Tạp chí Tin học và Điều khiển học, T. 18, S. 1 (2002), 15–21

SOME PROPERTIES OF CHOICE FUNCTIONS

BINA RAMAMURTHY, VU NGHIA, VU DUC THI

Abstract. The family of functional dependencies plays an important role in the relational database. The main goal of this paper is to investigate choice functions. They are equivalent descriptions of family of functional dependencies. In this paper, we give some main properties related to the composition of choice functions.

Tóm tắt. Họ các phụ thuộc hàm đóng vai trò quan trọng trong cơ sở dữ liệu quan hệ. Mục tiêu của chúng tôi là nghiên cứu về các hàm chọn. Các hàm chọn là các mô tả tương đương của họ các phụ thuộc hàm. Trong bài báo này chúng tôi trình bày một số các tính chất cơ bản liên quan đến các hàm chọn.

1. INTRODUCTION

The motivation of this study is equivalent descriptions of family of functional dependencies (FDs). FDs play an significant role in the implementations of relational database model, which was defined by E. F. Codd. Up to now, many kinds of databases have been studied, such as object oriented database, deductive database, distributed database, inconsistent database... For details, see [18], [19], [1], [20] and [17]. However, relational database is still one of the most powerful databases. One of the most important branches in the theory of relational database is that dealing with the design of database schemes. This branch is based on the theory of FDs and constraints. Armstrong observed that FDs give rise to closure operations on the set of attributes. And he shows that closure operation is an equivalent description of family of FDs, that is, the family of all FDs satisfying Armstrong axiom stated in next section. That the family of FDs can be described by closure operations on the attributes'set plays a very important role in theory of relational database. Because this representation was successfully applied to find many properties of FDs, studying those properties of closure operations is indirect way of finding that of the family of FDs. Besides closure operations, there are some other representations of family of FDs. Such as, the closed sets of a closure form a semilattice. And the semilattice with greatest elements give an equivalent description of FDs. The closure operations, and other equivalent descriptions of family of FDs have been studied widely by Armstrong [2], Beeri, Dowd, Fagin and Statman [4], Mannila and Raiha [16].

2. BASIC DEFINITIONS

Let us give some formal definitions that are used in the next sections. Those well-known concepts in relational database given in this section can be found in [2], [3], [4], [8], [10] and [20]. A relational database system of the scheme $R(a_1, \dots, a_n)$ is considered as a table, where columns correspond to the attributes a_i 's while the row are n-tuples of relation r. Let X and Y be nonempty sets of attributes in R. We say that instance r of R satisfies the FD if two tuples agree on the values in attributes X, they must also agree on the values in attributes Y. Here is the formal mathematical definition of FDs.

Definition 2.1. Let $U = \{a_1, \dots, a_n\}$ be a nonempty finite set of attributes. A functional dependency is a statement of the form $A \to B$, where $A, B \subseteq U$. The FD $A \to B$ holds in a relation $R = \{h_1, \dots, h_m\}$ over U 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.

Definition 2.2. Then $F = F_R$ satisfies

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

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

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

(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 over U.

Clearly, F_R is an f-family over U. It is known [2] 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 (1)–(4).

Definition 2.3. A relation scheme s is a pair $\langle U, F\rangle$, where U is a set of attributes, and F is a set of FDs over U.

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 $s = \langle U, F\rangle$ is a relation scheme, then there is a relation R over U such that $F_R = F^+$ (see [2]).

Definition 2.4. Let U be a nonempty finite set of attributes and $P(U)$ its power set. A map $L : P(U) \to P(U)$ is called a closure operation (closure for short) over U if it satisfies the following conditions:

(1) $A \subseteq L(A)$ (Extensiveness Property);

(2) $A \subseteq B$ implies $L(A) \subseteq L(B)$ (Monotonicity Property);

(3) $L(L(A)) = L(A)$ (Closure Property).

Let $s = \langle U, F\rangle$ be a relation scheme. Set $L(A) = \{a : A \to \{a\} \in F^+\}$, we can see that L is a closure over U.

