TẠP CHÍ TIN HỌC VÀ ĐIỀU KHIỂN HỌC - Tập I (1991), Số 4 (24-28)

VỀ CÁC PHÉP MÃ HÓA HỢP LÝ TRONG TIẾN TRÌNH THIẾT KẾ MÔ HÌNH CƠ SỞ DỮ LIỆU

HUỲNH HỮU NGHĨA

Scitech. tp. Hồ Chí Minh

Xem $R = < R^+, F >$, với R^+ là tập tất cả các thuộc tính và F tập các ràng buộc phụ thuộc hàm. Ta thường gặp các bài toán (các thủ tục) sau:

1. Tìm tất cả các tập chìa khóa (key attribute set) của R.
2. Tìm một cơ sở tối thiểu (unredundant cover) của F.
3. Phân rã R thành 1 lược đồ CSDL R_i, $R_i = < Ri^+, Fi >$ ít ra cũng phải đạt các tiêu chuẩn
   - Mọi f thuộc F_i đều là các phụ thuộc hàm nội bộ.
   - Các thuộc tính không phải là thuộc tính của chìa khóa (key attribute) đều phải phụ thuộc vào một chìa khóa bất kỳ (dạng chuẩn 2 Boyce–Code).

Cách làm thông thường là gán cho mỗi thuộc tính một mã số và tiến hành thao tác trên các con số đó. Chúng tôi xem xét các phương pháp gán từng mã số cho các tập thuộc tính thích hợp để làm giảm đi số "thuộc tính", khoảng lưu trữ dữ liệu. Chúng tôi cổ động việc thực hiện các giải thuật truyền thống trên máy vi tính mà độ phức tạp của chúng thường từ $|R^+|^2|F|^2$ trở lên (giải thuật tìm tất cả các tập chìa khóa).

Chúng tôi sẽ trình bày một giải thuật mã hóa có độ phức tạp thấp ($|R^+|^2 + |R^+||F|$) và phát sinh ít mã số nhất.

I - MỘT SỐ ĐỊNH NGHĨA

1. Quan hệ tổng quát R gồm tập các thuộc tính R^+ và tập các ràng buộc phụ thuộc hàm F. Ta viết $R = < R^+, F >$.

2. Xem hai tập $A, B \subseteq R^+$. Nếu B phụ thuộc hàm vào A ta ký hiệu $A \to B \in F^{**}$, với F^{**} là bao đóng của F suy từ hệ luật dẫn Amstrong.

3. Cho hai tập $A, B \subseteq R^+$: ta ghi $A \vee B$ là hội của A và B, $A \wedge B$ là giao của hai tập này.

4. Xem $A \subseteq R^+$. Tập hợp: $Cl_F(A) = \{b \in R^+ : \exists A' \subseteq A : A' \to b \in F^{**}\}$ được gọi là bao đóng của tập A ứng với tập phụ thuộc hàm F.

5. Cho $A, B \subseteq R^+$ và $A \to B \in F^{**}$. Ta nói $A \to B$ là một phụ thuộc hàm nguyên tố nếu và chỉ nếu mọi $A' \subseteq A$, nếu $A' \to B \in F^{**}$ thì $A = A'$.

6. Xem tập $K \subseteq R^+$. K được gọi là một chìa khóa của quan hệ $R = ; < R^+, F >$ nếu $K \to R^+$ nguyên tố trong F^{**}.

7. Một phụ thuộc hàm $A \to B \in F^{**}$ được gọi là phụ thuộc hàm nội bộ của một quan hệ $< T_i^+, F_i >$ nếu $A \vee B \subseteq T_i^+$ và A là một chìa khóa của $< T_i^+, F_i >$.

II - CÁCH MÃ HÓA HỢP LÝ (TRÊN TẬP CÁC THUỘC TÍNH CÁCH BIỆT NHAU)

Xem quan hệ tổng quát $R = < R^+, F >$. Ta gọi một phân hoạch (partition) $M(R^+)$ trên tập R^+ là một cách mã hóa hợp lý trên quan hệ R nếu $M(R^+)$ thỏa các điều kiện sau:

1. Từng vế trái và từng vế phải của từng phụ thuộc hàm thuộc F là hội cách biệt của một số tập trong $M(R^+)$.
2. Tập hợp các thuộc tính không thuộc về vế nào của F là hội cách biệt của một số tập trong $M(R^+)$.
3. Hai tập $T_i, T_j \in M(R^+)$, nếu $T_i \wedge T_j \neq 0$ thì $T_i = T_j$.

