VIỆN KHOA HỌC VIỆT NAM

tạp chí KHOA HỌC TÍNH TOÁN và ĐIỀU KHIỂN

Tập II — Số 3

HÀ NỘI 1986

Tạp chí

KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN

XUẤT BẢN VÀO CÁC THÁNG 3, 6, 9, 12.

Trụ sở: 70 Trần Hưng Đạo, Hà Nội. Dây nói: 52825

VÔ CÙNG THƯƠNG TIẾC GIÁO SƯ TẠ QUANG BỬU
(1910 – 1986)

NGÀY 21-8-1986, Giáo sư Tạ Quang Bửu kính yêu đã qua đời! Cùng với nhân dân và giới Khoa học kỹ thuật trong cả nước, những người làm công tác Khoa học tính toán và điều khiển xin bày tỏ lòng tiếc thương vô hạn đối với một nhà khoa học lớn của đất nước, một người Thầy, một người Anh gần gũi thân thiết của mình.

Là một nhà hoạt động chính trị, xã hội và quân sự lỗi lạc có nhiều cống hiến to lớn đối với sự nghiệp cách mạng của Tổ quốc ta trong hơn bốn chục năm qua, Giáo sư Tạ Quang Bửu đồng thời cũng là người có công gây dựng nền khoa học kỹ thuật và nền giáo dục đại học hiện đại của đất nước ta.

Ngành khoa học tính toán và điều khiển của nước ta còn rất non trẻ, Giáo sư Tạ Quang Bửu là người đã gây dựng và chăm sóc nó từ những bước đầu, và luôn theo dõi, động viên và khuyến khích trong mỗi bước phát triển về sau của nó. Đầu những năm 60, trên cương vị Phó chủ nhiệm kiêm Tổng thư ký Ủy ban Khoa học nhà nước, Giáo sư Tạ Quang Bửu đã thành lập những nhóm cán bộ khoa học và kỹ thuật đầu tiên về toán học tính toán và kỹ thuật máy tính, đã chỉ đạo việc trang bị máy tính điện tử đầu tiên của nước ta. Giáo sư Tạ Quang Bửu luôn dành sự quan tâm đặc biệt đối với việc đào tạo đội ngũ cán bộ giảng dạy và nghiên cứu khoa học trong lĩnh lực tính toán và điều khiển. Với tư cách là một người nghiên cứu khoa học, Giáo sư Tạ Quang Bửu đã liên tục trong hơn hai mươi năm qua tham dự đều đặn các sinh hoạt khoa học quan trọng của ngành khoa học tính toán và điều khiển, trình bày nhiều bài giảng sâu sắc về những vấn đề cơ bản của ngành khoa học kỹ thuật hiện đại này. Ảnh hưởng của Giáo sư Tạ Quang Bửu về tư tưởng khoa học, về phong cách học tập và nghiên cứu, đối với anh chị em cán bộ ngành khoa học tính toán và điều khiển là vô cùng to lớn và quí giá.

Anh Tạ Quang Bửu vô cùng kính mến!

Vĩnh biệt Anh, người Thầy và người Anh lớn vô vàn thân thiết của chúng tôi, những người làm công tác trong lĩnh vực khoa học tính toán và điều khiển xin nguyện luôn cố gắng noi gương Anh, học tập Anh để trở thành những cán bộ khoa học kỹ thuật giỏi, những con người trung thực và chân chính, có những đóng góp hữu ích đối với sự nghiệp xây dựng đất nước thân yêu của chúng ta

TẠP CHÍ

KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN

VỀ MỘT THUẬT TOÁN CÀI ĐẶT CẢI BIÊN NHÓM PHÉP TOÁN CHIẾU – CHỌN – NỐI

NGUYỄN THANH THỦY

TRONG [1] một vài thuật toán cài đặt hữu hiệu phép toán mở rộng chiếu–chọn–nối và đánh giá độ phức tạp của chúng đã được đưa ra. Trong bài này sẽ trình bày một vài cải biên của chúng kèm theo các đánh giá cụ thể độ phức tạp thời gian và bộ nhớ. Các khái niệm mở đầu có thể xem trong [1, 4].

Thuật toán T.L.I.

Vào: Quan hệ R có tập thuộc tính W; $A \subseteq W$.

Quan hệ S có tập thuộc tính V; $B \subseteq V$; $C \subseteq W \cup V$.

E, F tương ứng là biểu thức logic trên R, S.

Ra: $RESULT = (\sigma_E(R)\ [A = B]\ \sigma_F(S))\ [C]$

Phương pháp

```
   |* qui ước: có chỉ dẫn lên các thuộc tính A, B tương ứng của R, S*|.
begin xác định P = R [A] ∩ S [B]
   |* có thể lấy P = R [A] hoặc P = S [B] *|
   for each e ∈ P do
      begin truy nhập tới R để xác định M = {r ∈ R | r [A] = e} nhờ chỉ dẫn;
        if not eof (M) then
           begin for each r ∈ M do if r thỏa E then
                                          add r[C ∩ W] t° M1;
              if not eof (M1) then
                 begin truy nhập tới S để xác định
                     N = { s ∈ S | s [B ] = e} nhờ chỉ dẫn;
                    if not eof(N) then
                       for each s ∈ N do
                          if s thỏa F then
                               begin s1: = s [C ∩ V];
                                     for each r1 ∈ M1 do
                                        begin   t : = (r1, s1)
                                                add t to RESULT
                                        end
                               end
                 end
           end
      end
end
```

Mệnh đề 1. Độ phức tạp thời gian của thuật toán là:

$$HS \,.\, \sum_j card(R) \,.\, s(A = e_j) + \sum_j card(R) \,.\, s(A = e_j) \,.\, \tau(E) + HS \,.\, s(E) \,.\, \sum_j card(S) \,.\, s(A = e_j) +$$

$$+ s(E) \sum_j card(S) \,.\, s(B = e_j)\ \tau(F) + (HS + \eta)\, card(R) \,.\, card(S) \,.\, s(E) \,.\, s(F) \,.\, s(A = B) +$$

$$+ \gamma\Big(\sum_j card(S) \,,\, s(F) \cdot s(B = e_j) + \sum_j card(R)\ s(E) \,.\, s(A = e_j)\Big).$$

Thuật toán đòi hỏi

$k_1 . card(R) + k_2 . card(S) + (k_1 + k_2) . card(R) . card(S) . s(E) . s(F) . s(A = B)$ đơn vị nhớ ngoài.

$\max_j k_1 . card(R) . s(A = e_j) + \max_j k_1 . card(R) . s(A = e_j) . s(E) +$

$+ s(E) . \max_j k_2 . card(S) . s(B = e_j)$ đơn vị nhớ trong.

Ở đây k_1, k_2 lần lượt là độ dài mỗi bản ghi của quan hệ R, S, HS là tham số chỉ sự cân đố giữa các xử lý trong thiết bị trung tâm so với các xử lý vào ra hay so với các trao đổi dữ liệu giữa bộ nhớ trong và bộ nhớ ngoài.

η là chi phí kết nối hai bộ r_1, s_1.

γ là giá phải trả để chiếu lấy một số thuộc tính trong một bộ.

$\tau(E)$, $\tau(F)$ tương ứng là thời gian trung bình để xác định một bộ $r \in R$ cho trước thỏa E hay một bộ $s \in S$ cho trước thỏa F. [2].

Chứng minh: Truy nhập tới R để xác định M đòi hỏi

$HS . \sum_j card(R) . s(A = e_j)$ đơn vị thời gian

và $\max_j k_1 . card(R) . s(A = e_j)$ đơn vị nhớ trong.

Kiểm tra các bộ trong M thỏa E chi phí

$\sum_j card(R) . s(A = e_j) . \tau(E)$ đơn vị thời gian,

$\max_j k_1 . card(R) . s(A = e_j) . s(E)$ đơn vị nhớ trong.

Để truy nhập tới S, xác định N phải mất

$s(E) . HS . \sum_j card(S) . s(B = e_j)$ đơn vị thời gian,

$s(E) . \max_j k_2 . card(S) . s(B = e_j)$ đơn vị nhớ trong.

Thời gian kiểm tra các bộ $s \in N$ thỏa F là

$s(E) . \sum_j card(S) . s(B = e_j) . \tau(F)$.

Kết nối các bộ r1, s1 và cho vào RESULT mất

$(HS + \eta) card(R) . card(S) . s(E) . s(F) . s(A = B)$ đơn vị thời gian,

$(k_1 + k_2) card(R) . card(S) . s(E) s(F) . s(A = B)$ đơn vị nhớ ngoài.

Chi phí để chiếu các bộ là

$\gamma (\sum_j card(R) . s(E) . s(A = e_j) + \sum_j card(S) . s(F) . s(B = e_j))$.

Tổng cộng chi phí bộ nhớ trong là

$\max_j k_1 . card(R) . s(A = e_j) + \max_j k_1 . card(R) . s(A = e_j) . s(E) + s(E) . \max_j k_2 . card(S) . s(B = e_j)$.

Chi phí bộ nhớ ngoài là

$k_1 . card(R) + k_2 . card(S) + (k_1 + k_2) . card(R) . card(S) . s(E) . s(F) . s(A = B)$.

Chi phí thời gian là

$$HS . \sum_j card(R) . s(A = e_j) + \sum_j card(R) . s(A = e_j) . \tau(E) + s(E) . HS . \sum_j card(S) . s(B = e_j) +$$

$$+ \sum_j s(E) . card(S) . s(B = e_j) . \tau(F) + (HS + \eta) . card(R) . card(S) s(E) . s(F) . s(A = B)$$

$$+ \gamma (\sum_j card(R) . s(E) . s(A = e_j) + \sum_j card(S) . s(F) . s(B = e_j)),$$ đ. p. c. m.

Nhận xét 1: Đặt

$\alpha_1 = \sum_j s(A = e_j)$, $\beta_1 = \sum_j s(B = e_j)$,

$\gamma_1 = \max_j s(A = e_j)$, $\delta_1 = \max_j s(B = e_j)$,

hằng $= card(R) . card(S) s(E) . s(F) . s(A = B)$,

constant $=$ hằng $. (k_1 + k_2) + k_1 . card(R) + k_2 . card(S)$,

$L = (HS + \eta) \,.\, \text{hằng} + \Upsilon\,(card\,(R)\,.\,s\,(E)\,.\,\alpha_1 + card\,(S)\,.\,s\,(F)\,.\,\beta_1).$

Độ phức tạp thời gian (TC), độ phức tạp bộ nhớ (MC) của thuật toán T.L.1 là

$TC1 = \alpha_1\,.\,card\,(R)\,(HS + \tau\,(E)) + s\,(E)\,.\,\beta_1\,.\,card\,(S)\;(HS + \tau\,(F)) + L$

$MC1 = constant + k_1\,.\,\Upsilon_1\,.\,card\,(R)\;(1 + s\,(E)) + s(E)\,.\,k_2\,.\,card\,(S)\,\delta_1$

Độ phức tạp của thuật toán T.L.1 [1] là

$$(1)\quad \begin{cases} TC2 = TC1 \\ MC2 = constant + k_1\,.\,\Upsilon_1\,.\,card\,(R) + s\,(E)\,.\,k_2\,.\,card\;(S)\,.\,\delta_1\;(1 + S\,(F)). \end{cases}$$

Nếu thay đổi vai trò R, S (tương ứng E và F, A và B, k_1 và k_2,...) trong thuật toán T.L.1 [1], T.L.1 thì ta sẽ nhận được các thuật toán cải biên tương ứng.
với độ phức tạp TC3, MC3
TC4, MC4.

$$(2)\quad TC3 = TC4 = \beta_1\,.\,card\;(S)\,(HS + \tau\;(F)) + s\,(F)\,.\,\alpha_1\,.\,card\;(R)\,.\;(HS + \tau\;(E) + L,$$

$$(3)\quad MC4 = constant + k_2\,.\,\delta_1\,.\,card\;(S)\;(1 + s\;(F)) + s\,(F)\,.\,k_1\,.\,card\;(R)\,.\,\Upsilon_1,$$

$$(4)\quad MC3 = constant + k_2\,.\,\delta_1\,.\,card\;(S) + s\;(F)\,.\,k_1\,.\,card\;(R)\,.\Upsilon_1.\;(1 + s\,(E)),$$

$$(5)\quad TC1 \geqslant TC3 \Leftrightarrow \frac{\alpha_1\,.\,card\;(R)\;(HS + \tau\;(E))}{(1 - s\;(E)} > \frac{\beta_1\,.\,card\;(S)\,.\;(HS + \tau\;(F))}{(1 - s\;(F))}$$

Nhận xét 2:

Hoàn toàn tương tự, nếu trong các thuật toán T.L.2, T.L.3 [1] thay đổi vai trò R và S thì lại nhận được những thuật toán cải biên của chúng. Tùy theo yêu cầu tiết kiệm thời gian hay bộ nhớ mà lựa chọn thuật toán thích hợp.

Nhận xét 3:

Trong thuật toán T.L.1, T.L.3 [1], T.L.1 ta đòi hỏi các tệp phải có các chỉ dẫn lên các thuộc tính chọn và các chỉ dẫn lên các thuộc tính nối. Khi yêu cầu này không được thỏa mãn thì có thể tạo lập chỉ dẫn mới, tất nhiên khi đó cần phải tính đến chi phí để làm công việc này.

Nhận ngày 3-6-1985

TÀI LIỆU THAM KHẢO

[1] **Nguyễn Thanh Thủy**, Một vài thuật toán cài đặt tốt nhóm các phép toán chọn—chiếu—nối tự nhiên và đánh giá độ phức tạp thời gian và bộ nhớ của chúng. Tìm hiểu hệ cơ sở dữ liệu. Viện Khoa học thống kê, Hà Nội, 1985, 59—68.

[2] **Nguyễn Thanh Thủy**, Một thuật toán cài đặt tối ưu đối với phép chọn và phép tìm kiếm. Tìm hiểu hệ cơ sở dữ liệu. Viện Khoa học thống kê, Hà Nội, 1985, 42—46.

[3] **Blasgen M.W., Eswaran K. P.** [1977], Storage and access in relational data bases, IBM systems Journal, Vol 16, N°4, 1977.

[4] **Adiba M., Delobel C.** [1982], Bases de donneés et systèmes relationnels, Dunod, Paris, 1982.

[5] **Beck L. L.** [1979], Ageneralixed implementation method for relational data sublangunges. IEEE Trans. on Software enzeneering, vol SE-6, N°2, march, 1979.

[6] **Griffiths P. S. et al.** [1979], Access path selection in a relational manegement systems, Proc. ACM SIGMOD conf. 1979.

ABSTRACT

Some modified implementation algorithms for the generalized operation PROJECTION—SELECTION—JOIN

In [1] some effective implementation algorithms for the generalized opperation SELECTION—PROJECTION—JOIN and their memory and time cost estimations have been shown. In this paper, will be given some their modified implementation algorithms and their time, memory complexities.

TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN, TẬP II 1986), Số 3 (5-8)

QUAN HỆ GIỮA KHÓA CỦA HÀM ĐÓNG VÀ KHÓA CỦA LƯỢC ĐỒ QUAN HỆ

LÊ VĂN BÀO

Đặt vấn đề:

Trong việc nghiên cứu cơ sở dữ liệu mô hình quan hệ, một vấn đề quan trọng được đặt ra là xây dựng những thuật toán hữu hiệu để xác định khóa của lược đồ. Trong bài này sử dụng công cụ là hàm đóng để nghiên cứu một số tính chất khóa của lược đồ quan hệ. Trong phần đầu chứng minh tính đồng nhất giữa tập khóa của lược đồ quan hệ và khóa của hàm đóng. Sau đó nghiên cứu một số tính chất của hàm đóng, cuối cùng đưa ra biểu thức giao các khóa của hàm đóng.