Theorem 2.1. [2] *If F is a f-family and if $L_F = \{a : a \in U$ and $A \to \{a\} \in F\}$, then L_F is a closure. Inversely, if L is a closure, there exists only a f-family F over U such that $L = L_F$, and $F = \{A \to B : A, B \subseteq U, B \subseteq L(A)\}$.*

Let $L \subseteq P(U)$. L is called a meet-irreducible family over U (sometimes it is called a family of members which are not intersection of two other members) if $A, B, C \in L$, then $A = B \cap C$ implies $A = B$ or $A = C$.

Let $I \subseteq P(U)$, $U \in I$, and $A, B \in I \Rightarrow A \cap B \in I$. I is called a meet-semilattice over U. Let $M \subseteq P(U)$.

Denote $M^+ = \{\cap M' : M' \subseteq M\}$. We say that M is a generator of I if $M^+ = I$. Note that $U \in M^+$ but not in M, by convention it is the intersection of the empty collection of sets.

Denote $N = \{A \in I : A \neq \cap\{A' \in I : A \subset A'\}\}$.

In [8] it is proved that N is the unique minimal generator of I.

It can be seen that N is a family of members which are not intersections of two other members.

Let L be a closure operation over U. Denote $Z(L) = \{A : L(A) = A\}$ and $N(L) = \{A \in Z(L) : A \neq \cap\{A' \in Z(L) : A \subset A'\}\}$. $Z(L)$ is called the family of closed sets of L. We say that $N(L)$ is the minimal generator of L.

It is shown [8] that if N is a meet-irreducible family then there is a closure L such that N is the minimal generator of it.

Theorem 2.2. [2] *There is an on-to-one correspondence between meet-irreducible families and f-families on U.*

Theorem 2.3. [8] *There is a 1-1 correspondence between meet-irreducible families and meet- semi-lattices on U.*

Definition 2.5. Let $M \subseteq P(U)$. M is called a Sperner system over U if $A, B \in M$, then A is not a subset of B.

Definition 2.6. Let U be a nonempty finite set of attributes. A family $M = \{(A, \{a\}) : A \subset U, a \in U\}$ is called a maximal family of attributes over R iff the following conditions are satisfied:

(1) $a \notin A$.

(2) For all $(B, \{b\}) \in M$, $a \in B$ and $A \subseteq B$ imply $A = B$.

(3) $\exists (B, \{b\}) \in M : a \notin B$, $a \neq b$, and $L_a \cup B$ is a Sperner system over R, where $L_a = \{A : (A, \{a\}) \in M\}$.

Remark 2.1.

- It is possible that there are $(A, \{a\})$, $(B, \{b\}) \in M$ such that $a \neq b$, but $A = B$.

- It can be seen that by (1) and (2) for each $a \in U$, L_a is a Sperner system over U. It is possible that L_a is an empty Sperner system.

- Let U be a nonempty finite set of attribute and $P(U)$ its power set. According to Definition 2.6 we can see that given a family $Y \subseteq P(U) \times P(U)$ there is a polynomial time algorithm deciding whether Y is a maximal family of attribute over U. Let L be a closure over R. Denote $Z(L) = \{A : L(A) = A\}$ and $M(L) = \{(A, \{A\}) : A \notin A, A \in Z(L)$ and $B \in Z(L), A \subseteq B, A \notin B$ imply $A = B\}$. $Z(L)$ is called the family of closed sets of L. It can be seen that for each $(A, \{a\}) \in M(L)$, A is a maximal closed set which doesn't contain a. It is possible that there are $(A, \{a\})$, $(B, \{b\}) \in M(L)$ such that $a \neq b$, but $A = B$.

The following theorem which shows that closure operations and maximal families of attributes determine each other uniquely.

Theorem 2.4. [13] *Let L be a closure operation over U. Then $M(L)$ is a maximal family of attributes over U. Conversely, if M is a maximal family of attributes over U, then there exists exactly one closure operation L over U so that $M(L) = M$, where for all $B \in P(U)$*

$$H(B) = \begin{cases} \bigcap\limits_{B \subseteq A} A & \text{if } \exists A \in L(M) : B \subseteq A, \\ R & \text{otherwise}, \end{cases}$$

and $L(M) = \{a : (a, \{a\}) \in M\}$.

