Tạp chí Tin học và Điều khiển học, T. 14, S. 3 (1998) (18–22)

SOME COMBINATORIAL ALGORITHMS CONNECTING HYPERGRAPHS

VU DUC THI

Abstract. In the relational datamodel the combinatorial algorithms are constructed many authors. The hypergraph is a important concept in the combinatorial theory. The candidate keys play an essential role in the relational datamodel. In this paper, base on hypergraph we present a new combinatorial algorithm that finds all candidate keys of a give relation. Some another results related to the candidate keys are given.

1. INTRODUCTION

Let us give some necessary definitions that are used in the next sections. The concepts give in this section can be found in [1, 2, 3, 4, 7, 9, 10, 15, 16, 17, 18].

Let R be a nonempty finite set and $P(R)$ is power set. The family $H = \{E_i : E_i \in P(R),\ i = 1, \dots m\}$ is called a hypergraph over R if $E_i \neq \emptyset$. (In [4] author requires that the union of E_{i_s} is R. In this paper we do not).

A hypergraph H is simple if $E_i \subset E_j$ implies $i = j$.

The elements of R are called vertices, and the sets $E_1, \dots E_m$ are the edges of the hypergraph H.

It is easy to seen that a simple graph is simple hypergraph with $|E_1| = 2$.

Let $H = \{E_1, \dots E_m\}$ be a hypergraph over R. Set

$$m(H) = \{E_i \in H : \nexists E_j \in H : E_j \subset E_i\}.$$

It can be seen that $m(H)$ is simple hypergraph and the family H uniquely determines the family $m(H)$.

Let H be a hypergraph over R. A set $A \subseteq R$ is called a transversal of H (sometime it is called a hitting set) if $E \in H$ implies $A \cap E \neq \emptyset$.

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

Let $R = \{a_1, \dots, a_n\}$ be a nonempty finite set of attributes. A functional dependency is a statement of the form $A \to B$, where $A,\ B \subseteq R$. The FD $A \to B$ holds in a relation $r = \{h_1, \dots h_m\}$ over R if $\forall h_i,\ h_j \in r$ we have $h_i(a) = h_j(a)$ for all $a \in A$ implies $h_i(b) = h_j(b)$ for all $b \in B$. We also say that r satisfies the FD $A \to B$.

Let F_r be a family of all FDs that hold in r. Then $F = F_r$ satisfiles

(1) $A \to A \in F$,

(2) $(A \to B \in F,\ B \to C \in F) \Rightarrow (A \to C \in F)$,

(3) $(A \to B \in F,\ A \subseteq C,\ D \subseteq B) \Rightarrow (C \to D \in F)$,

(4) $(A \to B \in F,\ C \to D \in F) \Rightarrow (A \cup C \to B \cup D) \in F$.

A family of FDs satisfying (1) - (4) is called an f-family (some times it is called the full family) over R.

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

Given a family F of FDs, there exists a unique minimal f-family F^+ that contains F. It can be seen that F^+ contain all FDs which can be derived from F by the rules (1) - (4).

A relation scheme s is a pair $\langle R, F\rangle$, where R is a set of attributes, and F is a set of FDs over R. Denote $A^+ = \{a : A \to \{a\} \in F^+\}$. A^+ is called the closure of A over s, It is clear that $A \to B \in F^+$ iff $B \subseteq A^+$.

Clearly, if $s = \langle R, F\rangle$ be a relation scheme, then there is a relation r over R such that $F_r = F^+$ (see [1]).

Let r be a relation, $s = \langle R, F\rangle$ be a relation scheme. Then A is a key of r (a key of s) if $A \to B \in F_r$ ($A \to R \in F^+$). A is a candidate key of r (s) if A is a key of r (s) and any proper subset of A is not a key of r (s).

Denote K_r (K_s) the set of all candidate keys of r (s). It can be seen that K_r, K_s are simple hypergraph over R.

Let $I \subseteq P(R)$, $R \in I$, and $A, B \in I \Rightarrow A \cap B \in I$. I is called a meet-semilattice over R. Let $M \subseteq P(R)$. Denote $M^+ = \{\cap M : M \subseteq M\}$. We say that M is generator of I if $M^+ = I$. Note that $R \in M^+$ 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.

Let $s = \langle R, F\rangle$ be a relation scheme and r a relation over R. For every $A \subseteq R$, set $I(A) = \{a \in R : A \to \{a\} \notin F^+\}$. Then $I(A)$ is called the independent set of s. For r, put $I(a) = \{a : A \to \{a\} \notin F_r\}$.

Denote by I_s the family of all independent sets of s. Set $m(s) = \{B \in I_s : B \neq \emptyset, \exists C \in I_s : C \subset B\}$. $m(s)$ is called the family of all independent sets of s. It can be seen that A is a key of s if and only if $I(A) = \emptyset$.

