Tạp chí Tin học và Điều khiển học, T.21, S.2 (2005), 130–136

DESCRIBING MINIMAL KEYS BY DENSE FAMILIES OF DATABASE RELATIONS

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

[1]*Institute of Information Technology, VAST*
[2]*Department of Mathematics, College of Sciences, Hue University*

Abstract. The dense families of database relations were introduced by Järvinen [6]. The aim of this paper is to investigate some new properties of dense families of database relations, and their applications. That is, we characterize minimal keys in terms of dense families. 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. The complexity of this algorithm is also esimated.

Tóm tắt. Họ trù mật của quan hệ trong cơ sở dữ liệu được giới thiệu bởi Järvinen [6]. Mục đích của bài báo là nghiên cứu một số tính chất mới của họ trù mật của quan hệ và ứng dụng của nó. Đó là, chúng tôi mô tả khóa tối tiểu của quan hệ thông qua họ trù mật. Chúng tôi chứng tỏ được rằng với một quan hệ R cho trước, tập bằng nhau E_R là một R- trù mật mà kích thước tối đa của nó là $\frac{m(m-1)}{2}$, ở đây m là số các bộ trong R. Chúng tôi cũng chứng tỏ được rằng tập tất cả các khóa tối tiểu của quan hệ R chính là siêu đồ thị transveral của phần bù của tập bằng nhau E_R. Từ đây, chúng tôi đưa ra một thuật toán hiệu quả tìm tất cả các khóa tối tiểu của quan hệ cho trước R. Độ phức tạp của thuật toán này cũng được đánh giá.

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 [1,4,7,8,10].

Let U be a nonempty 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

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

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. 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.

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_is 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., $min(\mathcal{H}) = \{E_i \in \mathcal{H} : \nexists E_j \in \mathcal{H} : E_j \subset E_i\}$. It is clear that, $min(\mathcal{H})$ is a simple hypergraph. Furthermore, $min(\mathcal{H})$ is 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.

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

Proposition 2.1. *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 2.2. [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 $S_q = \{A : A \in L_q \text{ and } A \cap E_{q+1} \neq \emptyset\}$.

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

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

Theorem 2.3. ([5]) *For every q $(1 \leqslant q \leqslant m) L_q = Tr(\{E_1, \ldots, E_q\})$, 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. Denote $L_q = S_q \cup \{B_1, \ldots, B_{t_q}\}$, and l_q ($1 \leqslant q \leqslant m-1$) be the number of elements of L_q. Note that, $l_q \geq t_q$. 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 $l_q \leqslant l_m$ ($q = 1, ..., m-1$), 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 2.4. ([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 2.4 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

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

We have the following proposition.

Proposition 3.1. ([6]) *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 [6], 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. ([6]) *The family $\mathcal{L}_R$ is R-dense.*

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 3.3. ([6]) *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 $\{\overline{A} : A \in \mathcal{D}, A \neq U\}$.*

Note that an element $a \in U$ belongs to all minimal keys if $\overline{A} = \{a\}$ for some $A \in \mathcal{D}$, where $\mathcal{D}$ is an R-dense family. Now we investigate some properties of dense families of database relations, and their applications.

Let U be a nonempty 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.,

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

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

Proposition 3.4. *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 $h_i(X) = h_j(X)$, $1 \leqslant i < j \leqslant m$. 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.

Theorem 3.5. *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 2.1, Proposition 3.4 and Theorem 3.3, 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.,

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

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 3.5, we have the following corollary.

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

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

From Proposition 3.4 and the definition of dense family, the following proposition is obvious.

Proposition 3.7. *Let $R = \{h_1, \ldots, h_m\}$ be a relation over $U = \{a_1, \ldots, a_n\}$. Then $E_R \cup \{U\}$ is R-dense.*

4. FINDING THE SET OF ALL MINIMAL KEYS OF A RELATION

In this section, we present an effective application of Theorem 3.5, which is 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 $R = \{h_1, \ldots, h_m\}$ over U.
Output: K_R.
Method:
Step 1. Construct the equality set

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

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 $N_1, \ldots, N_k$

Step 3. From $\overline{E_R}$ compute the family $min(\overline{E_R}) = \{N_i \in \overline{E_R} : \nexists N_j \in \overline{E_R} : N_j \subset N_i\}$.

Step 4. By Algorithm 2.2 we construct the set $Tr(min(\overline{E_R}))$.

Based on Proposition 2.1, Algorithm 2.2 and Theorem 3.5, 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 2.2. In many cases this algorithm is very effective (see Remark 2).

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

Clearly, if we replace $\overline{E_R}$ by N_R, we have another similar effective algorithm finding all minimal keys of a relation.

5. CONCLUSIONS

In this paper we have investigated dense families of database relations and characterized minimal keys in terms of dense families. We 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 also give an effective algorithm finding all minimal keys of a given relation R.

Our further research will be devoted to the following problems:

1. Let R be a relation over U and $\mathcal{D} \subseteq \mathcal{P}(U)$. What is a necessary and sufficient condition for family $\mathcal{D}$ to be R-dense?

2. Let R be a relation over U. Can we use dense families to sloving the problem of determining a cover of a relation R?

REFERENCES

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

[2] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, North-Holland, Amsterdam , 1989.

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

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

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

[6] J. Järvinen, 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.

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

[8] V. D. Thi, N. H. Son, Describing normal forms for functional dependency by hypergraphs, *Proceeding of the First National Symposium on Fundamental and Applied Information Technology Research (FAIR)*, Hanoi, (2004) 52–60.

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

[10] V. D. Thi, N. H. Son, Some results related to dense families of database relations, *Acta Cybernetica* **17** (1) (2005) 173–182.

Received on December 6, 2004
Revised on April 27, 2005