Now, we introduce the following concept

Definition 2.7. Let $Y \in P(U) \times P(U)$. We say that Y is a minimal family over U if the following conditions are satisfied:

(1) $\forall (A, B), (A', B') \in Y : A \subset B \subseteq U$, $A \subset A'$ implies $B \subset B'$, $A \subset B'$ implies $B \subseteq B'$.

(2) Put $U(Y) = \{B : (A, B) \in Y\}$. For each $B \in U(Y)$ and C such that $C \subset B$ and there is no $B' \in U(Y) : C \subset B' \subset B$, there is an $A \in L(B) : A \subseteq C$, where $L(B) = \{A : (A, B) \in Y\}$.

Remark 2.2.

- $U(U(Y)$.

- From $A \subset B'$ implies $B \subseteq B'$, there is no a $B' \in U(Y)$ such that $A \subset B' \subset B$ and $A = A'$ implies $B = B'$.

- Because $A \subset A'$ implies $B \subset B'$ and $A = A'$ implies $B = B'$, we can be see that $L(B)$ is a Sperner system over R and by (2) $L(B) \neq \emptyset$.

Let I be a meet-semilattice over R. Put $M^*(I) = \{(A, B) : \exists C \in I$ such that $A \subset C$, $A \neq \cap\{C : C \in I, A \subset C\}, B = \cap\{C : C \in I, A \subset C\}\}$. Set $M(I) = \{(A, B) \in M^*(I) :$ there does not exist $(A', B) \in M^*(I)$ such that $A' \subset A\}$.

Theorem 2.5. [13] *Let I be a meet-semilattice over U. Then $M(I)$ is a minimal family over U. Conversely, if Y is a minimal family over U, then there is exactly one meet-semilattice I so that $M(I) = Y$, where $I = \{C \subseteq R : \forall(A, B) \in Y : A \subseteq C$ implies $B \subseteq C\}$.*

Let Z be an intersection semilattice on U and suppose that $H \subset U$, $H \not\subset Z$ hold and $Z \cup \{H\}$ is also closed under intersection. Consider the sets A satisfying $A \in Z$, $H \subset A$. The intersection of all of these sets is in Z therefore it is different form H. Denote it by $L(H)$. $H \subset L(H)$ is obvious. Let $H(Z)$ denote the set of all pairs $(H, L(H))$ where $H \subset U$, $H \notin Z$, but $Z \cup \{H\}$ is closed under intersection. The following theorem characterize the possible sets $H(Z)$:

Theorem 2.6. [7] *The set $\{(A_i, B_i) \,|\, i = 1 \to m\}$ is equal to $H(Z)$ for some intersection semilattice Z iff the following conditions are satisfied:*

$$A_i \subset B_i \subseteq U, \; A_i \neq B_i\,,$$

$$A_i \neq A_j \text{ implies either } B_i \subseteq A_j\,, \text{ or } A_j \subseteq B_i\,,$$

$$A_i \subseteq B_j \text{ implies } B_i \subseteq B_j\,,$$

for any i and $C \subset U$ satisfying $A_i \subset C \subset B_i$ $(A_i \neq C \neq B_i)$ there is a j such that either $C = A_j$ or $A_j \subset C$, $B_j \not\subset C$, $C \not\subset B_j$ all hold.

The set of pair (A_i, B_i) satisfying those condition above is called an extension. Its definition is not really beautiful but it is needed in some application. On the other hand it is also an equivalent notion to the closures:

Theorem 2.7. [7] *$Z \to H(Z)$ is a bijection between the set of intersection semilattices and the set of extensions.*

Definition 2.8. Let U be a nonempty finite set of attributes and $P(U)$ its power set. A map $C : P(U) \to P(U)$ is called a choice function, if every $A \in P(U)$, then $C(A) \subseteq A$.

U is interpreted as a set of alternatives, A as a set of alternatives given to the decision-maker to choose the best and $C(A)$ as a choice of the best alternatives among A.

Let L be a closure operation, we define C and H associated with L as follows:

$$C(A) = U - L(U - A), \qquad (*)$$

and

$$H(A) = A \cap L(U - A). \qquad (**)$$

We can easily prove that $C(A)$ and $H(A)$ are two choice functions. And we name $C(A)$ choice function - I (for short, CF - I), and H(A) choice function - II (for short, CF - II).

