Advances in Natural Sciences, Vol. 9, No. 1 (2008) (1– 8)

Information Technology

ON THE TIME COMPLEXITY OF THE PROBLEM OF CONSTRUCTING A RELATION SCHEME

Vu Duc Thi
Institute of Information Technology, VAST
Nguyen Hoang Son
Department of Mathematics, College of Sciences, Hue University

Received November 02, 2007

Abstract. The purpose of this paper is to investigate the time complexity of problem of constructing a relation scheme by hypergraphs and dense families. We prove that the time complexity of this problem is exponential in the number of attributes.

Keywords: relation, relational datamodel, functional dependency, relation scheme, minimal key.

1. INTRODUCTION

The relational datamodel introduced by Codd [3] 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 a very important role in logical and structural investigations of the relational datamodel, in both practice and theory. The most important of these constraints is functional dependency (FD for short). Informally, FD means that some attributes' values can be unambiguously reconstructed by the others.

Armstrong relations are objects of interest in relational database theory (see, e.g. [1, 4]). The following problem plays an important role in the theory of relational database design.

Problem 1.1 (Constructing a relation scheme)**.** *Let R relation on U. Construct a relation scheme $S = (U, F)$ such that R is the Armstrong relation of S.*

Hypergraph theory (see, e.g., [2]) is an important subfield of discrete mathematics with many relevant applications in both theoretical and applied computer science.

The dense families of database relations were introduced by Järvinen [6]. Järvinen has characterized FDs and minimal keys of relations in terms of dense families. The method of Järvinen is very effective.

We will prove the following result by means of hypergraphs and dense families.

Theorem 1.1. *The time complexity of constructing a relation scheme is exponential in the number of attributes.*

The rest of the paper is organized as follows: in Section 2, some basic concepts and results of the theory of relational databases are given. In Sections 3 and 4, we first introduce some basic concepts and properties of hypergraphs and dense families. Next, we prove some basic results of hypergraphs and dense families. In Section 5, we prove Theorem 1.1. The final Section is the conclusion.

2. RELATIONAL DATABASES

In this section, we show some key concepts of the theory of relational databases, which can be found in [1, 3].

Let $U = \{a_1, a_{1,} ..., a_n\}$ be a nonempty finite set of *attributes.* A map *dom* associates with each $a_i \in U$ its *domain* $dom(a_i)$. A *relation* Ron U is a subset of Cartesian product $dom(a_1)\times$ $dom(a_2)\times$ ... $\times$ $dom(a_n)$.

We can think of a relation R on U as being a set of tuples: $R = \{h_1, h_{2,} ..., h_m\}$,

$$h_j : U \to \bigcup_{i=1}^{n} dom(a_i), h_j(a_i) \in dom(a_i), j = 1, 2, ..., m.$$

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

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

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

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

R be a relation on U and $K \subseteq U$. And K is called a *minimal key* of R if it satisfies two following conditions:

$$\begin{array}{ll} (K1) & K \to U \in F_R, \\ (K2) & \nexists\, K' \subset K \text{ such that } K' \to U \in F_R. \end{array}$$

The subset K which satisfies only (K1) is called a *key* of R.

Note that a relation may have several minimal keys. Denote K_R the set of all minimal keys of R.

It is clear that $F = F_R$ satisfies

$$\begin{array}{llll} (F1) & X \to X \in F, & & \\ (F2) & (X \to Y \in F, \quad Y \to Z \in F) & \Rightarrow & (X \to Z \in F), \\ (F3) & (X \to Y \in F, \quad X \subseteq V, W \subseteq Y) & \Rightarrow & (V \to W \in F), \\ (F4) & (X \to Y \in F, \quad V \to W \in F) & \Rightarrow & (X \cup V \to Y \cup W \in F), \\ & \forall X,\ Y,\ Z,\ V,\ W \subseteq U. & & \end{array}$$

A family of FDs satisfying (F1) - (F4) is called a *f-family* on U.

F_R clearly 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 an 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 nonempty finite 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 S.

Let $S = (U, F)$ be a relation scheme. Clearly, if $S = (U,\ F)$ 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 S. Evidently, all FDs of S hold in R.

Subset K of U is called 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. K_S denote the set of all minimal keys of S.

