (1986), Minimal keys and antikeys, Acta Cybernetica, vol 7, pp. 361-371.
- 8. Thi, V.D. and Son, N.H. (2004), Some problems related to keys and the Boyce-Codd normal form, Acta Cybernetica, vol 16, pp. 473-483.