Theorem 2.8. *The relationship like $(*)$ is considered as a 1-1 correspondence between closures and choice functions, which satisfies the following two conditions:*
For every $A, B \subseteq U$,
(1) If $C(A) \subseteq B \subseteq A$, then $C(A) = C(B)$ (Out Casting Property),
(2) If $A \subseteq B$, then $C(A) \subseteq C(B)$ (Monotonicity Property).

Theorem 2.9. *The relationship like $(**)$ is considered as a 1-1 correspondence between closures and choice functions, which satisfies the following two conditions:*
For every $A, B \subseteq U$,
(1) If $H(A) \subseteq B \subseteq A$, then $H(A) = H(B)$ (Out Casting Property),
(2) If $A \subseteq B$, then $H(B) \cap A \subseteq H(A)$ (Heredity Property).

We also note that both C and H uniquely determine the closure L as the following

$$L(A(= U - C(U - A) \text{ and } H(A) = A \cup L(U - A).$$

For every $A \subseteq U$, the sets $C(A)$ and $H(A)$ form a partition of A, that is, $C(A) \cup H(A) = A$, and $C(A) \cap H(A) = \emptyset$.

Theorem 2.10. *There is a 1-1 correspondence between CFs - I and closure operations on U.*

Theorem 2.11. *There is a 1-1 correspondence between CFs - II and closure operations on U.*

3. RESULT

First of all, we are giving the formal definition of composition of functions.

Definition 3.1. Let f and g be two functions (e.g closure operations, CFs - I, or CFs - II) on U, and we determine a map T as a composition of f and g the following:

$$T(X) = f(g(X)) = f.g(X) = fg(X) \text{ for every } X \subseteq U.$$

In this section we are going to answer one question: given many CFs - II, what can be said about the composition of those CFs - II. We will soon see that

Theorem 3.1. *Let H_1 and H_2 be CFs - II on U, then composition H_1H_2 and H_2H_1 are a CFs - II on U, and $H_1H_2 = H_2H_1 = H_1 \cap H_2$.*

However, to achieve this results, we necessarily prove those following lemmas and propositions.

First we need to prove the following proposition

Proposition 3.1. *Let H_1 and H_2 be CFs - II on U, then for all $X \subseteq U$, $H_1(X) \cap H_2(X)$ is a CF - II on U.*

To prove $H_1 \cap H_2$ is a CF - II, we need to prove the following.

Lemma 3.1. *Let L_1 and L_2 be closure operations on U, then for all $X \subseteq U$, $L_1(X) \cap L_2(X)$ is a closure operation on U.*

Proof.

Assume L_1 and L_2 be two closure operations on U, then for all $X \subseteq U$, it is easy to obtain that $X \subseteq L_1(X) \cap L_2(X)$ since $X \subseteq L_1(X)$ and $X \subseteq L_2(X)$. Now, to prove the Monotonicity Property of $L_1 \cap L_2$, for every $X \subseteq Y$, we have $L_1(X) \subseteq L_1(Y)$ and $L_2(X) \subseteq L_2(Y)$. Therefore, $L_1(X) \cap L_2(X) \subseteq L_1(Y) \cap L_2(Y)$, so $L_1 \cap L_2$ satisfies Monotonicity Property. Then, we have to prove Closure Property of $L_1 \cap L_2$. We always have $X \subseteq L_1(X) \cap L_2(X) \subseteq L_1(X)$. Using Monotonicity Property of L_1, we attain $L_1(X) \subseteq L_1(L_1(X) \cap L_2(X)) \subseteq L_1(L_1(X)) = L_1(X)$. That means $L_1(X) = L_1(L_1(X) \cap L_2(X))$. Similarly, we attain that $L_2(X) = L_2(L_1(X) \cap L_2(X))$. Therefore, $L_1(X) \cap L_2(X) = L_1(L_1(X) \cap L_2(X)) \cap L_2(L_1(X) \cap L_2(X))$. That is, $L_1 \cap L_2$ satisfies Closure Property, so $L_1 \cap L_2$ is a closure on U. The proof is completed.

Now we are moving on proving Proposition 3.1.