Ta dự định gán cho mỗi tập trong $M(R^+)$ một con số (duy nhất) 1 đến m. Như vậy ta biến đổi tập F thành tập $M(F)$: tập "các phụ thuộc hàm với các vế gồm các con số".

Nếu một tập hợp $A \in M(R^+)$, A sẽ được gọi là một tập mã hóa được theo phép M. Ta ghi $M(A)$ là tập các mã số của các tập con thành phần của A.

III – MỘT SỐ TÍNH CHẤT CỦA MỘT PHÉP MÃ HÓA HỢP LÝ M

Ta xem một cách mã hóa hợp lý $M(R^+) = \{T_1, ..., T_n\}$.

(III.1) *Mệnh đề:* Với $0 \leq m \leq n$. Đặt T^m là hội của m tập hợp nào đó thuộc $M(R^+)$. Họ $\{T^m\}$ sẽ đóng kín với các phép tính giao, hội, hiệu các tập hợp.

(III.2) *Mệnh đề:* A và B là hai tập thuộc tính mã hóa được (theo M). $A = B$ nếu và chỉ nếu $M(A) = M(B)$.

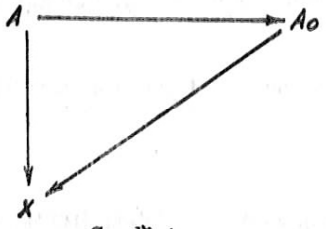
(III.3) *Định lý:* Cho $< M(R^+), M(F) >$ là kết quả mã hóa của một phép mã hóa M áp dụng trên quan hệ $< R^+, F >$. Xem $A \to \{x\}$ là một phụ thuộc hàm không tầm thường nào đó. Thế thì:

(1) Tồn tại một tập $A_0 \subseteq A$, A_0 mã hóa được, $A_0 \to \{x\} \in F^{**}$. Ta biểu diễn bằng sơ đồ 1.

Ghi chú: Ta nói $A \to x$ là kết quả bắc cầu giữa $A \to A_0$ (tầm thường) và $A_0 \to x$ với A_0 mã hóa được.

(2) Tồn tại một tập B mã hóa được sao cho A và B cách biệt và tồn tại một tập $A_0 \subseteq A$, A_0 mã hóa được và A_0 xác định hàm B. Ta biểu diễn bằng sơ đồ 2.

(Ta nói một phụ thuộc hàm bất kỳ $A \to x$ không tầm thường có thể được xấp xỉ bằng một phụ thuộc hàm $A_0 \to B$ với Ao và B là hai tập mã hóa được).



Sơ đồ 1

Sơ đồ 2

(III.4) *Định lý:* Xem tập $K \subseteq R^+$. T là một tập mã hóa được. Xem $T_1 \subset T$, $T_1 \neq T$. Ta có $Cl[(K \setminus T) \vee T_1] = Cl(K \setminus T) \vee T_1$ (ta thấy các tập "lỡ cỡ" không có giá trị gì trong phép tính bao đóng).

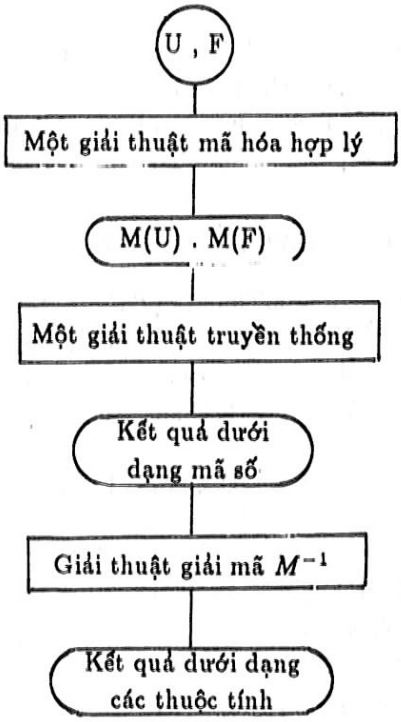
Hệ quả: Xem tập $K \subseteq R^+$. Gọi K_1 là hội tất cả các tập con mã hóa được, chứa trong K. Đặt $K_2 = K \setminus K_1$. Ta có: $Cl_F(K) = Cl_F(K_1) \vee K_2$.