1. Hàm đóng.

Cho $U = \{a_1, a_2, \ldots a_n\}$ tập n – thành phần, 2^U là tập các tập con của U. Hàm f $2^U \to 2^U$ gọi là hàm đóng khi và chỉ khi với mọi X, $Y \subseteq U$

a) $X \subseteq f(X)$

b) $f(f(X)) = f(X)$

c) $Y \subseteq X$ thì $f(Y) \subseteq f(X)$.

Cho f là hàm đóng trên U. $K \in 2^U$, K gọi là siêu khóa của f nếu $f(K) = U$. K được gọi là khóa nếu K là siêu khóa bé nhất.

2. Lược đồ quan hệ.

Hệ tiên đề Armstrong:

Cho X, Y, $Z \subseteq U$.

Qui tắc 1: Nếu $Y \subseteq X$ thì $X \to Y$.

Qui tắc 2: Nếu $X \to Y$ và $Y \to Z$ thì $X \to Z$.

Qui tắc 3: Nếu $X \to Y$ thì $X \cup Z \to Y \cup Z$.

Lược đồ quan hệ là một cặp (U, F),

a) U là tập các thuộc tính,

b) F là tập các phụ thuộc hàm.

Ta gọi F^+ là tập các phụ thuộc hàm được suy diễn từ F bằng cách áp dụng các tiên đề của Armstrong.

Cho $X \subseteq U$, gọi $X^+ = \{Y \in U \mid X \to Y \in F^+\}$ và ta có:

$$X \to Y \in F^+ \text{ khi và chỉ khi } Y \subseteq X^+.$$

Tập $K \subseteq U$ gọi là siêu khóa của lược đồ (U, F) nếu $K \to U \in F^+$. Gọi K là khóa của (U, F) nếu K là siêu khóa bé nhất của (U, F). Trong cách định nghĩa trên tất nhiên ta có cho lược đồ (U, F) thì bao giờ cũng có ít nhất một khóa.

Cho (U, F) là lược đồ quan hệ, ta gọi C (U, F) là tập tất cả các khóa của (U, F). Cho f là hàm đóng trên U ta gọi C(f) tập tất cả các khóa của f.

3. Quan hệ giữa khóa của hàm đóng và khóa của lược đồ quan hệ.

Trong phần này trình bày hai kết quả liên quan giữa khóa của hàm đóng và khóa của lược đồ quan hệ.

Định lý 1: Cho (U, F) là lược đồ quan hệ, chúng ta xây dựng ánh xạ: $f: 2^U \to 2^U$ như sau: $\forall X \in 2^U$: $f(X) = X^+$, thì ta có:

1) f là hàm đóng

2) $C(f) = C(U, F)$

Chứng minh: Trước tiên ta chứng minh 1).

a) X^+ là bao đóng của X. Vậy nên $X \subseteq X^+$, từ đó $X \subseteq f(X)$.

b) Mặt khác $X^+ = (X^+)^+$. Vậy nên $f(X) = f(f(X))$.

c) Từ $X \subseteq Y$ ta có bao đóng $X^+ \subseteq Y^+$, tức là $f(X) \subseteq f(Y)$.

Vậy ta có f là hàm đóng.

Bây giờ ta chứng minh phần 2). Cho K là khóa của lược đồ quan hệ (U, F), từ đó $K \to U$, và tất nhiên $U \subseteq K^+ \subseteq U$. Vậy $f(K) = U$, ta có k là siêu khóa của f. Giả sử K không phải là khóa tức tồn tại $K' \notin K$ và $f(K') = U$. Từ định nghĩa của f ta có $K'^+ = U$ tức $K' \to U$, điều này mâu thuẫn với K là khóa của (U, F).

Bây giờ giả sử K là khóa của f. Tất nhiên $f(K) = U$. Vậy $K^+ = U$. Từ đó $K \to U$, ta có K là siêu khóa của (U, F), Giả sử K không phải là khóa của (U, F) tồn tại K' khóa (U, F), $K' \notin K$ và $K' \to U$. Từ đó $K'^+ = U$ hay $f(K') = U$. Mâu thuẫn với K là khóa của f.

Định lý 2. Cho $f: 2^U \to 2^U$ là hàm đóng. Chúng ta xây dựng lược đồ quan hệ (U, F) như sau:

$$F = \{ X \to f(X) \mid X \in 2^U \}.$$

Thì $$C(f) = C(U, F).$$

Chứng minh: Trước tiên ta cần chứng minh rằng: $\forall X \in 2^U$ thì $f(X) = X^+$. Từ định nghĩa của hàm f ta có $X \to f(X)$. Vậy nên $f(X) \subseteq X^+$. Ta còn phải chứng minh bao hàm thức ngược chiều $X^+ \subseteq f(X)$.

Ta chứng minh bằng qui nạp

$$\forall n: X^{(n)} \subseteq f(X).$$

a) Ta có $X^{(o)} = X \subseteq f(X)$ vì f là hàm đóng.

b) Giả sử ta đã có $X^{(n)} \subseteq f(X)$. Bây giờ ta chứng minh $X^{(n+1)} \subseteq f(X)$.

$$X^{(n+1)} = X^{(n)} \cup \{ Y \mid Z \to Y \in F^+ \text{ và } Z \subseteq X^{(n)} \}.$$

Mặt khác từ $Z \in X^{(n)}$ suy ra

$$f(Z) \subseteq f(X^{(n)}) \subseteq f(f(X)) = f(X).$$

Vậy $\cup \{ f(Z) \mid Z \subseteq X^{(n)} \} \subseteq f(X)$.

hiển nhiên ta có $X^{(n+1)} \subseteq f(X)$. Cuối cùng ta được $X^+ = f(X)$. Áp dụng định lý 1 cho kết quả trên ta thu được

$$C(f) = C(U, F).$$

Qua kết quả của định lý 1 và định lý 2 ta thay trên một phần nào đó việc nghiên cứu khóa của lược đồ quan hệ, có thể qua việc nghiên cứu khóa của hàm đóng. Ngược lại việc nghiên cứu khóa của hàm đóng ta cũng có thể nghiên cứu thông qua việc nghiên cứu khóa của lược đồ quan hệ.

4. Khóa của hàm đóng.

Cho U là tập n phần tử $U = \{a_1, a_2 \dots a_n\}$, $f: 2^U \to 2^U$ là hàm đóng.

Với mọi $X \in 2^U$, ta gọi $\overline{X} = f(X) \setminus X$.

Ta định nghĩa hai tập P và T như sau

a) $T = \cup \{ X \mid X \in 2^U \text{ và } f(X) \neq X \}$

b) $P = \cup \{ \overline{X} \mid X \in 2^U \text{ và } f(X) \neq X \}$

Bổ đề 1. cho f là hàm đóng $X \subseteq U$ và $a \notin T$

Thì: $f(X \setminus a) = f(X)$ hoặc $f(X \setminus a) = f(X) \setminus a$

Chứng minh:

Ta xét hai trường hợp sau

a) Nếu $a \notin X$ thì $X = X \setminus a$ hiển nhiên $f(X) = f(X \setminus a)$

b) Nếu $a \in X$ và $a \notin T$ ta có $f(X) = X$ từ đó $f(X) \setminus a = X \setminus a$. Vậy $f(f(X) \setminus a) = f(X \setminus a)$ Mặt khác ta có

$$f(X) \setminus a \subseteq f(f(X) \setminus a) = f(X \setminus a) \subseteq f(X)$$

Vậy ta có:

$$f(X \setminus a) = f(X) \text{ hoặc } f(X \setminus a) = f(X) \setminus a$$

Bổ đề 2. Cho f là hàm đóng, khi đó, nếu $X \subseteq U$ thì

a) $f(X) \subseteq X \cup P$

b) $P \subseteq f(T)$

Chứng minh: a) Hiển nhiên

b) Cho $a \in P$ từ định nghĩa tồn tại $X \subseteq U$, $f(X) \neq X$ và $a \in f(X) \setminus X$. Hiển nhiên $X \subseteq T$ từ đó $f(X) \subseteq f(T)$. Vậy: $a \in f(T)$.

Bổ đề 3. Cho f là hàm đóng, $X \subseteq U$. Nếu

$$a \in f(X) \setminus P. \text{ Thì } a \in X.$$

Chứng minh. Ta có: $f(X) = X \cup P$, $a \in f(X)$ và $a \notin P$. Hiển nhiên $a \in X$.

Bổ đề 4. Cho f là hàm đóng trên U; $X, Y \subseteq U$

Nếu $a \notin T$ và $f(Y) \subseteq f(X)$ thì $f(Y \setminus a) \subseteq f(X \setminus a)$

Chứng minh: Vì $a \notin T$ theo bổ đề 1 ta có

$$f(X \setminus a) = f(X) \text{ hoặc } f(X \setminus a) = f(X) \setminus a$$

a) Nếu $f(X \setminus a) = f(X)$ ta có:

$$Y \setminus a \subseteq f(Y) \setminus a \subseteq f(X) \setminus a \subseteq f(X) = f(X \setminus a)$$

Vậy: $f(Y \setminus a) \subseteq f(f(X \setminus a)) = f(X \setminus a)$

b) Nếu $f(X \setminus a) = f(X) \setminus a$ ta có:

$$Y \setminus a \subseteq f(Y) \setminus a \subseteq f(X) \setminus a = f(X \setminus a)$$

Vậy $f(Y \setminus a) \subseteq f(f(X \setminus a)) = f(X \setminus a)$

Bổ đề 5. Cho f là hàm đóng $K \subseteq U$; Nếu $\exists a \in U$ và $a \in K \cap f(K \setminus a)$ thì K không phải là khóa của f.

Chứng minh: $a \in K$ Vậy $K \setminus a \notin K$ từ đó ta suy ra $f(K \setminus a) \subseteq f(K)$ Mặt khác từ $a \in f(K \setminus a)$ ta có: $a \cup (K \setminus a) \subseteq f(K \setminus a)$ tức là $K \subseteq f(K \setminus a)$ từ đó $f(K) \subseteq f(f(K \setminus a)) = f(K \setminus a) \subseteq f(K)$

Vậy K không phải là khóa của f.

Định lý 3. Cho f là hàm đóng trên U, K là khóa của f, thì $U \setminus P \subseteq K \subseteq (U \setminus P) \cup (P \cap T)$

Chứng minh: Trước tiên ta chứng minh cho $U \setminus P \subseteq K$ Giả sử điều đó không xảy ra tức là tồn tại $a \in (U \setminus P) \setminus K$ Tất nhiên $a \notin P$ và $a \notin K$; K là hàm đóng vậy $a \in f(K)$ Áp dụng bổ đề 3 ta có $a \in K$, điều này mâu thuẫn. Vậy

$$U \setminus P \subseteq K$$

Bây giờ ta chứng minh cho $K \subseteq (U \setminus P) \cup (P \cap T)$

Ta có $U = (U \setminus P) \cup P = (U \setminus P) \cup (P \cap T) \cup (P \setminus T)$ Giả sử $K \not\subseteq (U \setminus P) \cup (T \cap P)$ tức là tồn tại $a \in K \cap (P \setminus T)$ Vậy $a \in K$, $a \in P$ và $a \notin T$ Vì K là khóa f nên $f(K) = U = f(U)$, $a \notin T$ ta áp dụng bổ đề 4, ta có $f(K \setminus a) = f(U \setminus a)$, từ $a \notin T$ tức là $T \subseteq U \setminus a$ Vậy $f(T) \subseteq f(U \setminus a)$ Theo bổ đề 2, $P \subseteq f(T)$ và từ $a \in P$ ta có

$$a \in P \subseteq f(T) \subseteq f(U \setminus a) = f(K \setminus a)$$

Tức là $a \in K \cap f(K \setminus a)$ Bây giờ ta Áp dụng bổ đề 5 suy ra K không phải là khóa của f.

Định lý đã được chứng minh.

5. Giao các khóa của hàm đóng.

Bổ đề 6. Cho f là hàm đóng và $a \in P$, thì tồn tại khóa K của f mà $a \notin K$.

Chứng minh :

$$a \in P \text{ thì tồn tại } X \subseteq U \text{ mà } a \in f(X) \setminus X.$$

Gọi C là tập hợp $C \subseteq U : f(X) \cup C = U$ và $f(X) \cap C = \phi$ Hiển nhiên ta có: $U \subseteq f(X) \cup C \subseteq f(X \cup C) \subseteq U$. Vậy nên $f(X \cup C) = U$ từ đó ta suy ra tồn tại khóa $K \subseteq X \cup C$ và tất nhiên $a \notin K$.

Định lý 4. Cho f là hàm đóng trên U. Thì giao tất cả các khóa của f, ký hiệu là I sẽ bằng

$$I = U \setminus P$$

Chứng minh: Từ định lý 3 ta có $U \setminus P \subseteq I$. Theo bổ đề 6 ta nhận được $I \cap P = \phi$ Vậy nên $I \subseteq U \setminus P$. Cuối cùng $I = U \setminus P$

Nhận ngày 15-4-1986

TÀI LIỆU THAM KHẢO

1. **W.W. Armstrong:** Dependency structures of Data Base Relationship (Imforonation processing 74) North Holland Pub. C. Amsterdam.

2. **G. Burosch, J. Demetrovics and G.O.H. Katoma.** The Poset of Cloeure.

3. **Le Van Bao:** Sufficient and Necessary condition for which a Relation scheme has precisely one key (to appear).

4. **Nguyễn Xuân Huy:** Đẳng cấu giữa giàn các cấu trúc phụ thuộc hàm và giàn các hàm đóng.

Tạp chí Toán học, 1, 1986.

РЕЗЮМЕ

О СООТВЕТСТВИИ МЕЖДУ КЛЮЧАМИ ЗАМКНУТОЙ ФУНКЦИИ И КЛЮЧАМИ РЕЛЯЦИОННОЙ СХЕМЫ

Одной из важных задач при изучении реляционых моделей является нахождение наиболее эффективных алгоритмов определения ключей заданной реляционной схемы. В данной работе применяется понятие замкнутых функции для исследования некоторых свойств ключей схемы. Доказывается соответствие между множествами ключей реляционной схемы и замкнутой функции. Формируются некоторые свойства замкнутой функции и наконец описывается формула взятья пересечения ключей замкнутой функции.

Tạp chí KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN TẬP II (1986), Số 3 (9 - 13)

MỘT SỐ KẾT QUẢ VỀ HIỆU QUẢ CỦA THUẬT TOÁN NHẬN DẠNG TỔ HỢP

NGÔ QUỐC TẠO, HOÀNG KIẾM

I — ĐẶT VẤN ĐỀ

Phương pháp tổ hợp các thuật toán nhằm nhận được thuật toán có hiệu quả hơn hiệu quả của mỗi thuật toán ban đầu, cụ thể là, nếu xác suất sai lầm của mỗi thuật toán là ε_i thì xác suất sai lầm của thuật toán tổ hợp có thể nhỏ hơn ε_i. Tính cho đến nay đã xuất hiện nhiều công trình đề cập đến thuật toán nhận dạng tổ hợp với các công thức tính R_n [1], [2], song đó là những công thức đánh giá chưa được chứng minh chặt chẽ. Trong bài này trình bày một số đánh giá cho R_n chặt hơn một số kết quả đã công bố.