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

3. HYPERGRAPHS

In this section, we introduce some basic concepts and results of hypergraphs which will be needed in next sections. The concepts and facts given in this section can be found in [2, 4].

Let U be a nonempty finite set and put $P(U)$ be the family of all subsets of U (its power set). The family $H = \{E_1, E_2, ..., E_m\} \subseteq P(U)$ is called a *hypergraph* on U if $E_i \neq \emptyset$ holds for all i (in [2] it is required that

$$\bigcup_{i=1}^{n} E_i = U$$

but in the present paper this requirement is not necessary.)

The elements of U are called *vertices*, and the sets $E_1, E_2, ..., E_m$ the *edges* of the hypergraph H.

A hypergraph H is called *simple* if it satisfies

$$\forall E_i, E_j \in H : E_i \subseteq E_j \Rightarrow i = j.$$

It can be seen that K_R, K_S are simple hypergraphs.

In this paper we always assume that if a simple hypergraph H plays the role of the set of minimal keys (resp. antikeys, i.e., maximal non-keys), then $H \neq \emptyset$ and $\emptyset \notin H$ (resp. $\emptyset$, $U \notin H$). We consider the 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 H be a hypergraph on U. Then $\min(H)$ denotes the set of minimal edges of H with respect to set inclusion, i.e.,

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

It is clear that, $\min(H)$ is a simple hypergraph. Furthermore, $\min(H)$ is 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.,

$$\forall E \in H : T \cap E \neq \emptyset.$$

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

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

Algorithm 3.1 [5]**.**

Input: Let $H = \{E_1, E_2, ..., E_m\}$ be a hypergraph on U.

Output: $T\mathrm{r}(H)$.

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

$$L_q = S_q \cup \{B_1, B_2, ..., B_{t_q}\},$$

where $B_i \cap E_{q+1} = \emptyset$, $i = 1, 2, ..., t_q$ and $S_q = \{A \in L_q\colon A \cap E_{q+1} \neq \emptyset\}$.

For each i ($i = 1, 2, ..., t_q$) constructs the set $\{B_i \cup \{b\} | \; b \in E_{q+1}\}$. Denote them by $A^i_1, A^i_2, ..., A^i_{r_i} (i = 1, 2, ..., t_q)$. Let

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

Theorem 3.1 [5]. *For every q $(1 \leq q \leq m) L_q = Tr(\{E_1, E_2, ..., E_q\})$, i.e., $L_m = Tr(H)$.*

The determination of $Tr(H)$ based on our algorithm does not depend on the order of $E_1, E_2, ..., E_m$.

Proposition 3.2 [5]**.** *The time complexity of finding $Tr(H)$ of a given hypergraph H is (in general) exponential in the number of elements of U.*

Now we investigate some results about hypergraphs.

Let H be a simple hypergraph on U. We define a set H^{-1} as follows:

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

It is easy to see that if H^{-1} is a hypergraph on U, then H^{-1} is a simple hypergraph.

For each subset A of U, we define $\overline{A} = U \backslash A$. For every family A $\subseteq$ P(U), the *complemente family* of A is $A = \{\overline{A} : A \in A\}$ on U.

We then have following important relationship [8]:

Proposition 3.3. *Let H be a simple hypergraph on U. Then*

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

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

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

Theorem 3.4 ([4])**.**

$$\sqrt{2|K^{-1}|} \leq s(K) \leq |K^{-1}| + 1.$$

Because a simple hypergraph is also a Sperner system, from Theorem 3.4 and Proposition 3.3, we have the following corollary.

Corollary 3.1. *Let H be a simple hypergraph on U. Then*

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

Now we assume that $U = \{a_1,\ a_2,\ ...,\ a_n\}(n > 1)$. Thus we have the following remark.

Remark 3.1. Let us take a partition $U = X_1 \cup X_2 \cup \cdots \cup X_m \cup W$, where $m = [\frac{n}{3}]$ and $|X_i| = 3(1 \leq i \leq m)$.

We set

$H = \{B : |B| = 2,\quad B \subseteq X_i$ for some $i\}$ if $|W| = 0$.
$H = \{B : |B| = 2,\quad B \subseteq X_i$ for some $i : 1 \leq i \leq m-1$ or $B \subseteq X_m \cup W\}$ if $|W| = 1$.
$H = \{B : |B| = 2,\quad B \subseteq X_i$ for some $i : 1 \leq i \leq m$ or $B = W\}$ if $|W| = 2$.