Proof of Proposition 3.1. Assume H_1 and H_2 be CFs - II on U, then for all $X \subseteq U$, we have $H_1(X) = X \cap L_1(U - X)$, and $H_2(X) = X \cap L_2(U - X)$, with L_1 and L_2 two closure operations corresponding to H_1 and H_2 respectively. Thus $H_1(X) \cap H_2(X) = (X \cap L_1(U - X)) \cap (X \cap L_2(U - X)) = X \cap L_1(U - X) \cap L_2(U - X)$. However, due to Lemma 3.1, $L_1(U - X) \cap L_2(U - X)$ is a closure operation, that is, there exists a closure operation L_3 such that $L_3(U - X) = L_1(U - X) \cap L_2(U - X)$. Thus, $C_1(X) \cap C_2(X) = X \cap L_3(U - X) = C_3(X)$, with C_3 is a CF - II corresponding to L_3. The proof is completed.

Before proving Theorem 3.1, we need to prove the follows.

Lemma 3.2. *Let H_1 and H_2 be CFs - II on U, then*

1) $H_1H_2 = H_2H_1H_2$.

2) $H_2H_1 = H_1H_2H_1$.

Proof. Assume H_1 and H_2 be CFs - II on U. Then for all $X \subseteq U$, $H_1(X) = X \cap L_1(U - X)$ and $H_2(X) = X \cap L_2(U - X)$, with L_1 and L_2 two closure operations corresponding to H_1 and H_2 respectively. $H_1H_2(X) = H_1(H_2(X)) = X \cap L_2(U - X) \cap L_1(U - X \cap L_2(U - X)) \subseteq X$. Due to Heredity Property of CFs - II for H_2, we obtain $H_2(X) \cap H_1H_2(X) \subseteq H_2(H_1H_2(X))$. By using $H_1H_2(X) = H_1(H_2(X)) \subseteq H_2(X)$, we attain $H_1H_2(X) \subseteq H_2(H_1H_2(X)) \subseteq H_1H_2(X)$. Hence $H_1H_2(X) = H_2(H_1H_2(X))$, that is, $H_1H_2 = H_2H_1H_2$. Similarly, we obtain $H_2H_1 = H_1H_2H_1$. The proof is completed.

Lemma 3.3. *Let H_1 and H_2 be CFs - II on U, then following is equivalence:*

(1) $H_1 \subseteq H_2$;

(2) $H_1H_2 = H_1$.

Proof.

(1) $\rightarrow$ (2). Assume H_1 and H_2 be CFs - II on U and $H_1 \subseteq H_2$. Since H_1 is a CF - II, H_1 must satisfy Out Casting property: if $H_1(X) \subseteq Y \subseteq X$, then $H_1(X) = H_1(Y)$. Therefore, we have $H_1 \subseteq H_2$ or $H_1(X) \subseteq H_2(X) \subseteq X$ for every $X \subseteq U$, so $H_1(H_2(X)) = H_1(X)$ or we conclude that $H_1H_2 = H_1$.

(2) $\rightarrow$ (1). Assume H_1 and H_2 be CFs - II on U and $H_1H_2 = H_1$. Since H_1 and H_2 are CFs - II, according to definition of choice function, we have $H_1H_2 \subseteq H_2$, but $H_1H_2 = H_1$, so we have $H_1 \subseteq H_2$. The proof is completed.

Easily, we obtain the following Corollary.