Phương pháp tổ hợp các thuật toán được dựa trên quy tắc quyết định theo đa số được phát biểu như sau:

Giả sử cho trước N thuật toán nhận dạng $\mathcal{A}_1, ..., \mathcal{A}_N$ phân loại các dạng $\{S_i | i=1,...,n$ thành 2 lớp K_1 và K_2. Người ta xác định các ma trận $\|a_{ij}^k\|$ $n \times 2$, $k=1,..., N$ như sau:

$$a_{ij}^k = \begin{cases} 1, \text{ nếu thuật toán } \mathcal{A}_k \text{ xếp } S_i \text{ vào lớp } K_j \\ 0, \text{ ngược lại.} \end{cases}$$

Quy tắc quyết định của thuật toán nhận dạng tổ hợp như sau:

$$D_j(S_i) = \begin{cases} 1, \text{ nếu } \sum_{i=1}^{N} a_{ij}^k > E \\ 0, \text{ ngược lại.} \end{cases}$$

Ở đây ký hiệu $E = [LN/2]$ (số nguyên tố lớn nhất không vượt quá N/2).

$D_j(S_i) = 1$ có nghĩa là thuật toán nhận dạng tổ hợp xếp S_i vào lớp K_j, $i=1,..., n$; $j = 1,2$. Khi các thuật toán $\mathcal{A}_1, ..., \mathcal{A}_n$ độc lập và mỗi thuật toán có xác suất sai lầm $0<\varepsilon<1/2$ thì xác suất sai lầm của thuật toán nhận dạng tổ hợp được xác định theo công thức sau:

$$R_N = \sum_{k>E} C_N^k \varepsilon^k (1-\varepsilon)^{N-k}$$

Trong II sẽ chỉ ra rằng $\forall \varepsilon$, $0<\varepsilon<1/2$ thì $R_N < \varepsilon$. (Bất đẳng thức này chứng tỏ rằng thuật toán tổ hợp có hiệu quả hơn mỗi thuật toán đã cho). Và $R_N \leqslant \exp\{-N \times C(\varepsilon)\}$ (tốc độ tiến tới 0 của R_N có cấp hàm mũ theo N).

Ta ký hiệu $\vec{\lambda}$ là vectơ có N tọa độ nhận giá trị 0,1 tức là

$$\vec{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_N)$$
$$\lambda_k \in \{0,1\},\ k = 1,..., N$$

và

$$TA = \left\{ \vec{\lambda} \,\middle|\, \sum_{k=1}^{N} \frac{\lambda_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} = \frac{1}{2} \sum_{k=1}^{N} \frac{1}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right\}$$

Trong trường hợp các thuật toán $\mathcal{A}_i$ độc lập và có xác suất sai lầm ε_i tương ứng, thì xác suất sai lầm R_N có thể xác định theo công thức:

$$R_N = \sum_{\vec{\lambda} \in TA} \prod_{k=1}^{N} \varepsilon_k^{\lambda_k} (1 - \varepsilon_k)^{1-\lambda_k}$$

Hơn nữa

$$R_N \leqslant \frac{4}{N^3} \Big(\sum_{k=1}^{N} \sqrt{\varepsilon_k(1-\varepsilon_k)} \Big)^2 \Big/ \Big(1 - 2 \sum_{k=1}^{N} \varepsilon_k/N\Big)^2.$$

Các khẳng định này được chỉ ra trong [1].

II – ĐÁNH GIÁ XÁC SUẤT SAI LẦM CỦA THUẬT TOÁN NHẬN DẠNG TỔ HỢP ĐỐI VỚI N THUẬT TOÁN ĐỘC LẬP CÓ CÙNG XÁC SUẤT SAI LẦM GIỐNG NHAU ε.

Bổ đề 1 [1]: Nếu cho trước N thuật toán nhận dạng độc lập có cùng xác suất sai lầm ε thì xác suất sai lầm của thuật toán nhận dạng tổ hợp thỏa mãn:

$$R_N = \sum_{k > E} C_N^k \varepsilon^k (1 - \varepsilon)^{N-k}.$$

Mệnh đề 1: Với ε thỏa mãn $0 < \varepsilon < 1/2$ thì xác suất sai lầm của thuật toán nhận dạng tổ hợp thỏa mãn:

$$R_N < \varepsilon.$$

Chứng minh: Trước hết ta nhận thấy hàm $f_{a,b}(\varepsilon) = \varepsilon^a(1 - \varepsilon)^b$, với $0 < b \leqslant a$, tăng chặt trên $[0, 1/2]$, tức là

$$\forall \varepsilon_1, \varepsilon_2 \text{ thỏa mãn } 0 \leqslant \varepsilon_1 < \varepsilon_2 \leqslant 1/2$$

thì $f_{a,b}(\varepsilon_1) < f_{a,b}(\varepsilon_2)$.

Đặt $T_N = \sum_{k > E} C_N^k \varepsilon^{k-1}(1 - \varepsilon)^{N-k}$, do bổ đề 1 ta có $R_N = T_N . \varepsilon$.

Do nhận xét trên, với $k > E$ hàm $f_{k-1, N-k}(\varepsilon)$ tăng chặt trên $[0, 1/2]$. Do đó $f_{k-1, N-k}(\varepsilon) \leqslant f_{k-1, N-k}(1/2)$.

Hay $\varepsilon^{k-1}(1 - \varepsilon)^{N-k} < \left(\frac{1}{2}\right)^{k-1} \left(1 - \frac{1}{2}\right)^{N-k} = 2^{N-1}$.

Vì thế

$$T_N < \sum_{k > E} C_N^k \left(\frac{1}{2}\right)^{N-1} = \left(\frac{1}{2}\right)^{N-1} \sum_{k > E} C_N^k.$$

Dễ dàng nhận thấy rằng:

$$\sum_{k > E} C_N^k \leqslant 2^{N-1},$$

Do đó $T_N < 1$ hay $R_N < \varepsilon$. Mệnh đề được chứng minh.

Mệnh đề 2: Với các điều kiện tương tự như bổ đề 1, ta nhận được

$R_N < \exp\{-N \times c(\varepsilon)\}$, với $c(\varepsilon) = \ln \frac{1}{2\sqrt{\varepsilon(1-\varepsilon)}}$.

Chứng minh: Trước hết ta thấy rằng với $0 < \varepsilon < 1/2$

thì $\ln \frac{1 - \varepsilon}{\varepsilon} > 0$ và $\ln \frac{1}{2\sqrt{\varepsilon(1-\varepsilon)}} > 0$.

$\forall \alpha > 0, \forall k > E$ ta đều có:

$\exp(\alpha k) \geqslant \exp(\alpha.N/2)$ hay $\exp(-\alpha.N/2 + \alpha.k) \geqslant 1$. Kết hợp với bổ đề 1 ta suy ra:

$$R_N < \exp(-\alpha_n, N/2) \sum_{k > E} C_N^k \varepsilon^k (1 - \varepsilon)^{N-k} \exp(\alpha.k) =$$

$$= \exp(-\alpha . N/2) \sum_{k=1}^{N} C_N^k \varepsilon^k (1-\varepsilon)^{N-k} \exp(\alpha . k) = \left(\varepsilon . \exp\left(\frac{\alpha}{2}\right) + (1 - \varepsilon) \exp\left(-\frac{\alpha}{2}\right)\right)^N.$$

Chọn $\alpha = \ln \frac{1-\varepsilon}{\varepsilon}$ suy ra $R_N \leqslant (2\sqrt{\varepsilon(1-\varepsilon)})^N$. Đặt $c(\varepsilon) = \ln \frac{1}{2\sqrt{\varepsilon(1-\varepsilon)}}$ ta được

$R_N < \exp\{-N \cdot c(\varepsilon)\}$. Mệnh đề được chứng minh. Như vậy, nếu như ta tổ hợp càng nhiều thuật toán thành toán tổ hợp thì hiệu quả của thuật toán tổ hợp càng cao. Phương pháp này được ứng dụng rộng rãi trong trí tuệ nhân tạo và chẩn đoán.

III – ĐÁNH GIÁ XÁC SUẤT SAI LẦM CỦA THUẬT TOÁN NHẬN DẠNG TỔ HỢP TRONG TRƯỜNG HỢP CÁC XÁC SUẤT SAI LẦM CỦA MỖI THUẬT TOÁN KHÁC NHAU

Quy tắc quyết định của thuật toán nhận dạng tổ hợp như sau:

$$D_j(S_i) = \begin{cases} 1, \text{ nếu } \sum\limits_{k=1}^{N} \dfrac{a_{ij}^k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} > \dfrac{1}{2} \sum\limits_{k=1}^{N} \dfrac{1}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \\ 0, \text{ ngược lại} \end{cases}$$

Bổ đề 2: Nếu mỗi thuật toán ban đầu có xác suất sai lầm ε_i tương ứng, thì xác suất sai lầm của thuật toán tổ hợp được xác định như sau:

$$R_N = \sum_{\vec{\lambda} \in TA} \prod_{k=1}^{N} \varepsilon_k^{\lambda_k} (1-\varepsilon_k)^{1-\lambda_k}$$

$\vec{\lambda}$ và TA được xác định như trong I.

Chứng minh: Dễ dàng suy ra từ tính độc lập của các thuật toán đã cho. Bổ đề 1 là trường hợp đặc biệt của bổ đề 2 khi $\varepsilon_i = \varepsilon$, $\forall i$.

Mệnh đề 3: Với các điều kiện tương tự như bổ đề 2 thỏa mãn thì

$$R_N \leqslant \frac{4}{N^3} \left(\sum_{k=1}^{N} \sqrt{\varepsilon_k(1-\varepsilon_k)} \right)^2 \Big/ \left(1 - 2 \sum_{k=1}^{N} \tau_k / N \right)^2$$

Chứng minh:

Trước hết ta chứng minh bổ đề sau:

Bổ đề 3: Cho hai dãy số thực $a_1, a_2, \ldots, a_N$ và $b_1, b_2, \ldots, b_N$.

1) Nếu $\forall i, j$; $i > j$, $a_i \leqslant a_j$, $b_i \geqslant b_j$ thì

$$\sum_{i=1}^{N} a_i \sum_{i=1}^{N} b_i \geqslant N \sum_{i=1}^{N} a_i b_i$$

2) Nếu $\forall i, j$ $(a_i - a_j)(b_i - b_j) \leqslant 0$ thì

$$\sum_{i=1}^{N} a_i \sum_{i=1}^{N} b_i \geqslant N \sum_{i=1}^{N} a_i b_i.$$

Chứng minh: 1) chính là bất đẳng thức Trêbưsep.

2) Chỉ cần sắp xếp dãy $\{b_i\}$ theo thứ tự tăng dần thì ta cũng nhận được $\{a_i\}$ có thứ tự hoán vị tương ứng với $\{b_i\}$ sẽ giảm dần. Áp dụng 1) ta được 2) – điều phải chứng minh.

Chứng minh mệnh đề:

Từ bổ đề 2 ta suy ra được

$$R_N \leqslant \left(\frac{1}{2} \sum_{k=1}^{N} \frac{1-2\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^{-2} \sum_{\vec{\lambda} \in TA} \prod_{k=1}^{N} \varepsilon_k^{\lambda_k} (1-\varepsilon_k)^{1-\lambda_k}$$

$$\times \left(\sum_{k=1}^{N} \frac{\lambda_k - \varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^2$$

$$\to R_N \leqslant \left(\frac{1}{2} \sum_{k=1}^{N} \frac{1-2\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^{-2} \sum_{\vec{\lambda} \in \{0,1\}^N} \prod_{k=1}^{N} \varepsilon_x^{\lambda_k} (1-\varepsilon_k)^{1-\lambda_k}$$

$$\times \left(\sum_{k=1}^{N} \frac{1-\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^2.$$

Dễ dàng thấy rằng:

$$\sum_{\vec{\lambda}} \prod_{k=1}^{N} \varepsilon_k^{\lambda_k} (1-\varepsilon_k)^{1-\lambda_k} \left(\sum_{k=1}^{N} \frac{1-\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^2 = N.$$

Từ đó ta có bất đẳng thức:

$$R_N \leqslant \left(\frac{1}{2} \sum_{k=1}^{N} \frac{1-2\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \right)^{-2}$$

Đặt $a_k = \dfrac{1-2\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}}$, $b_k = \sqrt{\varepsilon_k(1-\varepsilon_k)}$ và áp dụng bổ đề 3 ta có:

$$\sum_{k=1}^{N} \frac{1-2\varepsilon_k}{\sqrt{\varepsilon_k(1-\varepsilon_k)}} \sum_{k=1}^{N} \sqrt{\varepsilon_k(1-\varepsilon_k)} \geqslant N \sum_{k=1}^{N} (1-2\varepsilon_k)$$

$$= N^2 \left(1 - 2 \sum_{k=1}^{N} \varepsilon_k / N\right). \qquad (2)$$

Áp dụng (1) và (2) ta được

$$R_N \leqslant \frac{4}{N^3} \left(\sum_{k=1}^{N} \sqrt{\varepsilon_k(1-\varepsilon_k)} \right)^2 / \left(1 - 2 \sum_{k=1}^{N} \varepsilon_k / N\right)^2,$$

Nhận xét: – Nếu áp dụng bất đẳng thức Svac cho 2N số thực ta có:

$$\sum_{k=1}^{N} \sqrt{\varepsilon_k(1-\varepsilon_k)} \leqslant \sqrt{\sum_{k=1}^{N} \varepsilon_k} \cdot \sqrt{\sum_{k=1}^{N} (1-2\varepsilon_k)}$$

Ta thu được $R_N \leqslant \dfrac{4}{N} \dfrac{\bar{\varepsilon}(1-\bar{\varepsilon})}{(1-2\bar{\varepsilon})^2}$; trong đó $\bar{\varepsilon} = \sum_{k=1}^{N} \varepsilon_k$,

Trong trường hợp $\varepsilon_i = \varepsilon$, $\forall i$ ta thu được $R_N \leqslant \dfrac{4}{N} \dfrac{\varepsilon(1-\varepsilon)}{(1-2\varepsilon)^2}$.

Như vậy trong trường hợp các $\varepsilon_i = \varepsilon$, $\forall i$, thì kết quả II mạnh hơn III.

Một số vấn đề còn mở:

– Phát biểu định nghĩa cho thuật toán độc lập.

— Trong trường hợp có định nghĩa về thuật toán độc lập thì hiệu quả của thuật toán tổ hợp từ những thuật toán không độc lập sẽ ra sao? Chúng tôi chân thành cảm ơn tiến sĩ Bạch Hưng Khang và phòng nhận dạng Viện KHTTĐK về sự giúp đỡ nhiệt tình và có những gợi ý qúy báu cho bài báo này. Chúng tôi xin cám ơn đồng nghiệp Nguyễn Thanh Thủy trường đại học Bách Khoa Hà Nội, đã có những ý kiến bổ ích cho bài báo.

Nhận ngày 8-3-1986

TÀI LIỆU THAM KHẢO

1. **Hoàng Kiếm**, Some methods improving the efficiency of pattern recognition algorithms. Computer and artificial intelligence, 3(1984), N°4, pp. 347–359, Czeckoslovakia.

2. **Gaillat G.**, Borne Superieuse de la probabilité d'erreur avec la gégle de decision majoritaire. Congres AFCET–IRIA, Paris 1978.

3. ЖУРАВЛЁВ Ю И., Об алгебраическом походе к решению задач распознавания или классификации. ВКН: Побл. Кибернетики. Выч. зз. М: Наука, 1978 с. 5–68.

4. РЯЗАНОВ В.В., Комитетный синтез алгоритмов распознавания и классификации. Ж. вычисл. Матем. и матем. физ., 1981, Т. 21, И°6. с. 1533–1543.

5. РЯЗАНОВ В.В., О синтезе классифпрующих алгоритмов на конечных множествах алгоритмов классификации. Ж. вычисл. Матем. и Матем. физ., Т.22, И°2, с. 429–440.