(III.5) *Mệnh đề:* K là một tập mã hóa được thì $Cl_F(K)$ là một tập mã hóa được.

(III.6) *Mệnh đề:* A là một tập mã hóa được. Thế thì $Cl_{M(F)}M(A) = M[Cl_F(A)]$ (một phép mã hóa hợp lý bảo toàn phép tính bao đóng).

IV - THỰC HIỆN CÁC GIẢI THUẬT TRUYỀN THỐNG CÓ DÙNG THÊM CÁCH MÃ HÓA HỢP LÝ

Chúng ta thử xem một tiến trình sau:



Bằng các kết quả toán học trong mục (III) chúng tôi chứng minh được rằng kết quả cuối cùng không thay đổi so với cách làm thông thường là gán cho mỗi thuộc tính một mã số khi giải các bài toán truyền thống nói trên. Nói cụ thể hơn là chúng tôi đã áp dụng có kết quả trên các giải thuật sau;

- Tìm tất cả các chìa khóa của R, Giải thuật Lucchessi - Osborn (1978) theo tài liệu [H THUAN].
- Giải thuật tìm một cơ sở tối thiểu [BerBer, 1979].
- Giải thuật quan niệm lược đồ CSDL nhất quán và đầy đủ BTHUY-86 [BTHUY-86].

Hơn nữa khi tiến hành nghiên cứu áp dụng trên giải thuật BTHUY-86 chúng tôi chứng minh được rằng việc biểu diễn các phụ thuộc hàm "có nguy cơ gây mâu thuẫn dữ liệu" sẽ được thực hiện dễ dàng và trọn vẹn nhờ các vế của chúng đều là các tập mã hóa được, xem [NGHIA].

V - GIẢI THUẬT MÃ HÓA HỢP LÝ TẠO RA ÍT MÃ SỐ NHẤT

Chúng tôi xin giới thiệu một giải thuật đã được chứng minh là một phép mã hóa hợp lý và tạo ra ít mã số nhất (xem mục D.II.1 và D.II.2 trong [NGHIA].

Trong giai đoạn mã hóa, chúng ta cần đến số thuộc tính, số phụ thuộc hàm được nhập vào và lập các sơ yếu lý lịch của chúng bằng các đặc tả

1) var tsốtht = integer; tsốfth = integer;

Để dễ theo dõi, các cấu trúc lưu trữ sẽ chứa tối đa 24 thuộc tính và 100 phụ thuộc hàm

2) type con_trỏ_tht = ^tht;

3) var tht = record {ứng từng bước thuộc tính}
tên = char;
vịtrí = array [1...200] of integer;
{thứ tự của vế, trong cấu trúc FTH, có chứa thuộc tính này}

4) var quan_hệ = array [1...24] of con_trỏ_tht

5) type vế = set of [A...Z]
tập con = set of [1...24]

6) var tập_fth = array [1...200] of vế;
mã_fth = array [1...200] of tập_con;

Cấu trúc "tập_fth" dành 100 thành tố đầu để chứa toàn các vế trái và 100 thành số sau để chứa toàn các vế phải. Cấu trúc "mã_fth" để chứa kết quả mã hóa của tập phụ thuộc hàm.

7) var CHKHOA = array [1...24] of vế.

Cấu trúc CHKHOA dùng để giải mã (biến một tập gồm các mã số thành tập các thuộc tính).

TÓM LƯỢC GIẢI THUẬT MÃ HÓA NGH

(bước 1) *So chế dữ liệu nhập*

(1.1) - Sáp nhập các phụ thuộc hàm có cùng vế trái lại để tạo ra một phụ thuộc hàm mới có vế phải bằng hội các vế phải cũ.
- Loại bỏ các thuộc tính ở vế phải của một phụ thuộc hàm, đã có mặt ở vế trái. Ví dụ: biến đổi $ABC \dashrightarrow ADC$ thành $ABC \rightarrow D$.

(1.2) - Đếm số phụ thuộc hàm và số thuộc tính nhập vào.
- Ghi lại thuộc tính nào ở vế nào của phụ thuộc hàm nào và nhờ vậy biết được có tất cả bao nhiêu vế chứa thuộc tính đó.