Denote by I_r and $m(r)$ the family of all independent sets and the family of all independent sets of r.

2. RESULTS

In this section, we give a new effective algorithm finding all candidate keys of a give relation. Some another results concerning the candidate keys also are give.

First, we give two following remarks.

Remark 2.1. Let H be a simple hypergraph over R. We define the next family of H, denote H^{-1}, as follows:

$$H^{-1} = \{A \subset R : (B \in H) \Rightarrow (B \notin A) \text{ and } (A \subset C) \Rightarrow (\exists B \in H)\,(B \subseteq C)\}.$$

It is easy to see that H^{-1} is also a simple hypergraph over R.

It can be seen that if H is a simple hypergraph over R, then from the definition of $\mathrm{tr}(H)$ we obtain $H^{-1} = \{R - A : A \in \mathrm{tr}(H)\}$.

Remark 2.2. Let $s = \langle R, F\rangle$ be a relation scheme over R. Set $Z_s = \{A^+ : A \subset R\}$, i.e., Z_s

is the set of all closure of s. Put $T_s = \{A \in Z_s : A \neq R \; \exists B \in Z_s : A \subset B\}$. Thus, T_s is the set of all maximal elements of $Z_s - R$. By the definition of the independent set of s, we can see that $T_s = \{R - B : B \in M(s)\}$.

The following proposition was shown in [11]

Proposition 2.3. *Let $s = \langle R, F\rangle$ be a relation scheme over R. Then*

$$K_s^{-1} = T_s\,.$$

Let r be a relation over R, Let $N_r = \{N_{ij} : 1 \leq i < j \leq |r|\}$, where $N_{ij} = \{a \in R : h_i(a) \neq h_j(a)\}$. Then N_r is led the nonequality set of r.

Let $M_r = \{A \in N_r :$ not exist $B \in N_r : B \subset A\}$. M_r is called the minimal nonequality system of r.

Put $T_r = \{R - A : A \in M_r\}$. T_r is called the maximal equality system of r.

Let r be a relation and H a simple hypergraph over R. We say that r represents H if $K_r = H$.

The following theorem is known ([16])

Theorem 2.4. *Let H be a simple hypergraph and r a relation over R. Then r represents s if $K_r = K_s$.*

If s is a relation scheme over R, and K_s is a set of all candidate keys of s. Then from Theorem 2.4 we obtain

Proposition 2.5. *Let $s = \langle R, F\rangle$ be a relation scheme and r a relation over R. Then r represents s iff $K_s^{-1} = T_r$, where T_r is the maximal equality system of r.*

Base on Remark 2.1, Proposition 2.5 and by the definitions of M_r and T_r we obtain

Theorem 2.6. *Let $s = \langle R, F\rangle$ be a relation scheme and r a relation over R. Then r represents s iff $tr(K_s) = M_r$ where M_r is the minimal nonequality system of r.*

From Theorem 2.6, the next proposition is clear.

Proposition 2.7. *Let $s = \langle R, F\rangle$ be a relation scheme and r a relation over R. Then r represents s iff $K_s = tr(M_r)$, where M_r is the minimal nonequality system of r.*

It is obvious that from the definition of the representation and by Proposition 2.7 we obtain

Proposition 2.8. *Let r be a relation over R. Then $K_r = tr(M_r)$, where M_r is the minimal nonequality system of r.*

Based on Proposition 2.8 and algorithm finding all minimal transversals that presented in [18] we give the algorithm finding all candidate keys of a given relation.

Algorithm 2.9 (Finding all candidate keys).

Input: Let r be a relation over R.

Output: K_r.

Step 1: From r compute N_r.

Step 2: From N_r compute the minimal nonequality system M_r.

Step 3: By algorithm presented in [18], construct $\mathrm{tr}(M_r)$.

Clearly, $\mathrm{tr}(M_r)$ is the set of all candidate keys of r.

It can be seen that M_r is computed in polynomial time in the size of r. Consequently, the time complexity of this algorithm is the time complexity of algorithm, presented in [18], that finds all minimal transversals. In many key this algorithm is very effective (see [18]).

The following example shows that for a give relation r Algorithm 2.9 can be applied to find all candidate keys of a give relation r.

Example 2.10. r is the following is relation over $R = \{a, b, c, d\}$:

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	5

It can be seen that

$$N_r = \{\{a\}, \{d\}, \{a,d\}, \{a,b\}, \{a,c,d\}, \{a,b,d\}, \{a,b,c\}, R\}$$

Clearly, $M_r = \{\{d\}, \{a\}\}$. From this, $K_r = \{\{a,d\}\}$.

An another effective application of algorithm finding all minimal transversals and Proposition 2.8 is finding a relation scheme s from a given relation r such that $F_r = F^+$ in Boyce-Codd normal form (BCNF for short).