ABSTRACT

Some results of the efficiencies of the combined pattern recognition algorithm

The paper presents the evalution of the error probability of the combined pattern recognition algorithm based on N. independing algorithms. These main results are follows: if every given algorithm has error probability $0 < \varepsilon < 1/2$ then $R_N < \exp\{-N \cdot c(\varepsilon)\}$, here

$$c(\varepsilon) = \ln \frac{1}{2\sqrt{\varepsilon(1-\varepsilon)}}.$$

When all of the given algorithms have error probabil ty $0 < \varepsilon_i < 1/2$ respectively, then the error probability of combined pattern recognition algorithms satisfies

$$R_N \leqslant \frac{4}{N^3}\left(\sum_{i=1}^{N} \sqrt{\varepsilon_i(1-\varepsilon_i)}\right)^2 / \left(1 - 2\sum_{i=1}^{N} \varepsilon_i/N\right)^2,$$

TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN TẬP II (1986), Số 3 (14–18; 27)

MỘT CÁCH TIẾP CẬN VIỆC BIỂU DIỄN ẢNH CỦA ĐỐI TƯỢNG QUA BÓNG CỦA NÓ

PHẠM NGỌC KHÔI

TÓM TẮT NỘI DUNG

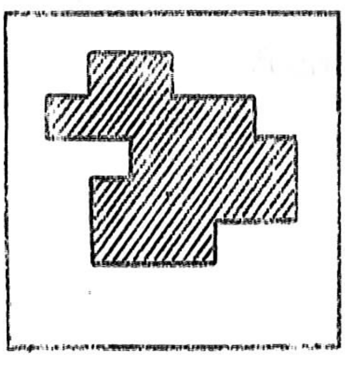
Bài báo đề cập đến một kiểu dữ liệu đặc biệt dùng để biểu diễn đối tượng trên ảnh. đó là cây tứ phân ảo. Các toán tử và hàm thao tác trên cây tứ phân ảo được đưa ra. cùng với các tính chất của chúng. Một vài áp dụng trực tiếp của cây tứ phân ảo được trình bày. Phần cuối bài báo có đưa ra ước lượng hiệu quả của biểu diễn cây tứ phân so với mã độ dài loạt và mã dây chuyền đối với một ảnh cụ thể cho trước.

I – MỞ ĐẦU

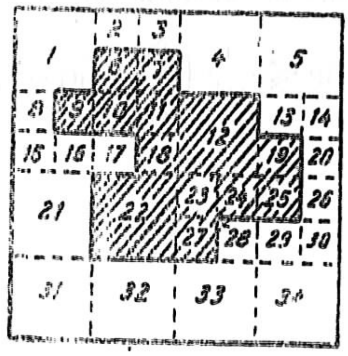
Bài toán xử lý ảnh dựa trên bóng các đối tượng được đề cập đến trong nhiều lĩnh vực áp dụng khác nhau: điều tra địa hình, phân tích sinh–y học và đặc biệt là trong việc kiểm định chất lượng các sản phẩm công nghiệp [1, 2]. Đối với các ảnh nhị phân, để biểu diễn các đối tượng trên ảnh, có 3 cách mã hóa sau được dùng một cách phổ biến nhất: mã độ dài loạt (run length code), mã dây chuyền (chain code), và cây tứ phân (quadtree) [7]. Bài báo này là mở rộng một vài kết quả nghiên cứu về cây tứ phân đã được công bố trong một bài báo trước [9]. Để cho tiện, ta giữ nguyên các ký pháp đã dùng.

Ở đây, một ảnh nhị phân được hiểu là một ma trận có cỡ $N \times N$, ($N = 2^n$), mỗi phần tử của nó nhận giá trị 0 hoặc 1 tùy thuộc giá trị của điểm ảnh tương ứng là trắng hoặc đen. Tập hợp các phần tử 1 liên thông với nhau thể hiện một đối tượng trên ảnh.

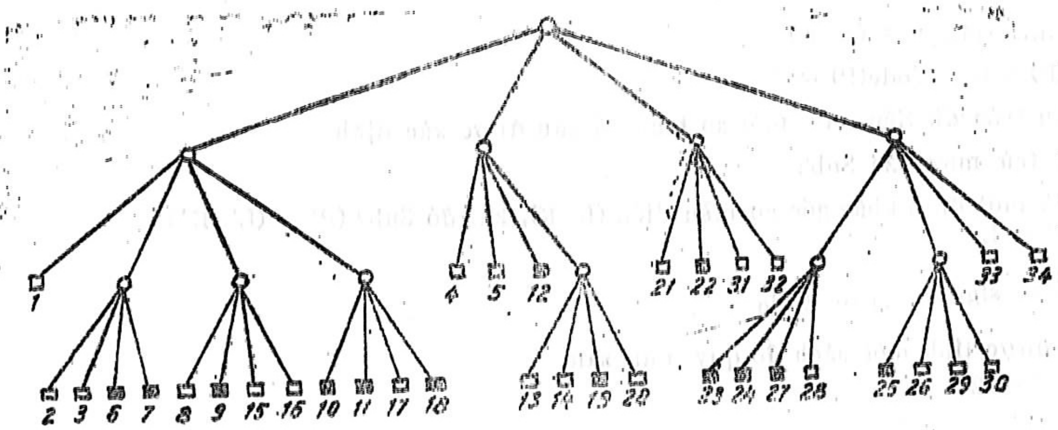
Cây tứ phân là một cấu trúc dữ liệu phân cấp dùng để biểu diễn cô đọng ảnh nhị phân 2 chiều. Cấu trúc này dựa trên việc chia một cách liên tiếp ma trận thành các khối phần tư cho đến khi ta thu được các khối đồng màu (tối thiểu là một điểm ảnh), sao cho trong một khối chỉ bao gồm toàn giá trị 1 hoặc 0. Để ý rằng nếu ảnh có cỡ $2^n \times 2^n$ thì sau lần chia thứ k, mỗi khối phần tư có cỡ $2^{n-k} \times 2^{n-k}$. Ví dụ như ảnh trong hình 1a tương ứng với quá trình làm mịn vẽ ở hình 1b. Quá trình này có thể biểu diễn bởi một cây tứ phân như sau: gốc của cây tương ứng với toàn bộ ảnh, 4 con của một đỉnh tương ứng với 4 khối phần tư, và các lá tương ứng với các khối phần tư đồng màu (không cần chia nữa) và nhận giá trị BLACK hoặc WHITE tùy theo các phần tử ảnh trong đó là 1 hay 0. Các đỉnh không phải là lá ứng với các khối phần tư có chứa cả hai loại điểm ảnh 1 và 0, mang giá trị thuộc tính GRAY. Biểu diễn cây cấp 4 của hình 1a được vẽ ở hình 1c.



Hình 1a
Đối tượng trên ảnh



Hình 1b
Việc chia mịn ảnh thành các khối phần tư



Hình 1c :

Biểu diễn cây tứ phân của ảnh trên hình 1a

Mặc dù ở đây một đối tượng được biểu diễn bởi hợp của một số khối vuông, nhưng các khối ở đây có độ dài chuẩn (lũy thừa của 2) và các vị trí chuẩn. Vì ma trận ban đầu có kích thước $2^n \times 2^n$, chiều cao của cây nhiều nhất là n.

Một trong những nghiên cứu đầu tiên về cây tứ phân và áp dụng của nó trong xử lý ảnh là bài báo của Sidhu và Boute [3]. Gần đây, Samet và Rosenfeld đã có những bài tổng quan về lịch sử của cây tứ phân và việc biểu diễn ảnh bởi bóng của đối tượng [4].

II – CẤU TRÚC DỮ LIỆU CHO CÂY TỨ PHÂN

2.1. Cây tứ phân ảo

Liên quan đến biểu diễn cây tứ phân, một cấu trúc dữ liệu truyền thống được sử dụng để tính toán các đặc trưng ảnh và phân tích ảnh [5]. Mỗi đỉnh của cây tứ phân có thể được lưu trữ dưới dạng một record chứa 6 trường. 5 trường đầu chứa tên các con trỏ trỏ tới đỉnh cha và 4 đỉnh con với nhãn : NW, NE, SW, SE, tương ứng với 4 khối con của một khối: Tây–Bắc, Đông–Bắc, Tây–Nam, Đông–Nam. Trường thứ 6 có tên là NODETYPE mô tả nội dung của khối: WHITE nếu khối chỉ chứa các điểm ảnh có giá trị 0, BLACK nếu khối chỉ chứa các điểm ảnh có giá trị 1, và GRAY nếu khối chứa cả 2 loại điểm ảnh.

Mới đây có một số tác giả đề xướng những cấu trúc dữ liệu cho riêng biểu diễn cây tứ phân, không sử dụng cơ chế con trỏ. Sau đây ta sẽ dùng một cấu trúc dữ liệu, gọi là cây tứ phân ảo, do John và Iyengar đưa ra [6].

Giả sử rằng một khối vuông có 4 khối phần tư được đánh chỉ số như sau

0	1
2	3

tương ứng với các khối phần tư NW, NE, SW, SE. Mỗi khối phần tư trong ảnh được ứng với một đỉnh trong cây tứ phân, đỉnh này có 4 con được đánh số từ trái sang phải theo thứ tự 0, 1, 2, 3. Ta sẽ biểu diễn mỗi đỉnh P của cây tứ phân bởi một cặp số nguyên (L, K) trong đó :

– L là mức của đỉnh (tức là khoảng cách từ gốc đến đỉnh), $0 \leqslant L \leqslant n$,

– K được xác định bởi $K = \sum_{i=0}^{L-1} a_i\, 4^i \quad 0 \leqslant a_i \leqslant 3$

ở đây, $(a_{L-1}, \ldots, a_1, a_0)$ là đường đi từ gốc tới đỉnh P.

Ký hiệu

$K = a_{L-1} \ldots a_1 a_0$, $P' = (L, K)$

Level (P) = L , Code(P) = K.

Trên cấu trúc dữ liệu này, một số toán tử sau được xác định

1) Phép trừ mượn 2: Sub2

Đặt P là một đỉnh khác gốc có biểu diễn (L, K), khi đó Sub2 (P) = (L', K')
trong đó $L' = L$

$$K' = \text{sign } a'_{L-1} \ldots a'_1 a'_0$$

với các a_i được tính một cách đệ quy như sau

– $b_0 := 2$

– $a'_i := ((a_i + 4) - b_i) \bmod 4$

$$b_{i+1} := \begin{cases} 0 & \text{nếu } a_i \geqslant b_i; \quad i = 0,1,\ldots L-1 \\ 2 & \text{ngược lại} \end{cases}$$

$$\text{sign} := \begin{cases} + & \text{nếu } b_{L-1} = 0 \\ - & \text{nếu } b_{L-1} = 2 \end{cases}$$

Ví dụ: Sub2 (3,311) = (3,133), Sub2 (3,111) = (3, –333)

2) Phép quay thuận: Rot^+

Trước hết, Rot^+ được định nghĩa trên tập hữu hạn $\{0, 1, 2, 3\}$

$Rot^+(0) = 1$, $Rot^+(1) = 3$,

$Rot^+(2) = 0$, $Rot^+(3) = 2$.

Sau đó Rot^+ được mở rộng một cách tự nhiên trên tập vô hạn $\{0, 1, 2, 3\}^*$:

$$Rot^+(xa) = Rot^+(x)\, Rot^+(a)$$

trong đó $x \in \{0, 1, 2, 3\}^*$, $a \in \{0, 1, 2, 3\}$.

Tiếp theo, Rot^+ được mở rộng một lần nữa trên tập các đỉnh của cây tứ phân

$$Rot^+(L, K) = (L, Rot^+(K)).$$

3) Phép quay nghịch: Rot^-

Rot^- được định nghĩa là: $Rot^- = (Rot^+)^{-1}$.

2.2. Một vài tính chất của cây tứ phân ảo

Ta có thể nhận được một số tính chất sau đây của cây tứ phân ảo dùng để biểu diễn ảnh nhị phân [8].

Tính chất 1: Mỗi đỉnh P = (L, K) ứng với một khối phần tư của ảnh có cỡ 2^{n-L}, nói riêng, nếu L = n thì đỉnh P ứng với một điểm ảnh.

Tính chất 2: Rot^+ (P) cho ta biểu diễn của chính đỉnh P sau khi quay ảnh một góc 90° thuận chiều kim đồng hồ.

Tính chất 3: Rot^- (P) cho ta biểu diễn của chính đỉnh P sau khi quay ảnh một góc 90° ngược chiều kim đồng hồ.

Định lý 4: Nếu Sign (Code Sub2(P))) > 0 thì Sub2(P) xác định khối phần tư giáp P về phía Bắc, ngược lại, P nằm sát biên phía bắc của ảnh.

Hệ quả 5: Cho biến side thuộc tập {Bắc, Tây, Nam, Đông} tập này được mã hóa bởi {0, 1, 2, 3}. Đặt P là một khối phần tư của ảnh 1. Khối phần tư Q được xác định bởi công thức

$$Q = Rot^{-side}(Sub2(Rot^{side}(P))).$$

Khi đó, nếu sign (Code (Sub2 (Rot^{side} (P)))) > 0 thì Q là khối phần tư nằm kề P về phía side và có cùng cỡ như P.

Tính chất 6: Nếu một điểm ảnh P = (L, K) có các tọa độ biểu diễn dưới dạng nhị phân bởi

$x = x_{n-1} \dots x_1 x_0$, $y = y_{n-1} \dots y_1 y_0$, $x_i, y_i \in \{0, 1\}$ thì ta có hệ thức sau

$$K = \sum_{i=0}^{i-1} (2y_i + x_i) \cdot 4^i,$$

Tính chất 7: Nếu đỉnh P ứng với một khối vuông ($L < n$), thì các tọa độ của góc trên bên trái được tính bởi

$$x(P) = \sum_{i=0}^{i-1} (a_i \bmod 2) 2^{i+n-L}, \quad y(P) = \sum_{i=0}^{L-1} (a_i \operatorname{div} 2) 2^{i+n-L}.$$

Các tính chất trên đóng một vai trò cơ bản trong việc xây dựng các thuật toán xử lý ảnh trên biểu diễn mã tứ phân: tính mômen và các đặc trưng hình học của ảnh, phân tích tính liên thông của ảnh, đối sánh ảnh, chuyển đổi từ biểu diễn mã cây tứ phân sang các biểu diễn mã khác và ngược lại, Có thể xem các thuật toán chi tiết trong [8]

III – ĐÁNH GIÁ HIỆU QUẢ CỦA BIỂU DIỄN MÃ CÂY TỨ PHÂN SO VỚI BIỂU DIỄN KHÁC

Các kỹ thuật mã hóa ảnh khác nhau đều có một lợi điểm chung là rút gọn bộ nhớ lưu trữ ảnh. Tuy nhiên đối với một ảnh cụ thể, kỹ thuật mã hóa này lại tỏ ra tốt hơn kỹ thuật khác. Trong phần này ta sẽ chỉ ra các điều kiện để chuyển sang các loại mã hóa khác, khi cho ảnh biểu diễn bởi mã cây tứ phân

3.1. Đối với mã độ dài loạt

Giả sử một ảnh có cỡ $2^n \times 2^n$ được biểu diễn bởi cây tứ phân Q với Q đỉnh. Mỗi đỉnh P thuộc Q có 3 thuộc tính: Level (P), Code (P) và Color (P), chúng tương ứng cần $\log_2 n$ bit, 2n bit và 2 bit để lưu trữ. Vậy, bộ nhớ cần thiết cho Q là:

$$Q(2 + 2n + \log_2 n).$$

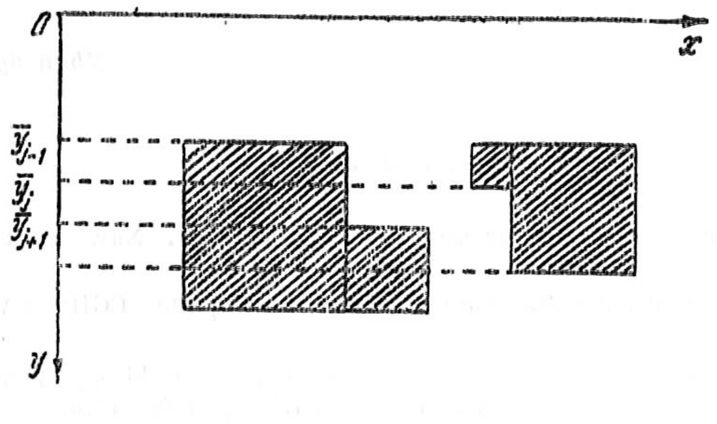
Với mỗi đỉnh $P = (L, K) \in Q$, theo các tính chất 1 và 7, ta định nghĩa các hàm

$$X(P) = \sum_{t=0}^{L-t} (a_i \bmod 2)\, 2^{i+n-L}$$

$$y(P) = \sum_{t=0}^{L-1} (a_i \operatorname{div} 2)\, 2^{i+n-L}$$

$$s(P) = 2^{n-L}.$$

Đặt $\mathcal{B}$ là tập các lá đen của cây tứ phân, ta có tập số nguyên BY được xác định như sau $BY = \{y(P) : P \in \mathcal{B}\} \cup \{y(P) + s(P) + 1, P \in \mathcal{B}\}$ Với quan hệ bằng nhau « = » theo nghĩa thông thường trên BY, ta có tập $\overline{BY} = BY/=$. Ký hiệu b là lực lượng của tập $\overline{BY}$. mỗi $\overline{y_j} \in \overline{BY}$ xác định dòng ảnh trên đó có thể xuất hiện một hình trạng mới (hình 2).