It can be seen that H is a simple hypergraph on U and $n - 1 \leq |H| \leq n + 2$.

By Proposition 3.3, we have

$Tr(H) = \{A : |A \cap X_i| = 1$ for all $i\}$ if $|W| = 0$.
$Tr(H) = \{A : |A \cap X_i| = 1(1 \leq i \leq m-1)$ and $|A \cap (X_m \cup W)| = 1\}$ if $|W| = 1$.
$Tr(H) = \{A : |A \cap X_i| = 1(1 \leq i \leq m)$ and $|A \cap W| = 1\}$ if $|W| = 2$.

Thus, $|\overline{Tr(H)}| < 3^{[\frac{n}{4}]}$.

Set $K = (\overline{Tr(H)})^{-1}$, we obtain

$K = \{C : |C| = n - 3, C \cap X_i = \emptyset$ for some $i\}$ if $|W| = 0$.
$K = \{C : |C| = n - 3, C \cap X_i = \emptyset$ for some $i(1 \leq i \leq m-1)$
or $|C| = n - 4, C \cap (X_m \cup W) = \emptyset\}$ if $|W| = 1$.
$K = \{C : |C| = n - 3, C \cap X_i = \emptyset$ for some $i(1 \leq i \leq m)$ or $|C| = n - 2, C \cap W) = \emptyset\}$
if $|W| = 2$.

It is easy to see that | K | $\leq m + 1$.

4. DENSE FAMILIES

In this section, we introduce some basic concepts and results about dense families of database relations [6, 7].

The notion of dense family of a database relation is defined in [6] as follows:

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

The problem is how to find dense families. Järvinen [5] guarantees the existence of at least one dense family. In the sequel we denote L_{F_R} simply by L_R, i.e.,

$$L_R = \{X_R^+ : X \subseteq U\},$$

where $X_R^+ = \{a \in U : X \to \{a\} \in F_R\}$.
Propositon 4.1 ([6])**.** (1) *The family L_R is R-dense.*

(2) *If $\mathcal{D}$ is R-dense, then $\mathcal{D} \subseteq L_R$.*

In [6] J. Järvinen proved the following important theorem:
Theorem 4.2. *Let R be a relation on U. If $\mathcal{D} \subseteq P(U)$ 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 is minimal with respect to the property of containing an element from each set in $\{\overline{A} : A \in \mathcal{D}, A \neq U\}$.*

Now we investigate some results about dense families. In [7] we also presented another dense family of database relations.

Let $R = \{h_1, h_2, ..., h_m\}$ be a relation on 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 4.3 ([7])**.** *The equality set E_R is R-dense.*

It is easy to see that the dense family E_R has at most $\frac{m(m-1)}{2}$ elements.

In [8] we proved the following important result.
Theorem 4.4. *Let R be a relation on U. Then*

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

We present an effective application of Theorem 4.4, which is the algorithm of finding all minimal keys of a given relation.
Algorithm 4.1.
Input: a relation $R = \{h_1, h_2, ..., h_m\}$ on $U = \{a_1, a_2, ..., a_n\}$.
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 elemens of $\overline{E}_R$ by $N_1, N_2, ..., N_k$.
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 3.1 we construct the set $K_R = \mathrm{Tr}(\min(\overline{E}_R))$.

By Algorithm 3.1 and Theorem 4.4, we have $K_R = \mathrm{Tr}(\min(\overline{E}_R))$. It can be seen that the time complexity of our algorithm is the time complexity of Algorithm 3.1.

From Algorithm 4.1, we have the following application, which is the following algorithm of finding a BCNF relation scheme S from a given relation R in BCNF such that $F^+ = F_R$, i.e., R is an Armstrong relation of S.

Algorithm 4.2.

Input: Let R be a BCNF relation on $U = \{a_1, a_{2,}..., a_n\}$.

Output: A BCNF relation scheme $S = (U, F)$ such that R is an Armstrong relation of S.

Method:

Step 1. By Algorithm 4.1 constructs K_R.

Step 2. Denoting elements of K_R by K_1, K_2, ..., K_m. We construct a relation scheme as follows: $S = (U, F)$, where $F = \{K_1 \to U,\ K_2 \to U, ..., K_m \to U\}$.

