Tạp chí Tin học và Điều khiển học, T. 17, S. 1 (2001), 35-39

SOME RERULTS ABOUT CHOICE FUNCTIONS

VU DUC NGHIA

Abstract. The family of functional dependencies (FDs) is an important concept in the relational database. The choice function is the equivalent description of the family of FDs. This paper gives some results about choice functions. Some properties of choice functions, such as comparison between and composition of two choice functions, are investigated.

Tóm tắt. Họ các phụ thuộc hàm là một khái niệm quan trọng trong cơ sở dữ liệu quan hệ. Bài này đưa ra khái niệm hàm chọn là một sự mô tả tương đương của họ các phụ thuộc hàm và trình bày một số kết quả nghiên cứu về hàm chọn.

1. INTRODUCTION

The relational datamodel which was introduced by E. F. Codd is one of the most powerful database models. The basic concept of this model is the relation, which is a table that every row of which corresponds to a record and every column to an attribute. Because the structure of this model is clear and simple, and mathematical instruments can be applied in it, it becomes the theoretical basis of database models. Semantic constraints among sets of attributes play very important roles in logical and structural investigations of relational data model both in practice and design theory. The most important among these constraints is the family of FDs. Equivalent descriptions of the family of FDs have been widely studied. Based on the equivalent descriptions, we can obtain many important properties of the family of FDs. Choice function is one of many equivalent descriptions of the family of FDs. In this paper we investigate the choice functions. We show some properties of choice functions, which concerntrate much on the comparison between and composite of two choice functions.

Let us give some necessary definitions that are used in the next section. The concepts given in this section can be found in [1-8, 11, 12].

Definition 1.1. Let $U = \{a_1, \dots, a_n\}$ be a nonempty finite set of attributes. A functional dependency (FD) 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 $A \to B$.

Definition 1.2. Let F_R be a family of all FDs that hold in R. 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) $(A \to B \in F,\ A \subseteq C,\ D \subseteq B) \Rightarrow (C \to D \in F)$,

(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 (sometimes it is called the full 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 (1) - (4).

Definition 1.3. 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 $A \to B \in F^+$ if $B \subseteq A^+$.

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

Definition 1.4. Let U be aq nonempty finite set of attributes and $P(U)$ its power set. A map $L : P(U) \to P(U)$ is called a cosure over U if it satisfies the following conditions:
(1) $A \subseteq L(A)$,
(2) $A \subseteq B$ implies $L(A) \subseteq L(B)$,
(3) $L(L(A)) = L(A)$.

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 1.1. *If F is a f-family and if $L_F(A) = \{a : a \in U$ 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 $F = \{A \to B : A, B \subseteq U,\ B \subseteq L(A)\}$.*

So we can conclude that there is a 1–1 correspondence between closures and f-families on U.

Definition 1.5. 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 $C(A) \subseteq A$.

If we assume that $C(A) = U - L(U - A)$ $(*)$, we can easily see that C is a choice function.

Theorem 1.2. *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 $C(A) = C(B)$,
(2) If $A \subseteq B$, then $C(A) \subseteq C(B)$.

We call all of choice functions satisfying those two above conditions special choice functions.

From Theorems 1.1 and 1.2, we have the following important result.

Theorem 1.3. *There is a 1–1 correspondence between special choice functions and f-families on U.*

We define Γ as a set of all of special choice (SC) functions on U. Now we investigate some properties of those functions.

2. RESULTS

First of all we give the definition of a composite function of two SC functions.

Definition 2.1. Let $f, g \in \Gamma$, and we determine a map k as a composite function of f and g as the following:

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

Definition 2.2. Let U be a nonempty set finite set of attributes, and $f, g \in \Gamma$. We say that f is smaller than g, denoted as $f \leq g$ or $g \geq f$, if for every $X \subseteq U$ we always have $f(X) \subseteq g(X)$.

The "smaller" relation, $\leq$, satisfies these following properties. For every $f, g, h \in \Gamma$:
1) $f = f$ (Reflexive),
2) If $f \leq g$, and $g \leq f$, then $g = f$ (Symmetric),
3) If $f \leq g$, and $g \leq h$, then $f \leq h$ (Transitive).

Proposition 2.1. *If $f, g \in \Gamma$, then*

(1) $fg \leq f$
(2) $fg \leq g$
(3) $gf \leq f$
(4) $gf \leq g$.

Proof. Since $f, g \in \Gamma$, f and g must be SC functions on U. Therefore, we have $g(X) \subseteq X$ for every $X \subseteq U$, then $f(g(X)) \subseteq f(X)$ (1). And f is a SC function on U, so $f(g(XX)) \subseteq g(X)$ (2). So we can conclude that $fg \leq f$ and $fg \leq g$.