Hình 2: Các tọa độ y trên đó có thể xuất hiện hình trạng mới của loạt chạy các điểm ảnh

Để xác định số loạt chạy của các điểm ảnh, với mỗi $\overline{y_j} \in \overline{BY}$ được đặt tương ứng tập $H_j \in \mathcal{B}$ và quan hệ $\gamma_j \subset H_j \times H_j$ theo cách sau

$$H_j = \{P \in \mathcal{B} : \overline{y_j} \in [y(P), y(P) + s(P)]\}$$

$$P_m \gamma_j P_l \leftrightarrow x(P_m) = x(P_l) + s(P_l) + 1 \text{ hoặc } x(P_l) = x(P_m) + s(P_m) + 1 ; \; P_m, P_l \in H_j$$

Tiếp theo, nếu đặt γ_j^* là bao đóng phản xạ, bắc cầu của γ_j trên H_j thì ta có tập $H_j^* = H_j/\gamma_j^*$. mỗi phần tử của nó chứa các khối kề nhau liên tiếp theo chiều ngang.

Tổng số loạt chạy trong biểu diễn mã độ dài loạt sẽ là:

$$\sum_{j=1}^{b} (\overline{y}_{j+1} - \overline{y_j}) \operatorname{card} \sum (H_j^*), \quad \text{trong đó } \overline{y}_{b+1} = 2^n.$$

Như vậy, tổng số khối lượng bộ nhớ cần dùng cho ảnh trong biểu diễn mã độ dài loạt là

$$2(n+1) \sum_{j=1}^{b} (\overline{y_{j+1}} - \overline{y_j}) \operatorname{card} (H_j^*) \text{ bit.}$$

Dựa trên cơ sở đó, ta có thể kết luận rằng khi cho một ảnh biểu diễn bởi mã cây tứ phân với Q đỉnh, mã độ dài loạt sẽ tối ưu hơn về mặt lưu trữ khi

$$\sum_{j=1}^{b} (\overline{y_{j+1}} - \overline{y_j}) \operatorname{card} (H_j^*) < \frac{Q(2 + 2n + \log_2 n)}{2(n+1)}$$

3. 2. Đối với mã dây chuyền

Với mỗi B thuộc tập các lá đen $\mathcal{B}$, bằng một thủ tục đơn giản (xem [9] có thể xác định được tập

$$N_B = \{P \in \mathcal{B} : P \text{ là kề với } B\}.$$

Độ dài biên thuộc B được tính bởi

$$4(n - \text{Level}(B)) - \sum_{P \in N_B} \min(n - \text{Level}(B), n - \text{Level}(P))$$

Do đó số vectơ chỉ phương trong biểu diễn mã dây chuyền là

$$4 \sum_{B \in \mathcal{B}} [n - \text{Level}(B)] - \sum_{B \in \mathcal{B}} \sum_{P \in \mathcal{B}} \min(n - \text{Level}(B), n - \text{Level}(P)).$$

Ta nhận được công thức tính tổng số bộ nhớ cần dùng để lưu trữ ảnh trong mã dây chuyền

$$3\Big(4 \sum_{B \in \mathcal{B}} n - \text{Level}(B) - \sum_{B \in \mathcal{B}} \sum_{P \in N_B} \min(n - \text{Level}(B), n - \text{Level}(P))\Big).$$

Đến đây, ta có thể kết luận rằng khi cho một ảnh biểu diễn bởi mã cây tứ phân, mã dây chuyền sẽ tối ưu hơn về mặt lưu trữ khi

$$4 \sum_{B \in \mathcal{B}} [n - \text{Level}(B)] - \sum_{B \in \mathcal{B}} \sum_{P \in N_B} \min(n - \text{Level}(B), n - \text{Level}(P)) < \frac{Q(2 + 2n + \log_2 n)}{3}$$

Nhận ngày 21-4-1986

TÀI LIỆU THAM KHẢO

1 **E.L. Hall**, Computer image processing and recognition. New York — Academic Press 1979.

2. **M. Onoe et al**, Chromosome Analysis by minicomputer. CGIP — Vol 2, 1973, p. 402 — 416.

3. **G.S. Sidhu, R.T.Boute**, Property encoding: Application in binary picture encoding and boundary following. IEEE Trans. Comput. Vol C-21, 1972, p. 1206 - 1216.

(Xem tiếp trang 27)

Tạp chí KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN TẬP II (1986), Số 3 19-27)

SỰ TƯƠNG ĐƯƠNG CỦA CÁC BIỂU MỞ RỘNG ĐƠN

ĐỖ XUÂN THỌ

I – MỞ ĐẦU

Một trong những vấn đề hiện nay được nhiều người quan tâm là tối ưu hóa việc xử lí các câu hỏi trong cơ sở dữ liệu (CSDL) nhằm làm giảm bớt thời gian thực hiện trên máy tính khi xử lí những câu hỏi đó.

Aho A.V., Sagiv Y., Ullman J.D., [1,2,3] đã sử dụng biểu (tableaux) để tối ưu hóa một lớp các câu hỏi gồm 3 phép tính chọn, chiếu và kết nối tự nhiên). Các thủ tục tối ưu được xây dựng trên cơ sở cực tiểu hóa số dòng của biểu mô tả câu hỏi nhằm loại bỏ được các phép kết nối thừa. Đây là việc làm có ý nghĩa do phép kết nối là « đắt nhất » trong 3 phép tính nêu ra ở trên.

Trong [9, 10] tác giả đã nêu ra những hạn chế của khái niệm biểu, trên cơ sở đó đề xuất dựa vào khái niệm biểu mở rộng. Kết quả cho phép mở rộng lớp các câu hỏi có thể sử dụng biểu để tối ưu hóa.

Do bài toán kiểm tra sự tương đương của 2 biểu là bài toán NP – đầy đủ, vì vậy trong [1,2] các tác giả đã đưa ra khái niệm biểu đơn (simple tableaux) tương ứng lớp con các câu hỏi mà trên thực tế không khác biệt với lớp những câu hỏi xét ở trên là bao. Điều có ý nghĩa quan trọng là thủ tục để kiểm tra sự tương đương của các biểu đơn có độ phức tạp về thời gian là đa thức. Trong bài này nghiên cứu sự tương đương của các biểu mở rộng đơn và điều kiện để một biểu mở rộng đơn có biểu thức quan hệ tương ứng.

II – MỘT SỐ KHÁI NIỆM

Các khái niệm cơ bản và các thuật ngữ về CSDL quan hệ có thể xem chi tiết trong [10]. Ở đây chỉ nhắc lại vài khái niệm sẽ dùng cho các phần sau.

Trên cơ sở đại số quan hệ, câu hỏi của người sử dụng có thể mô tả dưới dạng biểu thức quan hệ với những phép tính chọn, chiếu, kết nối....

Nếu A là thuộc tính của quan hệ r, $\mathcal{C} \subseteq \mathrm{dom}(A)$, phép chọn $\sigma_{A \in \mathcal{C}}(r)$ được định nghĩa [10] :

$$\sigma_{A \in \mathcal{C}}(r) = \{\mu \in r \mid \mu[A] \in \mathcal{C}\}.$$

r_1, r_2 là 2 quan hệ có lược đồ tương ứng R_1 và R_2. Phép kết nối tự nhiên của r_1, và r_2, kí hiệu là $r_1 * r_2$:

$$r_1 * r_2 = \{\mu \mid (\mu \text{ là một bộ của quan hệ có lược đồ } R_1 \cup R_2) \wedge ((\exists\, \partial_1 \in r_1) \wedge (\exists\, \partial_2 \in r_2) : \partial_1[R_1] = \mu[R_1],\ \partial_2[R_2] = \mu[R_2])\}.$$

Biểu mở rộng $\mathcal{T}$ là một ma trận 2 chiều có số cột bằng số thuộc tính của quan hệ vũ trụ theo trật tự cố định. $\mathcal{T}$ gồm hàng đầu tiên gọi là giản lược, kí hiệu w_0 và các hàng w_1, w_2,..., w_m. φ là tập các kí hiệu xuất hiện trong biểu.

$$\varphi = \{a_i, b_j, \tilde{\mathcal{C}}_k, \tilde{\mathcal{C}}^*_i, \sqcup\}$$

trong đó a_j là các biến phân biệt, b_j – các biến không phân biệt; $\tilde{\mathcal{C}}$, $\tilde{\mathcal{C}}^*$ – kí hiệu tập những giá trị $\mathcal{C}$ thuộc miền xác định của các thuộc tính. $\tilde{\mathcal{C}}$ được gọi là kí hiệu tập những giá trị phân biệt. $\tilde{\mathcal{C}}^*$ được gọi là kí hiệu không phân biệt. Trên mỗi cột của biểu $\mathcal{T}$ chỉ xuất hiện nhiều nhất một biến phân biệt a_i hoặc một kí hiệu phân biệt $\tilde{\mathcal{C}}_i$.

Một định giá ρ đối với $\mathscr{C}$ là phép thế mỗi kí hiệu trong φ một giá trị tương ứng và thỏa mãn các điều kiện:

a) Nếu $\widetilde{\mathcal{C}}$ là kí hiệu tập giá trị $\mathcal{C}$ thì $\rho(\widetilde{\mathcal{C}}) = c$. Trong trường hợp $\mathcal{C} = \phi$ thì $\rho(\widetilde{\mathcal{C}}) = \phi$

b) Với $w = v_1 v_2 \ldots v_n$ là 1 hàng của $\mathscr{C}$ thì

$$\rho(w) = \rho(v_1)\,\rho(v_2)\ldots\rho(v_n).$$

Nếu I là một thể hiện của quan hệ vũ trụ, khi đó giá trị của biểu $\mathscr{C}$ ở thể hiện I được xác định như sau:

$$\mathscr{C}(I) = \{\rho(w_o) \mid \rho \text{ là một định giá: } \rho(w_j) \in I,\ j = \overline{1,m}\}$$

w_o – giản lược, w_j – các dòng của $\mathscr{C}$.

Trong [9,10] đã nêu qui tắc xây dựng biểu mở rộng $\mathscr{C}$ cho một biểu thức quan hệ thu hẹp E Áp dụng quy tắc đó, chúng ta có thể xây dựng cho mỗi biểu thức quan hệ thu hẹp E một biểu mở rộng tương ứng.

$$V_I(E) = \mathscr{C}(I),$$

$V_I(E)$ – giá trị của biểu thức E với thể hiện I của quan hệ vũ trụ [1,2].

Hai biểu mở rộng $\mathscr{C}_1$ và $\mathscr{C}_2$ là tương đương ($\mathscr{C}_1 \equiv \mathscr{C}_2$), nếu với tất cả thể hiện I của quan hệ vũ trụ:

$$\mathscr{C}_1(I) = \mathscr{C}_2(I)$$

$\mathscr{C}_1$ và $\mathscr{C}_2$ là hai biểu mở rộng. θ là ánh xạ bao hàm nếu ánh xạ các hàng của $\mathscr{C}_1$ vào các hàng của $\mathscr{C}_2$ thỏa những điều kiện sau:

– Nếu w_o là giản lược của $\mathscr{C}_1$ thì $\theta(w_o)$ là giản lược của $\mathscr{C}_2$. θ ánh xạ biến phân biệt sang biến phân biệt, $\widetilde{\mathcal{C}}$ sang $\widetilde{\mathcal{C}}_1$,

$$\mathcal{C}_1 \subseteq \mathcal{C}$$

– θ ánh xạ kí hiệu $\widetilde{\mathcal{C}}^*$ sang $\widetilde{\mathcal{C}}_1$ hoặc $\widetilde{\mathcal{C}}_1^*$: $\mathcal{C}_1 \subseteq \mathcal{C}$. Nếu trong cột A_k trên 2 hàng i và j của $\mathscr{C}_1$ có cùng $\widetilde{\mathcal{C}}^*$ thì trên hàng $\theta(i)$ và $\theta(j)$ của $\mathscr{C}_2$ có cùng kí hiệu.

– Nếu trong cột A_k trên 2 hàng i và j của $\mathscr{C}_1$ có cùng một biến không phân biệt thì trong cột A_k của $\theta(i)$ và $\theta(j)$ có cùng một kí hiệu. Kí hiệu này có thể là biến phân biệt, biến không phân biệt, $\widetilde{\mathcal{C}}$ hoặc $\widetilde{\mathcal{C}}^*$.

$\mathscr{C}_1$ và $\mathscr{C}_2$ có cùng lược đồ quan hệ đích. Điều kiện cần và đủ để $\mathscr{C}_1 = \mathscr{C}_2$ là tồn tại ánh bao hàm từ $\mathscr{C}_1$ vào $\mathscr{C}_2$ và từ $\mathscr{C}_2$ vào $\mathscr{C}_1$ [9, 10].

III – BIỂU MỞ RỘNG ĐƠN, SỰ TƯƠNG ĐƯƠNG CỦA CÁC BIỂU MỞ RỘNG ĐƠN

Biểu mở rộng đơn là biểu mà nếu trong một cột có biến không phân biệt (b) lặp hoặc kí hiệu tập các giá trị không phân biệt ($\widetilde{\mathcal{C}}^*$) lặp thì không có kí hiệu khác lặp trong cột đó Lớp biểu đơn là khá rộng, do đó trong thực tế chúng ta thường ít gặp biểu thức quan hệ tương ứng với biểu là không đơn.

x và w là hai hàng. Nói x phủ w nếu các điều kiện sau đây thỏa:

– x và w có cùng số cột,

– Nếu w có biến a_i (hoặc kí hiệu $\widetilde{\mathcal{C}}$) trong một cột thì x cũng có biến a_i (hoặc trong cột đó.