We say that a relation scheme $s = \langle R, F\rangle$ (a relation r) is in BCNF if $A \to a \in F^+$ $(A \to a \in F_r)$, and $a \notin A$, then A is a key.

Algorithm 2.11 (Finding a relation scheme s such that $F_r = F^+$).

Input: Let r be a BCNF relation over R.

Output: $s = \langle R, F\rangle$ such that $F^+ = F_r$.

Step 1: By Algorithm 2.9 construct the set of all candidate keys K_r.

Step 2: Denoting elements of K_r by $A_1, \ldots, A_m$, construct a relation scheme, s follows: $s = \langle R, F\rangle$, where $F = \{A_1 \to R, \ldots, A_m \to R\}$.

Because K_r is a simple hypergraph over R, and by our construct we have $K_r \subseteq K_s$. Conversely, if A is a candidate key of s, then there is an A_i such that $A_i \subset A$. Because A_i a key and it is minimal for this property, we obtain $A = A_i$. Hence, $K_r = K_s$ holds. Clearly, s is in BCNF. From this, we have $F^+ = F_r$. It is obvious that the time complexity of this algorithm is the time complexity of Algorithm 2.9.

REFERENCES

[1] Armstrong W. W., *Dependency Structures of Database Relationship*, Information Processing 74, Holland Publ. Co., 1974, pp. 580-583.

[2] Beeri C., Bernstein P. A., Computainonal problems related to the design of normal form relational schema, *ACM Trans on Database Syst.* **4** (1) (1979) 30-59.

[3] Beeri C., Dowd M., Fagin R., Staman R., On the structure of Armstrong relations for functional dependencies, *J. ACM* **31** (1) (1984) 30-46.

[4] Berge C., *Hypergraph: Combinatories of Finite Sets*, North - Holland, Amsterdam, 1989.

[5] Demetrovics J., Thi V. D., Some result about functional dependencies, *Acta Cybernetica* **8** (3) (1988) 273-278.

[6] Demetrovics J., Thi V. D., Relations and minimal keys, *Acta Cybernetica* **8** (3) (1988) 279-285.

[7] Demetrovics J., Thi V. D., On key in the Relations Datamodel, *Inform. Process Cybern. EIK* **24** (10) (1988) 515-519.

[8] Demetrovics J., Thi V. D., Algorithm for generating Armstrong relations and inferring functional dependencies in the relational datamodel, *Computers and Mathematics with Applications*, Great Britain, **26** (4) (1993) 43-45.

[9] Demetrovics J., Thi V. D., Some problems concerning keys for relation scheme and relations in the relational datamodel, *Information Processing Letters*, North Holland, **46** (4) (1993) 179-183.

[10] Demetrovics J., Thi V. D. Some computational problems related to the functional dependency in the relational datamodel, *Acta Acientiarum Mathematicarum* **57** (1-4) (1993) 627-628.

[11] Demetrovics J., Thi V. D., Armstrong Relation Functional Dependencies and Strong Dependencies, *Comput. and AI.* (submitted for publication).

[12] Gare M. R., Johnson D. S., *Computers and Intractability: A Guide to theory of NP-Completeness*, Bell Laboratories, W. H. Freeman and Company, San Francisco, 1979.

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

[14] Lucchesi C. L., Osborn S. L., Candidate keys for relations, *J. Comput. Syst. Sci.* **17** (2) (1978) 270-279.

[15] Thi V. D., *Investigation on Combinatorial Characterizations Related to Functional Dependency in the Relational Datamodel*, MTA-SZTAKI Tanulmanyok, Budapest, 191, 1986, pp. 1-157, Ph. D. Dissertation.

[16] Thi V. D., Minimal keys and antikeys, *Acta Cybernetica* **7** (4) (1986) 361-371.

[17] Demetrovics J., Thi V. D., Some result about normal forms for functional dependency in the relation datamodel, *Discrete Applied Mathematics*, North Holland, **69** (1996) 61-74.

[18] Thi V. D. Some results about hypergraph, *J. Informatics and Cybernetics* **13** (2) (1997) 8-15.

Tóm tắt. Trong mô hình dữ liệu quan hệ, nhiều tác giả đã xây dựng các thuật toán tổ hợp. Siêu đồ thị là một khái niệm quan trọng trong lý thuyết tổ hợp. Các khóa tối tiểu đóng vai trò rất thiết thực trong mô hình dữ liệu này. Trong bài báo này, trên cơ sở khái niệm siêu đồ thị, chúng tôi trình bày một thuật toán tổ hợp mới để tìm toàn bộ các khóa tối tiểu của một quan hệ (file) cho trước. Một số các kết quả liên quan đến các khóa tối tiểu cũng được trình bày trong bài báo này.

Received: January 30, 1998

Institute of Information Technology
National Centre for Sciences and Technology of Vietnam