Clearly, S is in BCNF and R is an Armstrong relation of S.

Note that in BCNF class a relation R is an Armstrong relation of relation scheme S (i.e. $F_R = F^+$) if and only if $K_R = K_S$.

It can be seen that the time complexity of Algorithm 4.2 is the time complexity of Algorithm 4.1.

5. PROOF OF THEOREM 1.1

The time complexity of *constructing relation scheme* in BCNF class is exponential in the number of attributes. Indeed, we shall prove that:

Claim 1: There is an algorithm of finding a BCNF relation scheme S from a given BCNF relation R such that R is an Armstrong relation of S, and the time complexity of this algorithm is exponential in the number of attributes.

Claim 2: There exists a BCNF relation R such that the number of elements of F of any BCNF relation scheme $S = (U, F)$ so that R is an Armstrong relation of S is exponential in the number of attributes.

For *Claim 1*: We have Algorithm 4.2.

For *Claim 2*: By Remark 3.1 we have $|\mathcal{K}| \leq m + 1$. Set $\mathcal{M} = \{C\backslash\{a\} : C \in \mathcal{K}, a \in U\}$. Denote elements of $\mathcal{M}$ by C_1, ..., C_t. Construct a relation $R = \{h_0,\ h_1,\ ...,\ h_t\}$ as follows:

$$\text{for all } a \in U, h_0(a) = 0$$

$$h_i(a) = \begin{cases} 0 \text{ if } a \in C_i, \\ i \text{ otherwise}, \end{cases} \qquad \forall i = 1, 2, ..., t.$$

It is easy to see that

$$|R| \leq (m+1)|U| + 1.$$

Now we construct a relation scheme $S = (U, F)$ with F $= \{A \to U :\ A \in \overline{Tr(H)}\}$. It is clear that S is in BCNF, $|F| > 3^{[\frac{n}{4}]}$ and R is an Armstrong relation of S.

Hence we can always construct a BCNF relation R in which the number of rows of R is at most $(m + 1)|U| + 1$ but for any BCNF relation scheme $S = (U, F)$ such that R is an Armstrong relation of S, the number of elements of F is exponential in the number of attributes.

Because BCNF relation (resp. relation scheme) class is subset class of relations (resp. relation scheme), hence, from the above proof it is clear that the time complexity of *constructing relation scheme* is exponential in the number of attributes.

The proof is complete. □

6. CONCLUSION

In this paper we investigated the time complexity of *constructing a relation scheme* by hypergrahs and dense families. First, we gave an algorithm which from a given BCNF relation R finds a BCNF relation scheme S such that R is an Armstrong relation of S. Next, we proved that in BCNF class the time complexity of problem which from a given BCNF relation R finds a BCNF relation scheme S such that R is an Armstrong relation of S is exponential in the number of attributes. Because BCNF relation (resp. relation scheme) class is subset class of relation (resp. relation scheme), the time complexity of *constructing a relation scheme* is exponential in the number of attributes.

REFERENCES

1. W.W. Armstrong, Dependency structure of database relationship, in the book: *Information Processing 74*, North-Holland Pub. Co., Amsterdam, 1974, p.580.
2. C. Berge, *Hypergraphs: Combinatorics of Finite Sets,* North-Holland Pub. Co., Amsterdam, 1989.
3. E.F. Codd, *Comm. ACM* **13**, 377 (1970).
4. J. Demetrovics and V. D. Thi, Annales Univ. Sci. Budapest. *Sect. Comp.* **19**, 119 (2000).
5. J. Demetrovics and V.D. Thi, *Computers and Artificial Intelligence* **18**, 191 (1999).
6. J.Järvinen, *Dense families and key functions of database relation instances*, in the book: Freivalds R. (Ed.) *Fundamentals of Computation Theory, Proceedings of the 13th International Symposium*, Lecture Notes in Computer Science **2138,** Springer-Verlag, Heidelberg, 2001, p.184.
7. V.D. Thi and N.H. Son, *Vietnam Journal of Computer Science and Cybernetics* **21** 130 (2005).
8. N.H. Son, Annales Univ. Sci. Budapest., *Sect. Comp.* **26**, 79 (2006).

Corresponding author: *Vu Duc Thi, Institute of Information Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet Rd., Hanoi, Vietnam, E-mail: vdthi@ioip.vast.ac.vn*