– Nếu w có kí hiệu $\widetilde{\mathcal{C}}^*$ trong một cột thì x có kí hiệu

$\widetilde{\mathcal{C}}_1$ hoặc $\widetilde{\mathcal{C}}_1^*$ trong cột tương ứng: $\mathcal{C}_1 \subseteq \mathcal{C}$.

S là tập các hàng của $\mathscr{C}$.x phủ S nếu x phủ từng hàng trong S.

Từ khái niệm phủ và ánh xạ bao hàm, chúng ta có nhận xét: Nếu $\mathscr{C}_1$ và $\mathscr{C}_2$ là hai biểu mở rộng đơn có giản lược đồng nhất (chỉ khác biệt bằng sự đổi tên các biến phân biệt hoặc k hiệu tập các giá trị phân biệt) và không có biến không phân biệt hoặc kí hiệu tập các giá trị không phân biệt lặp, khi đó $\mathscr{C}_1 \equiv \mathscr{C}_2$ nếu và chỉ nếu mỗi hàng của $\mathscr{C}_1$ phủ một hàng của $\mathscr{C}_2$ và mỗi hàng của $\mathscr{C}_2$ phủ một hàng của $\mathscr{C}_1$.

$\mathscr{C}$ là biểu mở rộng đơn, S – tập các hàng, w – một hàng của biểu $\mathscr{C}$. Bao đóng của tập S đối với w, kí hiệu $CL_w(S)$ là tập tối thiểu hàng chứa S thỏa những điều kiện sau

Nếu $x_1 \in CLw(S)$ và x_2 là một hàng nào đó của $\mathcal{C}$ và x_1, x_2 có cùng biến không phân biệt trong một cột nào đó nhưng trong cột ấy w có kí hiệu khác hoặc x_1 và x_2 có cùng kí hiệu không phân biệt $\tilde{\mathcal{C}}^*$ còn w có $\tilde{\mathcal{C}}_1^*$, hay $\tilde{\mathcal{C}}_1$ với $\mathcal{C}_1 \subseteq \mathcal{C}$ và $\mathcal{C}_1 \neq \mathcal{C}$ thì $x_2 \in CLw(S)$.

Khái niệm bao đóng được đưa vào với mục đích tìm tập các hàng của biểu $\mathcal{C}$ có khả năng ánh xạ vào w. Ta có định lí sau:

Định lí 1.

$\mathcal{C}$ là biểu mở rộng đơn. CLw(S) – bao đóng của tập hàng S đối với w. Ánh xạ θ được xác định như sau:

$$\theta(x) = \begin{cases} w, & \text{nếu } x \in CLw(S) \\ x, & \text{với các hàng còn lại.} \end{cases}$$

Khi đó θ là ánh xạ bao hàm từ $\mathcal{C}$ vào $\mathcal{C}$ nếu và chỉ nếu w phủ CLw(S).

Chứng minh:

Nếu: Giả thiết w phủ CLw(S). Cần chứng minh rằng $\theta(x)$ là ánh xạ bao hàm. Chứng minh theo phương pháp phản chứng. Giả thiết θ không phải là ánh xạ bao hàm, có nghĩa tồn tại 2 hàng y và z của $\mathcal{C}$ có cùng kí hiệu d trong cột B (d có thể là biến không phân biệt hoặc kí hiệu $\tilde{\mathcal{C}}^*$) và có những kí hiệu khác biệt trên các hàm $\theta(y)$ và $\theta(z)$ tương ứng trong B. Xét các trường hợp sau:

a) Cả y và z không thuộc CLw(S). Khi đó $\theta(y)=y$, $\theta(z)=z$. Điều giả thiết mâu thuẫn ở trên không thể xảy ra.

b) $y \in CLw(S)$ và $z \notin CLw(S)$ hoặc ngược lại.

Rõ ràng d không thể là biến không phân biệt hoặc kí hiệu $\tilde{\mathcal{C}}^*$. Thật vậy:

– Nếu $y[B] = z[B] = b$, $w[B] = d' \neq b$. Do $y \in CLw(S)$ suy ra $z \in CLw(S)$ dẫn đến mâu thuẫn với giả thiết.

– Nếu $y[B] = z[B] = \tilde{\mathcal{C}}^*$. Do w phủ CLw(S) và $y \in CLw(S)$ suy ra $w[B] = \tilde{\mathcal{C}}_1$ hoặc $w[B] = \tilde{\mathcal{C}}_1^*$: $\mathcal{C}_1 \subseteq \mathcal{C}$, $\mathcal{C}_1 \neq \mathcal{C}$. Theo quy tắc xây dựng CLw(S) suy ra $\in CLw(8)$, trái với giả thiết.

Như vậy d chỉ có thể là biến phân biệt (u) hoặc kí hiệu tập các giá trị phân biệt ($\tilde{\mathcal{C}}$). Suy ra $z[B] = y[B] = w[B]$.

c) – Cả y và z đều thuộc CLw (S). Khi đó $\theta(y) = \theta(z) = w$.

Trong trường hợp này không thể có giả thiết mâu thuẫn. θ là ánh xạ bao hàm.

Chỉ nếu: Điều cần chứng minh được suy ra trực tiếp từ định nghĩa của ánh xạ bao hàm và quan hệ phủ.

Định lí đã được chứng minh.

w – chuỗi là một dãy các hàng $z_1, z_2, \ldots, z_k$, $k \geq 1$ sao cho $\forall i$, $1 \leq i < k$, nếu z_i và z_{i+1} có cùng biến không phân biệt trong một cột thì w có kí hiệu khác biệt trong cột đó; nếu z_i và z_{i+1} có cùng kí hiệu $\tilde{\mathcal{C}}^*$ thì w có kí hiệu $\tilde{\mathcal{C}}_1^*$ hoặc $\tilde{\mathcal{C}}_1$: $\mathcal{C}_1 \subseteq \mathcal{C}$, $\mathcal{C}_1 \neq \mathcal{C}$.

Theo định nghĩa của bao đóng $z_k \in CLw(S)$ nếu và chỉ nếu tồn tại w–chuỗi $z_1, z_2, \ldots, z_k$ mà $z_1 \in S$.

Định lý 2.

A và B là 2 cột của biểu mở rộng đơn $\mathcal{C}$ với các biến không phân biệt lặp hoặc $\tilde{\mathcal{C}}^*$ lặp trên tập các hàng S_1 và S_2 tương ứng. Giả thiết x phủ $CL_x(S_1)$, θ là ánh xạ bao hàm từ $CLx(S_1)$ vào x và ánh xạ các hàng khác vào chính nó. $\mathcal{C}'$ là biểu nhận được từ $\mathcal{C}$ sau khi loại bỏ các hàng $CLx(S_1) - \{x\}$. $S_3 = S_2 - (CLx(S_1) - \{x\})$. Khi đó nếu S_3 có số hàng ≥ 2 và trong $\mathcal{C}'$ w phủ $CLw(S_3)$ thì trong $\mathcal{C}$ w phủ $CLw(S_2)$.

Chứng minh:

1. Xét trường hợp $x \notin CLw(S_3)$. Cần chứng minh nếu tồn tại w–chuỗi trong $CLw(S_2)$ thì w–chuỗi đó cũng trong $CLw(S_3)$. Suy ra nếu w phủ $CLw(S_3)$ trong $\mathcal{C}'$ thì w cũng phủ $CLw(S_2)$ trong $\mathcal{C}$. Việc chứng minh được tiến hành quy nạp theo độ dài của w – chuỗi trong $\mathcal{C}$ từ s tới y, $z \in S_2$.

Bước cơ sở: Độ dài của w−chuỗi bằng 1. Do y và z có cùng b hoặc $\tilde{\mathscr{C}}^*$ trong cột A nào đó nên $y \in S_2$.

Giả thiết ngược lại $y \notin S_3$ suy ra $y \in CL_w(S_1)$. Do S_3 có ít nhất 2 hàng nên ta có thể giả thiết $z \in (S_3 - \{x\})$ vì vậy $z \notin CL_x(S_1)$. Khi đó x phải có cùng biến không phân biệt (b) hoặc kí hiệu $\tilde{\mathscr{C}}^*$ trong cột A như trên hàng y và z, vì nếu không $z \in CL_x(S_1)$. Như vậy $x \in S_2$ và $x \in S_3$ dẫn tới mâu thuẫn với giả thiết $x \notin CL_w(S_3)$. Do đó $y \in S_3$ và $y \in CL_w(S_3)$.

Bước quy nạp:

Giả thiết điều cần chứng minh đã đúng với trường hợp w−chuỗi có độ dài k−1. Chứng minh cũng đúng với k. Giả sử w−chuỗi có dạng $z = z_1, z_2, \ldots, z_{k-1}, z_k = y$ trong $\mathscr{C}$ và theo giả thiết quy nạp thì $z_{k-1} \in CL_w(S_3)$ trong $\mathscr{C}'$. Do $x \notin CL_w(S_3)$ nên $x[A] \neq z_k[A]$ vì trong trường hợp ngược lại $x \in CL_w(S_3)$.

$y \notin CL_x(S_1)$. Thật vậy, nếu $y \in CL_x(S_1)$ thì do $y[A] = z_{k-1}[A]$ và x phủ y nên $z_{k-1} \in CL_x(S_1)$. Dẫn đến trái với giả thiết quy nạp là $z_{k-1} \in CL_x(S_3)$. Suy ra $y \in CL_w(S_3)$ và y là một hàng của $\mathscr{C}'$.

2. Xét trường hợp $x \in CL_w(S_3)$ trong $\mathscr{C}'$. θ_2 là ánh xạ bao hàm trong $\mathscr{C}'$; $\theta_2: CL_w(S_3) \to w$ và θ_2 ánh xạ các hàng còn lại của $\mathscr{C}'$ vào chính nó. Khi ấy $\theta_2\theta_1$ ánh xạ $CL_w(S_2)$ vào w Thật vậy

– Nếu $y \in CL_w(S_2)$ và $y \notin CL_x(S_1)$, khi đó $\theta_1: y \to x$ và $\theta_2: x \to w$.

– Nếu $y \in CL_w(S_2)$ và $y \notin CL_x(S_1)$ thì tồn tại w − chuỗi có dạng $z_1, z_2, \ldots, z_n = y$ trong $CL_w(S)$. Tương tự như trên bằng phương pháp quy nạp chúng ta chứng minh được $y \in CL_w(S_3)$ trong $\mathscr{C}$.

Ta đã chứng minh được $\theta_2\theta$; $CL_w(S_2) \to w$ và w phủ $CL_w(S_2)$ trong $\mathscr{C}$.
Sử dụng định lí trên có thể giảm số hàng của biểu mở rộng đơn theo quy tắc sau:
Với mỗi hàng i, j của $\mathscr{C}$ $(i \neq j)$, xác định có tồn tại hay không bao đóng $CL_i(j)$ bị i phủ. Trong trường hợp tồn tại $CL_i(j)$ như vậy, có thể loại bỏ các hàng của $CL_i(j)$ từ $\mathscr{C}$. Kết quả nhận được $\mathscr{C}'$ tương đương với $\mathscr{C}$. Bằng cách đó sẽ đưa biểu mở rộng đơn về dạng rút gọn.
Thủ tục COVERS có chức năng kiểm tra hàng i có phủ j không?

Procedure COVERS (i, j, result) /* i có phủ j ? */

for each column A *do*

if row j has distinguished variable of distinguished symbol in column A, and row has a different symbo in column A *then goto* (2) *else*

if row j has nondistinguished symbol $\tilde{\mathscr{C}}^*$ in column A, and row i has no symbol $\tilde{\mathscr{C}}_1$ or $\tilde{\mathscr{C}}_1^*$: $\mathscr{C}_1 \subseteq \mathscr{C}$ *then*

(2) *return* result ← false

return result ← true

end COVERS

Nếu biểu $\mathscr{C}$ có r hàng, c cột và các tập giá trị $\mathscr{C}_1$ được sắp có lực lượng lớn nhất là n thì thủ tục COVERS có độ phức tạp $O(rc + n^2 \log n)$.

Thủ tục CLOSE có chức năng tìm bao đóng $CL_i(j)$ mà bị hàng i phủ. CLOSE có độ phức tạp $O(r^2c + n^2 \log n)$.

Procedure CLOSE (i, j, NLi (j), result 1)

$k_2 \leftarrow j$

(2) COVERS (i, k_1, result)

(3) *if* result = « true » *then*
begin

(4) add row k_1 to CLi (j)
for each column A *do*
for each row k *do*
for each row l in CLi (j) *do*

(5) *if* (row k and l have the same nondistinguished
variable in A and row i has another symbol in A)
or (row k and l have the same nondist — symbol
in A and row i has $\tilde{\mathcal{C}}_1$ or $\tilde{\mathcal{C}}_1^*$; $\mathcal{C}_1 \subseteq \mathcal{C}$, $\mathcal{C}_1 \neq \mathcal{C}$) *then*
begin
(6) $k_1 \leftarrow k$; *goto* (2)
end
end
return result 1 ← result
end CLOSE

Thủ tục REDUCE có chức năng giảm các hàng của biểu $\mathcal{C}$ nếu điều đó cho phép. REDUCE có độ phức tạp $0(r^4c + r^2n^2\log n)$.

Procedure REDUCE ($\mathcal{C}$)
for each row i of $\mathcal{C}$ *do*
for each row $j \neq i$ of $\mathcal{C}$ *do*
begin
CLOSE (i, j, CLi (j), result 1)
if result 1 = « true » *then*
remove the rows in CLi (j) from $\mathcal{C}$
end
end REDUCE

Dễ dàng chứng minh định lí sau đây ;

Định lí 3. $\mathcal{C}_1$ và $\mathcal{C}_2$ là 2 biểu mở rộng đơn. A là cột với kí hiệu d lặp (d có thể là biến b hoặc kí hiệu $\tilde{\mathcal{C}}^*$) trong tập hàng S_1 của $\mathcal{C}_1$. Trong $\mathcal{C}_1$ không có hàng w nào phủ CLw (S_1). Khi đó : – Nếu $\mathcal{C}_1 \equiv \mathcal{C}_2$ và d là $\tilde{\mathcal{C}}^*$ thì trong cột A tương ứng của $\mathcal{C}_2$ cũng có $\tilde{\mathcal{C}}_1^*$ lặp: $\mathcal{C} = \mathcal{C}_1$. Do đó có thể đồng nhất $\tilde{\mathcal{C}}^*$ với $\tilde{\mathcal{C}}_1^*$.

– Nếu $\mathcal{C}_1 \equiv \mathcal{C}_2$ và d là biến không phân biệt b thì trong cột A của $\mathcal{C}_2$ cũng có biến không phân biệt b_1 lặp. Khi đó có thể thay thế b và b_1 trong $\mathcal{C}_1$, $\mathcal{C}_2$ bằng kí hiệu $\tilde{\mathcal{C}}^*$ mới : $\mathcal{C}$ = dom (A). Sử dụng định lí trên chúng ta có thể xây dựng thủ tục TEST để kiểm tra sự tương đương của 2 biểu đơn $\mathcal{C}_1$ và $\mathcal{C}_2$. Thủ tục TEST có độ phức tạp $o(r^4c + r^2n^2\log n)$.