Corollary 3.1. *If H is a CF - II on U, then $HH = H$.*

Proof of Theorem 3.1. Assume H_1 and H_2 be CFs - II on U. Then for all $X \subseteq U$, $H_2(X) \subseteq X$. Due to Heredity Property of CF - II for H_1, we obtain $H_1(X) \cap H_2(X) \subseteq H_1(H_2(X))$. Besides that, $H_1(H_2(X)) \subseteq H_2(X) \subseteq X$, we obtain $H_1 \cap H_2(X) \subseteq H_1H_2(X) \subseteq X$. By Proposition 3.1, $H_1(X) \subseteq H_2(X)$ is a CF - II. Using Out Casting Property for $H_1 \cap H_2$, we achieve $H_1 \cap H_2(H_1H_2(X)) = H_1 \cap H_2(X)$ or $H_1(H_1H_2(X)) \cap H_2(H_1H_2(X)) = H_1 \cap H_2(X)$. Due to Corollary 3.1, we obtain $H_1(H_1H_2(X)) = H_1H_2(X)$, and Lemma 3.2, we obtain $H_1H_2(X) = H_2H_1H_2(X)$. Therefore, we obtain that $H_1H_2(X) = H_1 \cap H_2(X)$, that is $H_1H_2 = H_1 \cap H_2$. That means H_1H_2 is a CF - II. Similarly, we obtain $H_2H_1 = H_1 \cap H_2$ and H_2H_1 is a CF - II. The proof is completed.

REFERENCES

[1] Arenas M., Bertossi L., Chomicki J., Consistent Query Answers in Inconsistent Databases, In *Proc. ACM Symposium on Principles of Database Systems* (ACM PODS' 99, Philadelphia), 1999, 68–79.

[2] Armstrong W. W., Dependency Structures of Database Relationships, *Information Processing 74*, Holland Publ. Co., 1974, 580–583.

[3] Beeri C., Bernstein P. A., Computational problems related to the design of normal form relation schemes, *ACM Trans. on Database Syst.* **4** (1) (1979) 30–59.

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

[5] Demetrovics J., Furedi Z., Katona G. O. H., Minimum matrix representation of closure operations, *Discrete Applied Mathematics*, North Holland, **11** (1985) 115–128.

[6] Demetrovics J., Hencsey G., Libkin L., Muchnik I. B., Normal form relation schemes: A new characterization, *Acta Cybernetica*, Hungary, **10** (3) (1992) 141–153.

[7] Demetrovics J., Katona G. O. H., Extremal combinatorial problems of databases, In: MFDBS'87, 1st *Symposium on Mathematical Fundamental of Database Systems*, Dresden, GDR, January, 1987, Lecture Notes in Computer Science, Berlin: Springer 1987, 99–127.

[8] Demetrovics J., Katona G. O. H., A survey of some combinatorial results concerning functional dependencies in database relations, *Annals of Mathematics and Artificial Intelligence* **7** (1993) 63–82.

[9] Demetrovics J., Libkin L., Muchnik I. B. Functional dependencies in relational databases: A lattice point of view, *Discrete Applied Mathematics*, North Holland, **40** (1992) 155–185.

[10] Demetrovics J., Thi V. D., On algorithms for generating Armstrong relations and inferring functional dependencies in the relational datamodel, *Computers and Mathematics with Applications*, Greate Britain, **26** (4) (1993) 43–55.

[11] Demetrovics J., Thi V. D., Some results about normal forms for functional dependencies in the relational data model, *Discrete Applied Mathematics*, North Holland, **69** (1996) 61–74.

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

[13] Demetrovics J., Thi V. D., Family of functional dependencies and its equivalent descriptions, *J. Computer and Math. with Application*, Great Britain, **29** (4) (1995) 101–109.

[14] Demetrovics J., Thi V. D., Some computational problems related to Boyce–Codd normal form, *Annales Univ. Sci. Budapest.*, Sect. Comp., Hungary, **19** (2000) 119–132.

[15] Libkin L., Direct product decomposition of lattices, closures and relation schemes, *Discrete Mathematics*, North Holland, **112** (1993) 119–138.

[16] Mannila H., Raiha K. J., On the complexity of inffering functional dependencies, *Discrete Applied Mathematics*, North Holland, **40** (1992) 237–243.

[17] Ramakrishnan R., Gehrke J., *Database Management Systems*, McGraw - Hill, 2000.

[18] Schewe K.-D., Thalheim B., Fundamental concepts of object oriented databases, *Acta Cybernetica*, Hungary, **11** (1-2) (1993) 49–83.

[19] Thayse A., *Modal Logic to Deductive Databases*, John Wiley & Son, 1989.

[20] Ullman J., *Principles of Database and Knowledge Base Systems*, Vol 1, Computer Science Press, 1988.

Received May 3, 2001

Bina Ramamurthy, Vu Nghia,
State University of New York at Buffalo.
Vu Duc Thi, Institute of Information Technology.