Similarly, we can easily prove (3) and (4).

Proposition 2.2. *If $f, h, g \in \Gamma$, and $f \leq g$, then*

(1) $fh \leq gh$
(2) $hf \leq hg$.

Proof. Because f, g and h are three SC functions and $f \leq g$, we always have $f(f(X)) \subseteq g(h(XX)$, for every $X \subseteq U$. Since $f \leq g$, we have $f(X) \subseteq g(X)$. h is a SC function, so we have $h(f(X)) \subseteq h(g(X))$. We can conclude that $fh \leq gh$ and $hf \leq hg$.

Proposition 2.3. *If $f, g, h, k \in \Gamma$, and $f \leq g$, $k \leq h$, then $fk \leq gh$.*

Proof. Assume $f, g, h, k \in \Gamma$, and $f \leq g$, $k \leq h$. According to Proposition 2.2, we have $fk \leq gk$ and $gk \leq gh$. Therefore, according to the transitive property, we have $fk \leq gh$.

Theorem 2.1. *If $f, g \in \Gamma$, then these following two conditions are equivalence:*

(1) $f \leq g$;
(2) $fg = f$.

Proof.

(1) $\Rightarrow$ (2). Assume $f, g \in \Gamma$ and $f \leq g$. Since f is a SC function, f must satisfies this property: if $f(X) \subseteq Y \subseteq X$, then $f(X) = f(Y)$. Therefore, we have $f \leq g$ or $f(X) \subseteq g(X) \subseteq X$ for every $X \subseteq U$, so $f(g(X)) = f(X)$ or we conclude that $fg = f$.

(2) $\Rightarrow$ (1). Assume $f, g \in \Gamma$ and $fg = f$. Since f and g are SC function, according to Proposition 2.1, we have $fg \leq g$, but $fg = f$, so we have $f \leq g$. The proof is completed.

From Theorem 2.1, we can easily see that if $f \leq g$, then fg is a SC function (since $fg = f$, and f is a SC function).

We also can generalize Theorem 2.1 as the following:

Let $f_1, \ldots, f_n$ be SC functions and $f_1 = \min\{f_1, \ldots, f_n\}$ (That is, f_1 is samllest among $f_1, \ldots, f_n$. That means $f_1 \leq f_i$, for all $i = 1, \ldots, n$).

Then $f_1 f_{i2} f_{i3} \ldots f_{in} = f_1$, and $\{f_{i2}, f_{i3}, \ldots, f_{in}\}$ is a permutation of $\{f_2, f_3, \ldots, f_n\}$.

This statement can be proved easily by induction method. (Key: $f_1.f_i = f_1$ whenever $f_1 \leq f_i$, for $i = 1, \ldots, n$).

Lemma 2.1. *If $f \in \Gamma$, then $ff = f$.*

Proof. We have $f \in \Gamma$, so f is a SC function. Besides that, we always have $f = f$, so according to Theorem 2.1, we have $ff = f$.

Theorem 2.2. *Let $f, g \in \Gamma$. A composite function of f and g, denoted as fg, is a SC function if and only if $fgf = fg$:*

$$(fg \text{ is a SC function} \Leftrightarrow fgf = fg)$$

Proof. First of all we need to prove that fg is a choice function.

For every $X \subseteq U$, we have $g(X) \subseteq X$ because g is a SC function. And f also is a SC function, so if $g(X) \subseteq X$, then $f(g(XX)) \subseteq f(X) \subseteq X$. Therefore, we can conclude that $fg(X) \subseteq X$, in other word, we can say that fg is a choice function. similarly, we can prove that gf is also a choice function.

Now, we must prove that fg is a SC function $\Leftrightarrow fgf = fg$.

First, we need to prove the statement: if fg is a SC function, then $fgf = fg$. According to Proposition 2.1, we have $fg \leq f$. And fg is a SC function, so $fgf = fg$ due to Theorem 2.1.

Then, we just need to prove that if $fgf = fg$, then fg is a SC function. In other words, we need to prove that if $fgf = fg$, then fg satisfies these following two conditions:

1) If $X \subseteq Y$, then $fg(X) \subseteq fg(Y)$.

2) If $fg(X) \subseteq Y \subseteq X$, then $fg(X) = fg(Y)$.

We prove that 1) is true. When $X \subseteq Y$, we have $g(X) \subseteq g(Y)$ since g is a SC function. And when $g(X) \subseteq g(Y)$, we have $f(g(X)) \subseteq f(g(Y))$ or $fg(X) \subseteq fg(Y)$ since f is also a SC function. So we can conclude that 1) is true.