Procedure TEST ($\mathcal{C}_1$, $\mathcal{C}_2$)
begin
(1) REDUCE ($\mathcal{C}_1$) ;
(2) REDUCE ($\mathcal{C}_2$) ;
(3) *for* each column A of $\mathcal{C}_1$ in which a repeated nondisting.
variable b_1 (or nondistinguished symbol $\tilde{\mathcal{C}}_1^*$) *do*
begin
(4) *if* the column A in $\mathcal{C}_2$ has no repeated nondistinguished
varisble (or nondisting–symbol $\tilde{\mathcal{C}}_2^*$: $\mathcal{C}_1 = \mathcal{C}_2$) *then*
(5) *return* false ; / * $\mathcal{C}_1 \not\equiv \mathcal{C}_2$ * /
(6) let b_2 (or $\mathcal{C}_2^*$) be the repeated nondisting – variable (or repeated nondisting – symbol) in column A of $\mathcal{C}_2$;
(7) make b_1 and b_2 (or $\tilde{\mathcal{C}}_1^*$ and $\tilde{\mathcal{C}}_2^*$) be the same new symbol
$\tilde{\mathcal{C}}$; label rows in which has repeated b or $\tilde{\mathcal{C}}^*$
end
(8) *if* every labeled row of $\mathcal{C}_1$ is covered by a labeled row
– of $\mathcal{C}_2$ and vice versa *then*

(9) *if* every nolabeled row of $\mathcal{T}_1$ is covered by a nolabeled row of $\mathcal{T}_2$ and vice versa *then*

(10) *return true* ; /* $\mathcal{T}_1 \equiv \mathcal{T}_2$ */

(11) *return false* /* $\mathcal{T}_1 \not\equiv \mathcal{T}_2$ */

end TEST

IV – LỚP CÁC BIỂU MỞ RỘNG ĐƠN CÓ BIỂU THỨC QUAN HỆ TƯƠNG ỨNG

4.1. Một số khái niệm về siêu đồ thị

Giả thiết $X = \{x_1, x_2, ..., x_n\}$ tập hữu hạn và $\varepsilon = (E_i \mid i \in \mathcal{J})$ là họ các tập con của X thỏa những điều kiện sau:

1. $E_i \neq \phi$ $(i \in \mathcal{J})$

2. $\bigcup_{i \in \mathcal{J}} E_i = X$

Khi đó cặp $H = (X, \varepsilon)$ được gọi là siêu đồ thị, X là tập các đỉnh, E_i là cạnh của siêu đồ thị.

Giả thiết $A \subseteq X$. $\varepsilon_A = \{E_i \cap A \mid E_i \in \varepsilon ; E_i \cap A \neq \phi$. Khi đó $H_A = (A, \varepsilon_A)$ được gọi là siêu đồ thị con của H.

Cho siêu đồ thị $H = (X, \varepsilon)$; s và t là 2 đỉnh. Đường đi từ s tới t là dãy các cung $E_1, ..., E_k$ thỏa những điều kiện sau:

1. $s \in E_1$

2. $t \in E_k$

3. $E_i \cap E_{i+1} \neq \phi$, $1 \leqslant i < k$.

Khi có đường đi từ đỉnh s tới đỉnh t của siêu đồ thị H, ta nói rằng 2 đỉnh s và t là liên thông. Tương tự, ta nói 2 cung E_1 và E_k là liên thông.

$M \subseteq X$, M là tập đỉnh liên thông của siêu đồ thị nếu $\forall$ s,t a $\in M$, $t \in M$ có đường đi từ s tới t.

Thành phần liên thông của siêu đồ thị H là tập đỉnh (hoặc cung) liên thông lớn nhất. Giả sử H_A là siêu đồ thị con của H. H_A được gọi là siêu đồ thị con liên thông không tầm thường nếu các đỉnh của nó là liên thông và số đỉnh của siêu đồ thị con > 1.

4.2. Lớp các biểu mở rộng đơn có biểu thức quan hệ tương ứng. Không phải bất kỳ một biểu đơn nào cũng có biểu thức quan hệ tương ứng. Điều kiện để lớp biểu mở rộng đơn có biểu thức quan hệ tương ứng được thể hiện qua định lí dưới đây:

Định lí 4. $\mathcal{T}$ là biểu mở rộng đơn với lược đồ R. $G_{\mathcal{T}}$ (W, ε) là siêu đồ thị gán nhãn được xây dựng theo quy tắc sau: W – tập đỉnh, $W = \{w_i \mid w_i$ hàng của $\mathcal{T}$, $i = \overline{1, n}\}$. ε – tập cạnh, $\varepsilon = \{E_x \mid E_x \subseteq W\}$. E_x là cạnh được gán nhãn X. $E_x = \{u, v, .., w\}$. Trong đó u, v, ..., w, x – các hàng của $\mathcal{T}$; $\exists A \in R$: $u[A] = v[A] = ... = w[A] = b$ (hoặc $\mathcal{C}^*$) và $x[A] = a$ hoặc $\mathcal{C}_1^*$. Khi đó biểu $\mathcal{T}$ là tương ứng với một biểu thức quan hệ SPJ, nếu và chỉ nếu không tồn tại siêu đồ thị con H liên thông không tầm thường của $G_{\mathcal{T}}$ mà có cạnh được gán nhãn bởi các nút trong H.

Chứng minh: Chỉ nếu: Giả thiết $\mathcal{T}$ là biểu mở rộng đơn tương ứng với một biểu thức quan hệ SPJ. Cần chứng minh không tồn tại siêu đồ thị con H liên thông không tầm thường. Chứng minh bằng phương pháp phản chứng.

Giả thiết $\exists$ H với các tính chất kể trên và $\exists$ biểu thức quan hệ E: $\mathcal{T}_E = \mathcal{T}$. Xây dựng cây biểu diễn E theo các quy tắc sau: – Dùng những quy tắc biến đổi để gộp dãy phép chọn nhằm thực hiện chúng một lần và đẩy xuống phía lá [6, 7]

$$\sigma_{F_1}(... \sigma_{F_n}(R)...) = \sigma_{F_1 \wedge ... \wedge F_n}(R)$$

$$F_i \text{ có dạng } A_i \in \mathcal{C}_i,\ \mathcal{C} \subseteq \text{dom}(A_i)$$

$$\sigma_F(\pi_{A_1 ... A_m}(E)) = \pi_{A_1 ... A_m}(\sigma_F(E))$$

$\sigma_F(E_1 * E_2) = (\sigma_F(E_1)) * E_2$ nếu những thuộc tính trong F là các thuộc tính của E_1.

$\sigma_F(E_1 * E_2) = \sigma_{F_1}(E_1) * \sigma_{F_2}(E_2)$ nếu $F = F_1 \wedge F_2$ và F_1 chứa những thuộc tính E_1, F_2 chứa thuộc tính E_2.

– Mỗi đỉnh của siêu đồ thị tương ứng với một cây con cơ sở. Mỗi cây con cơ sở chứa một lá tương ứng với lược đồ quan hệ R. Cây con cơ sở–biểu diễn của biểu thức con có dạng tổng quát

$$\Pi X_i(\sigma_{F_{i1} \wedge \ldots F_{in}}(R))$$

X_i gồm những thuộc tính tương ứng với các cột trên hàng i của biểu $\mathcal{C}$ có kí hiệu lặp, biến a_k hoặc kí hiệu $\tilde{\mathcal{C}}$.

Trong cây biểu diễn biểu thức E, gọi z là nút cha thấp nhất của các cây con cơ sở tương ứng với những nút của siêu đồ thị con H. Giả thiết $y_1, y_2, \ldots, y_k$ – các con trực tiếp của z. $T_1, T_2, \ldots, T_k$ – những cây con có gốc là $y_1, y_2, \ldots, y_k$. Theo cách chọn z $\exists$ T_i và T_j chứa các cây con cơ sở tương ứng với các đỉnh u, v thuộc H: u, v thuộc cung E_W có nhãn w. Điều đó có nghĩa $\exists A \in R$, $u[A] = v[A] = b$ hoặc $u[A] = v[A] = \tilde{\mathcal{C}}^*$ và phép chiếu để tạo ra b hoặc $\tilde{\mathcal{C}}^*$ phải nằm trên z (theo cách chọn z). Mặt khác, $w \in H$ nên trong cây biểu diễn biểuthức E cây con cơ sở tương ứng với w cũng nằm trong T_k nào đó sau z. Vì vậy cây con đó cũng chịu tác động của phép chiếu trên. Nhưng theo giả thiết $w[A] = a$ hoặc $w[A] = \tilde{\mathcal{C}}$ dẫn tới mâu thuẫn.

Nếu: Giả thiết không tồn tại siêu đồ thị con liên thông không tầm thường H với các tính chất kể trên, khi đó tồn tại biểu thức quan hệ E tương ứng với $\mathcal{C}$: $\mathcal{C}_E = \mathcal{C}$. Trước hết ta mô tả thủ tục để xây dựng biểu thức quan hệ E. Sau đó chứng minh $\mathcal{C}_E = \mathcal{C}$. Thủ tục gồm 2 phần:

Phần 1: Xây dựng và thao tác với siêu đồ thị $G_{\mathcal{C}}$

Phần 2: Trên cơ sở kết quả của phần 1 tiến hành xây dựng biểu thức quan hệ tương ứng.

Thủ tục được tiến hành đệ quy.

– Xây dựng siêu đồ thị $G_{\mathcal{C}}$ tương ứng với $\mathcal{C}$ theo quy tắc đã nêu ra ở trên.

– Giả sử $G_1^{(1)}, G_2^{(1)}, \ldots, G_k^{(1)}$, $k \geq 2$ là các siêu đồ thị con liên thông của $G_{\mathcal{C}}$. Những siêu đồ thị con này đều thỏa điều kiện của định lí. $\mathcal{C}_1^{(1)}, \mathcal{C}_2^{(1)}, \ldots, \mathcal{C}_k^{(1)}$ là các biểu con của $\mathcal{C}$ tương ứng với những đồ thị con liên thông $G_1^{(1)}, \ldots, G_k^{(1)}$. Giản lược của $\mathcal{C}_j^{(i)}$ được xác định theo quy tắc sau: Với mỗi cột A_k nếu trong biểu $\mathcal{C}_j^{(i)}$ $\exists$ hàng w: $w[A] = a_k$ thì trên giản lược có biến phân biệt a_k; nếu $w[A] = \tilde{\mathcal{C}}$ thì trên giản lược có kí hiệu $\tilde{\mathcal{C}}$. Trong các trường hợp còn lại, trên giản lược dấu trắng.

– Từ $\mathcal{C}_j^{(i)}$ xây dựng $\mathcal{C}_j^{'(i)}$ bằng cách đổi biến không phân biệt b (hoặc kí hiệu $\tilde{\mathcal{C}}^*$) thành biến phân biệt a (hoặc kí hiệu $\tilde{\mathcal{C}}$) nếu nó đồng thời xuất hiện trong cột A của một $\mathcal{C}_k^{(i)}$ khác. Trên giản lược của $\mathcal{C}_j^{'(i)}$ được ghi thêm biến phân biệt a hoặc $\tilde{\mathcal{C}}$ trong cột A tương ứng. Do $\mathcal{C}$ là biểu mở rộng đơn nên điều đó chỉ xảy ra trên mỗi cột với nhiều nhất là một kí hiệu mà ở cột đó không chứa a hoặc $\tilde{\mathcal{C}}$ (theo cách xây dựng $G_{\mathcal{C}}$).

Rõ ràng là các biểu con $\mathcal{C}_1^{'(1)}, \ldots, \mathcal{C}_k^{'(1)}$ cũng là biểu mở rộng đơn và $G(\mathcal{C}_i^{'(1)})$ là những siêu đồ thị con tương ứng.

Quá trình trên sẽ được lặp với các $\mathcal{C}_i^{'(1)}$ mà có số hàng > 1. Kết quả sẽ nhận được các biểu con $\mathcal{C}_i^{(k)}$ chỉ gồm một hàng.

Phần 3 gồm những thủ tục để xây dựng biểu thức quan hệ $E_j^{(k)}$ tương ứng với biểu đơn $\mathscr{C}_i^{(k)}$.

Bước cơ sở là thủ tục để biến đổi biểu đơn $\mathscr{C}_j^{(k)}$ có một hàng. Thủ tục làm việc theo quy tắc:

– Duyệt từng cột của biểu theo hàng. Nếu trong cột Ai nào đó xuất hiện kí hiệu $\tilde{\mathscr{C}}$ hoặc $\mathscr{C}^*$ thì thiết lập phép chọn $\sigma_{A_i} \in \mathscr{C}(R)$: R – lược đồ quan hệ vũ trụ. Kết quả nhận được biểu thức chọn có dạng: $\sigma_{F_1 \cap \ldots \cap F_2}(R)$

– $E_j^{(k)} = \Pi_{X_j^{(k)}} (\sigma_{F_1 \cap \ldots \cap F_1}(R))$

$X_j^{(k)}$ – gồm các thuộc tính tương ứng với những cột của biểu $\mathscr{C}_j'^{(k)}$ trên giản lược có kí hiệu khác «␣».

Theo quy tắc xây dựng trên, dễ dàng chứng minh: $\mathscr{C}_{E_j^{(k)}} = \mathscr{C}_i^{(k)}$.

Giả thiết ở bước thứ (n–1) chúng ta có siêu đồ thị $G_{\mathscr{C}}$ gồm k thành phần liên thông $G_{\mathscr{C}_1}, \ldots, G_{\mathscr{C}_k}$ và đã xây dựng được các biểu thức quan hệ $E_1', \ldots, E_k'$ tương ứng với $\mathscr{C}_1', \ldots, \mathscr{C}_k'$:

$$\mathscr{C}_{E_i'} = \mathscr{C}_i' \qquad i = \overline{1, k}$$

Khi đó biểu thức quan hệ E tương ứng với $\mathscr{C}$:

$$E = \Pi_X (*_i E_i')$$

X là tập thuộc tính tương ứng với các cột trong $\mathscr{C}$ có kí hiệu khác «␣» trên giản lược của $\mathscr{C}$. Rõ ràng $\mathscr{C}_E = \mathscr{C}$.

*
* *

Tác giả xin chân thành cám ơn sự giúp đỡ và những ý kiến góp ý, trao đổi của đồng chí Phó tiến sĩ Lê Tiến Vương.

Ngày nhận bài 1-8-1986

TÀI LIỆU THAM KHẢO

1. Aho A. V., Sagiv Y., Ullman J. D., Equivalences among relational expressions. SIAM J.Comp. 8,2 (1979).

2. Aho A.V., Sagiv Y., Ullman J.D., Efficient optimization of a relational expressions. ACM Trans. Database Syst. 4,4 (Dec, 1979), p. 435 – 454.

3. Sagiv Y. Yannakakis M., Equivalences among relational expressions with the union and difference operators. J.of ACM Vol. 27,N°4, Oct. 1980.

4. Yannakakis M., Papadimitrion C. H., Algebraic Dependencies. J. of Computer and System Sciences 25,2 – 41 (1982).

5. Johnson D. S., Klug A., Testing containment of conjunctive queries under functional and inclusion dependencies.

J. of Comp. and Syst. Sciences 28, 167 — 189 (1984).

6. **Ullman J. D.**, Principles of Database Systems.

Comp. Science Prese 1980.

7. **Talbot S.**, An investigation into logical optimization of relational quary languages

The Computer Jouraal Vol 27, N°4, 1984.

8. **Berge C.**, Graphs and Hypergraphs 1973.

9. **Đỗ Xuân Thọ**, Biểu mở rộng và sự tương dương một lớp các biểu thức quan hệ.

Báo cáo tại Hội nghị toán học toàn quốc lần thứ 3 Hà-nội tháng 7/1985.

10. ДО СУАН ТХО, Эквивалентность одного класса реляционных выражении.

Acta Matomatica Vietnamica N° 1, 1985.

ABSTRACT

Equivalences among simple generalized tableaux

Tableaux are used as a instrument for the optimization of query processing in relational database systems. But it has been proved that the problem of testing the equivalence of two tableaux is a NP - complete one. Therefore the problem which arises is to find such class of tableaux that the procedure for testing the equivalence of which is solvable in polynomial time. The class of simple tableaux possesses such property. The paper investigates the equivalence among simple generalized tableaux and conditions for the existence of corresponding relational expression to a simple generalized tableau.