(bước 2) *Sắp thứ tự array các con trỏ quan hệ để chuẩn bị xử lý các thuộc tính theo chiều giảm dần tính bởi tổng số vế có chứa thuộc tính đó.*

(bước 3) *Lặp đi lặp lại đến hết các thuộc tính thuộc ít nhất 1 vế.*

(3.1) Xem thuộc tính x.

(3.2) Xét từng thuộc tính y có cùng tổng số vế với x. Nếu x và y cùng có mặt trong các vế:

(a) Đưa x và y vào chung một tập CHKHOA [i] nào đó và tập này mang mã số là i.

(b) Che thuộc tính y lại để lần sau không xét nữa.

(c) Ghi mã số i vào cấu trúc mã phụ thuộc hàm. Cụ thể là nếu x thuộc các vế $v_1, ..., v_m$ thì mã số i được ghi vào các thành tố thứ $v_1, ..., v_m$ của array mã_fth.

(bước 4) *Gom tất cả các thuộc tính không thuộc vế nào vào chung tập CHKHOA [24].*

(bước 5) *Trả về các vị trí mã_fth[] và CHKHOA[].*

Nhận xét:

(1) Ở bước 3 ta thao tác các thuộc tính theo cách tuần tự trên một hàng đợi:

var QUEUE = arrat [1...24] of boolean

- chỉ xử lý những thành tố có trị 1
- những thuộc tính đã giải quyết xong (đã bị che) thì thành tố tương ứng sẽ có trị 0.

(2) Độ phức tạp của giải thuật trên là $|U|^2 + |U|.|F|$, trong đó:
- $|U|$ là các số thuộc tính được nhập vào
- $|F|$ là độ dài lưu trữ các phụ thuộc hàm.
- $|U|^2$ là độ dài phức tạp (tối đa) của một thuật toán sắp thứ tự (xấu nhất) được sử dụng đến.

Xin dành ít dòng cuối cùng để tỏ lòng biết ơn Trung tâm Điện toán trường ĐHBK thành phố HCM đã giúp đỡ đăng bài tóm lược tiểu luận chưa được công bố của tôi. Các tài liệu có liên quan có thể liên hệ với tác giả để sao lại.

Nhận ngày 1 - 11 - 1991

TÀI LIỆU THAM KHẢO

1. [BTHUY-86]: Bích Thủy, Lương Đổng Thi, Une approache de conception d'une base de données cohérentes et complete. Luận văn tiến sĩ số 314 khoa KH KTXH trường đại học Genève, 1986.
2. [ULLMAN]: J. D. Ullman, Principles of database systems, 2nd edition, 1982.
3. [NGHIA]: H. H. Nghĩa, Các phép mã hóa hợp lý, tiểu luận chưa công bố, 1989.
4. [BerBer 79]: Beeri, C và Berstein, P. A., Computationnal problems related to the design of normal forms for relational schemas, ACM Transactions on DB systems, T. 4, No. 1, 1979.
5. [HTHUAN], Some results about keys of relation schemas, Acta Cybernetica VII/I, Viện MT & TĐH hàn lâm viện HUNGARY, 1985.

ABSTRACT

On logical coding methods in process of designing of relational scheme

Let $R = < R^+, F >$ be a universal in which R^+ is the set of attributes of R, and F is the set of functional dependencies (fds) holding in R. In process of designing of relational scheme, we usually want to pass rapidly, as fast as possible, the procedures as follows:
- To find all key attribute sets of R.
- To find an unredundant cover of F.
- To decompose R into a Database (DB) scheme $\{R_i = < R_i^+, Fi >, i = 1...n\}$ which satifies at least 2 conditions:
  1/ each f of F_i is an embodied fd,
  2/ each R_i satifies Boyce Code Norm 2, i.e. every nonkey attribute depends functionally on a key of R_i.

At the first stage, the coding stage, one normally assign to each attribute one number, then manipulate on those numbers. Now, we study some method (tricky or logical ?) to assign to each "selected" disjoint set a number (a code number). So, we hope not only to reduce the "total attributes" but also to get around with condition of "as long as a century", even that of "Memory Overflowed", that is almost indispendable when we are in the process on a microcomputer. For the complexity, on a whole, is a multiple of total attributes, most often.

Also, we present our coding algorithm, the NGH coding algorithm. It not only has low complexity but also produces the least amount of code number – all over logical coding methods.