After that, we move to prove that 2) is true. We have $fg(X) \subseteq Y \subseteq X$, so $g(fg(X)) \subseteq g(Y) \subseteq g(X)$ or $gfg(X) \subseteq g(Y) \subseteq g(X)$ since g is a SC function. And since f is also a SC function, we also have $f(gfg(X)) \subseteq f(g(Y)) \subseteq f(g(X))$ or $fgfg(X) \subseteq fg(Y) \subseteq fg(X)$. We can rewite that expression as $fgg(X) \subseteq fg(Y) \subseteq fg(X)$. Therefore, $fg(X) = fg(Y)$.

Consequencely, we can conclude that fg is a SC function iff $fgf = fg$. The proof is completed.

Theorem 2.3. *Let $f, g \in \Gamma$. Then fg and gf are simultaneously SC functions if and only if $fg = gf$.*

Proof. In the proof of Theorem 2.2, already we have proved that fg and gf are always choice functions when f and g are SC functions.

Now, we need to prove this statement: if fg and gf are simultaneously SC functions, then $fg = gf$, for $f, g \in \Gamma$.

According to Proposition 2.1, we have $fg \leq g$ and $fg \leq f$. So due to Proposition 2.3, we have $(fg)(fg) \leq gf$. But we also have fg is a SC function, so $(fg)(fg) = fg$ due to Lemma 2.1. Thus, $(fg)(fg) = fg \leq gf$. Similarly, we also have $gf \leq fg$. Hence, we have $fg \leq gf \leq fg$, so we can conclude that $fg = gf$.

Then, we just need to prove that: if $fg = gf$, then fg and gf are simultaneously SC functions for $f, g \in \Gamma$. In other words, we need to prove that if $fg = gf$, then fg and gf satisfies these following two conditions:

1) If $X \subseteq Y$, then $fg(X) \subseteq fg(Y)$ and $gf(X) \subseteq gf(Y)$.

2) If $fg(X) \subseteq Y \subseteq X$, then $fg(X) = fg(Y)$, and if $gf(X) \subseteq Y \subseteq X$, then $gf(X) = gf(Y)$.

We prove that 1) is true. In the proof of Theorem 2.2, we have already proved 1): if $X \subseteq Y$, then $fg(X) \subseteq fg(Y)$. Similarly, we also can prove that $gf(X) \subseteq gf(Y)$.

After that, we move to prove 2) is true. We have $fg(X) \subseteq Y \subseteq X$, so $g(fg(X)) \subseteq g(Y) \subseteq g(X)$ or $gfg(X) \subseteq g(Y) \subseteq g(X)$ since g is a SC function. And since f is also a SC function, we also have $f(gfg(X)) \subseteq f(g(Y)) \subseteq f(g(X))$ or $fgfg(X) \subseteq fg(Y) \subseteq fg(X)$. We can rewite that expression as $ffgg(X) \subseteq fg(Y) \subseteq ffgg(X) = fg(X) \subseteq fg(Y) \subseteq fg(X)$. Therefore, $fg(X) = fg(Y)$.

Similarly, we also prove that if $gf(X) \subseteq Y \subseteq X$, then $gf(X) = gf(Y)$.

Consequencely, we can say that fg ans gf are simultaneously SC functions if and only if $fg = gf$ for $f, g \in \Gamma$. The proof is completed.

So far, we have covered some properties of the composition of two SC functions and found out some very interesting results. At the end of this article, we would like to raise the following two questions:

1) Can we generalize Theorem 2.2 for the composition of n SC functions? And will we get the same answer? More generally, what is a necessary and sufficient condition such that a composite function of n SC functions is a SC function?

2) Is the union, intersection, or subtraction of two SC functions a SC function?

REFERENCES

[1] Armstrong W. W., *Dependency structures of Database Relationships*, Information ProceSCing 74, Holland Publ. Co., 1994, 580-583.

[2] 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.

[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] Demotrovics J., Katona G. O. H., A survey of some combinatorial results concerning functional dependencies in database relations, *Anals of Mathematics and Arficial Intelligence* **7** (1993) 63-82.

[5] Demetrovics J., Libkin L., Muchnik I. B., Functional dependencies and the semilattice of closed classes, *Proceedings of MFDBS 87, Lecture Notes in Computer Science*, 1987, 136-147.

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

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

[8] 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.

[9] Demetrovics J., Thi V. D., Armstrong relation, functional dependencies and strong dependencies, *Computers and AI* **14** (3) (1995) 279-298.

[10] Demetrovics J., Thi V. D., Some computational problems related to the functional dependencies in the relational datamodel, *Acta Scientiarum Mathematicarum* **57** (1993) 627-638.

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

Received August 30, 2000
Revised September 10, 2000

University of Buffalo