MỘT CÁCH TIẾP CẬN...

Tiếp theo trang 18)

4. **H. Samet, A. Rosenfeld**, Quadtree structure for region processing. In Proceeding of IJCPR-80, 1980, p. 36 — 41.

5. **L.G. Shapiro**, Data structure for picture processing: A survey. CGIP — Vol 11, 1979, p. 162 — 184.

6. **L.P. Johu, S.S. Iyengar**, Space and Time efficient virtual Quadtree. IEEE Trans. PAMI, Vol PAMI-6, N°2, 1984, p. 244—248.

7. **P.N. Khôi, H. Kiêm**, Code conversion in image processing. Institute of Tech. Cybernetics, Bratislava, Rept. N°14. 1984.

8. **P.N. Khôi**, Image processing based on compact representation. Institute of Tech. Cybernetics, Bratislava, Rept., 1985.

9. **P.N. Khôi**, A data structure for quadtree codes and application. Kozlemények, Budapest. N°32, 198 .

ABSTRACT

An approach to represent image objects by shaps

The paper deals with a special data structure – virtual quadtree – for representing and approximating shaps of image objects. Some operator manipulating on this data structure is proposed and followed by their properties. The last section presents the evaluation of memory space efficiency of quadtree code in relation to run length code and chain code for a given image

THÔNG BÁO NGHIÊN CỨU

VỀ MỘT BỘ CHƯƠNG TRÌNH TIỆN ÍCH XÂY DỰNG TRÊN HỆ PHÁT TRIỂN VI TÍNH FT 68K

NGUYỄN CHÍ CÔNG

MỞ ĐẦU

Một trong các mục tiêu nghiên cứu ứng dụng của chúng tôi là thực hiện những chức năng khai thác và triển khai phần mềm hệ thống của các máy vi tính khác nhau. Bộ chương trình tiện ích giới thiệu ở đây có thể giúp cho những người sử dụng nó can thiệp được sâu hơn vào nhiều hệ thống quen thuộc hoặc mới lạ và xây dựng cùng một chương trình trên những máy không cùng họ mà vẫn dễ dàng chuyển đổi. Việc đó cũng tiện lợi cho sự hợp tác giữa các đơn vị nghiên cứu hoặc áp dụng vi tin học của chúng ta nhằm tránh sự lãng phí phần mềm và thời gian thảo chương trùng lặp.

MÔ TẢ

Bộ chương trình khai thác và triển khai phần mềm hệ thống được thực hiện và cài đặt trên hệ phát triển vi tính FT 68K [1] trong năm 1985. Nó có thể dễ dàng sửa lại và cài đặt sang các máy khác khi cần thiết. Bộ chương trình tiện ích này gồm có ba bộ phận chính sau đây:

– Chương trình tạo khuôn FORMAT đa năng,

– Chương trình sao chép và sửa đổi đĩa mềm,

– Chương trình mô phỏng các hệ điều hành vi tính.

1. Chương trình FORMAT đa năng

Sử dụng nó có thể tạo khuôn cho tất cả các loại đĩa mềm 5" 1/4 và 8" có soft sectoring với mật độ đơn hoặc kép, rãnh đơn hoặc kép và trên một mặt hoặc cả hai đều được. Chương trình này còn cho phép chọn độ dài của các rãnh và các sector theo nhiều tiêu chuẩn khác nhau hiện hành, theo chế độ sang rãnh hoặc sang cylindre. Việc sử dụng là thuận tiện nhờ các đối thoại người–máy hiện trên màn hình và việc báo lỗi cũng được tự động hóa.

2. Chương trình sao chép và sửa đổi đĩa mềm

Chương trình này cho phép sao chép nguyên mẫu hoặc sao chép có sửa đổi tham số và dữ liệu trên các loại đĩa mềm. Hệ FT 68K có thể đọc được các đĩa mẫu và sao chép linh hoạt một rãnh hay nhiều rãnh đồng thời trên cùng một ổ đĩa hay trên nhiều ổ khác nhau. Đây là một thuận lợi lớn giúp chúng ta tìm hiểu và khai thác một hệ thống điều hành mới lạ ngay cả khi chưa có máy mẫu. Một ưu điểm nữa là chương trình này có thể sao chép và sửa đổi từ một sector đến mọi sector trong cùng một rãnh của đĩa mềm. Chúng ta có thể dễ dàng sửa đổi một số hệ điều hành vi tính để cài đặt sang một hệ máy khác họ. Những sửa đổi này tập trung chủ yếu vào việc thiết lập sự tương hợp giữa các khối trong hệ điều hành, do đó chương trình đã làm tăng tính mềm dẻo của hệ thống.

3. Chương trình mô phỏng các hệ điều hành vi tính

Đây là một chương trình tiện ích tạo điều kiện cho ta làm việc với các hệ điều hành thuộc những họ vi tính quen biết. Mục lục tạm thời hiện nay là như sau:

1 Họ CPM 68K FT 68K)
2 Họ máy VT (82, 83)
3 Họ máy IBM PC (XT, AT)
4 Họ máy PKD (80,82)
5 Họ máy OSBORNE−1
6 Họ máy APPLE−II.

Thông qua đơn vị bàn phím và màn hình ta có thể chọn một trong sáu khả năng trên và ổ đĩa B của hệ FT 68K được coi như là một máy ảo tương hợp với hệ điều hành vừa chọn. Sự mô phỏng hiện đang còn ở mức độ sử dụng những lệnh thường trú phổ biến như:

– DIR: xem danh mục các tập trên đĩa
– REN: thay đổi tên cho một tập
– AVB: Cất giữ một tập lên đĩa
– TYPE: đọc một tập vào bộ nhớ trong
– TYPED: đọc một tập vào bộ nhớ trong và hiện lên màn hình.

Dùng hệ phát triển vi tính FT 68 K có chương trình mô phỏng các hệ điều hành khác nhau còn cho phép ta tạo ra một thiết bị vào dữ liệu đa năng với các biểu format phong phú khác nhau và để sử dụng trên nhiều máy tính khác nhau. Chọn một trong sáu họ trên cũng có thể vào dữ liệu cho các họ máy còn lại, và ngược lại khả năng mở rộng sang các họ mới cũng không có gì khó thực hiện.

Kết luận: Trên cơ sở hệ FT 68K với bộ chương trình vừa nói trên đây còn có thể tiếp tục xây dựng các chức năng khác, phục vụ cho nghiên cứu những bộ vi xử lý mới hoặc những hệ điều hành vi tính khác. Việc này cũng không ngoài mục đích thử thách và lựa chọn những họ vi mạch hoặc những chương trình thích hợp vào từng trường hợp ứng dụng cụ thể do thực tế đặt ra cần giải quyết có hiệu quả nhanh. Tuy còn đang viết ở hợp ngữ Motorola 68000 và chạy dưới CP/M, kiểu chương trình này có khả năng chuyển sang các họ khác và bổ xung nhiều chi tiết tỉ mỉ hơn.

TÀI LIỆU THAM KHẢO

1. Nguyễn Chí Công, Trần Xuân Thuận, Hệ phát triển vi tính FT 68K, Tạp chí Khoa học tính toán và điều khiển, tập I, số 3, 1985.

ABSTRACT

On an Utility Realized on the FT 68K

On the Microcomputer Development System FT 68K has been written and implemented a set of Universal Floppy Disk Utilities, of which three programs are discussed here:

–FORMAT for all types of diameters 5" $\frac{1}{4}$ and 8", single – sided or double – sided, with single or donble or quad density, soft sectoring by track or by cylinder.

– COPY and MODIFY: neads and writes with different formats on a track or more, of a dive or two ones, from a sector to all sectros of the same track.

– DOS SIMULATION: the dive B in this mode is a vintual machine with the residen commands of any DOS selected among the CP/M 68K, CP/M80, PC – DOS, Apple DOS etc.

TIN TỨC HOẠT ĐỘNG

Hội thảo: CƠ SỞ TOÁN HỌC CỦA TIN HỌC VÀ BẢO VỆ TIN

Từ ngày 4 đến 6 tháng 7 năm 1986 tại Hà Nội, Viện toán học Viện khoa học Việt Nam đã phối hợp với một số trường đại học và cơ quan nghiên cứu tổ chức hội thảo khoa học « Cơ sở toán học của tin học và bảo vệ tin ».

Gần 100 đại biểu đã nghe và thảo luận 8 báo cáo tổng quan và 25 báo cáo kết quả nghiên cứu thuộc các lĩnh vực:

– Ôtomat và ngôn ngữ hình thức

– Nhận dạng

– Cơ sở dữ liệu và lập trình

– Độ phức tạp tính toán

– Các vấn đề tổ hợp.

Hội nghị đã dành một buổi mạn đàm trao đổi

THÔNG TIN KHOA HỌC – CÁC VẤN ĐỀ CÒN MỞ

Các báo cáo tổng quan:

– Đỗ Long Vân (Viện toán học, Hà Nội): Dáng điệu vô hạn của ôtômat hữu hạn và một số vấn đề có liên quan.

– Nguyễn Xuân My (Đại học kỹ thuật quân sự, Vĩnh Phú): Một số hướng nghiên cứu của lý thuyết ngôn ngữ hình thức.

– Nguyễn quốc Toản (Đại học tổng hợp Hà Nội): Về các cấu trúc dữ liệu.

– Lê Công Thành (Viện toán học, Hà Nội), Nguyễn Hữu Ngự (Đại học tổng hợp Hà Nội): Về lý thuyết độ phức tạp tính toán.

– Bạch Hưng Khang, Hoàng Kiếm (Viện khoa học tính toán và điều khiển, Hà Nội): Một số vấn đề toán học của trí tuệ nhân tạo.

– Phạm Trà Ân (Viện toán học, Hà Nội): Một số vấn đề toán học của mạng Petri.

– Vũ Duy Mẫn (Viện khoa học tính toán và điều khiển, Hà Nội): Đồng bộ hóa các quá trình tương tranh.

– Vũ Đình Hòa (Viện khoa học tính toán và điều khiển, Hà Nội): Về bài toán của Hamilton

25 báo cáo còn lại đều đề cập đến những kết quả nghiên cứu gần đây nhất của hầu hết các tác giả trẻ đang độ sung sức.

NGUYỄN XUÂN HUY

GIỚI THIỆU SÁCH

N.Wirth. CẤU TRÚC DỮ LIỆU + GIẢI THUẬT = CHƯƠNG TRÌNH

Nguyễn Văn Ba, Nguyễn Văn Lư, Vũ Duy Mẫn, Hồ Thuần dịch.

Hồ Thuần hiệu đính và giới thiệu. NXB THỐNG KÊ Hà Nội, 1982, 500tr.

Nguyên bản: N.Wirth. Algorithms + Data structures = Programs.

Đây là một trong những cuốn sách hay về lập trình hiện có trên thế giới. Tác giả của cuốn sách này—HiklauxWirth—là người khởi thảo và trực tiếp phát triển ngôn ngữ lập trình PASCAL, kể từ những phương án ban đầu cho tới những phương án hiện đại như PASCAL MODULA—2. Hơn thế nữa, cùng với Dijkstra, Hoare, Kauth, v.v... những nhà lý luận về lập trình có tên tuổi. N.Wirth đã tham gia một cách tích cực và sáng tạo trong cuộc «cách mạng lập trình» diễn ra từ những năm 60 của thế kỷ.

Nội dung của cuốn sách minh họa một cách tài tình, đầy sức thuyết phục quan niệm chương trình là những mô tả cụ thể của các giải thuật trừu tượng dựa trên những biểu diễn và cấu trúc dữ liệu đặc biệt, như công thức ngắn gọn A + DS = P được dùng để đặt tên cho cuốn sách. Các tư tưởng về lập trình có cấu trúc, lập trình hệ thống, theo cách tiếp cận từ trên xuống được thể hiện một cách sáng sủa. Kỹ thuật «tinh dần từng bước» được tác giả minh họa thông qua việc chi tiết hóa từng bước biểu diễn của dữ liệu song song với sự chi tiết hóa của giải thuật để dần càng thích nghi với các khả năng cho phép của một hệ lập trình sẵn có.

Trong chương thứ nhất tác giả giới thiệu cách kiến trúc nên các cấu trúc dữ liệu phức tạp từ những cấu trúc dữ liệu nguyên thủy và những khuôn mẫu mực thước của những chương trình dùng để xử lý các cấu trúc dữ liệu tương ứng.

Chương thứ hai đề cập đến những giải thuật sắp xếp một công cụ lý tưởng để thể hiện nhiều nguyên lý của lập trình và những tình huống xuất hiện trong nhiều ứng dụng khác nhau. Như tác giả nhận xét «Thường ta có thể xây dựng toàn bộ một khóa giảng về lập trình từ những thí dụ về vấn đề sắp xếp».

Chương thứ ba xét các giải thuật đệ qui và chứng tỏ đệ qui là một sự tổng quát hóa của lặp. Các thí dụ nêu trong chương này và các chương tiếp theo mang một ý nghĩa quan trọng về phương pháp luận cho việc thiết kế các chương trình lớn và phức tạp như chương trình dịch, các hệ cơ sở dữ liệu, các hệ trí tuệ nhân tạo.

Chương thứ tư dành cho các kỹ thuật mô tả, tổ chức và xử lý các cấu trúc dữ liệu động.

Chương thứ năm—cấu trúc của ngôn ngữ và trình dịch có thể xem như là một minh họa toàn cảnh cho những vấn đề cốt yếu trong lập trình thông qua việc xây dựng một chương trình dịch cụ thể.

Cần nói thêm rằng trước N.Wirth chưa một tác giả nào có thể diễn đạt lại một cách minh bạch, chặt chẽ chương trình dịch do chính mình xây dựng.

N.Wirth. LẬP CHƯƠNG TRÌNH HỆ THỐNG

Phạm Ngọc Khôi dịch. NXB KHOA HỌC VÀ KỸ THUẬT, Hà Nội, 1986, 182 tr.

Nguyên bản: N.Wirth. Systematic Programming (An Introduction).

Tác giả trình bày dưới dạng dễ hiểu và súc tích những kiến thức tối thiểu mà bất kỳ người lập trình nào cũng cần vũ trang cho mình: Cấu trúc máy tính, ngôn ngữ máy, các ngôn ngữ lập trình, cấu trúc dữ liệu, các kỹ thuật xây dựng và triển khai một chương trình. Một số vấn đề được thể hiện với những thao tác chi tiết và đặc biệt tiện lợi như chứng minh chương trình, ý nghĩa và cách vận dụng chú thích trong một chương trình, các thao tác trên tập tuần tự, các phép chuyển đổi giữa các cách thể hiện của dữ liệu. Đặc biệt, tác giả nêu nhiều thí dụ xử lý dữ liệu dạng số và dạng văn bản: giải hệ phương trình tuyến tính, tìm ước chung lớn nhất của hai số tự nhiên, tìm số nguyên tố đầu tiên, điều chỉnh khuôn dạng dữ liệu, xác định từ chân chính, các từ chính quy.

Khái niệm về chương trình con và các cách truyền tham trị được trình bày cặn kẽ. Các kỹ thuật lập trình cao cấp như tinh dần từng bước, thử và nghiệm, giải thuật quay lui được phân tích sâu sắc.

Cuốn sách cung cấp cho bạn đọc nhiều tư liệu kinh điển quý báu như hệ tiên đề cho số học trong máy tính (tiên đề về kiểu dữ liệu), hệ tiên đề chứng minh chương trình của Hoare, v.v...

Cùng với cuốn «Cấu trúc dữ liệu + giải thuật = chương trình», cuốn «Lập trình hệ thống» của cùng tác giả lập thành bộ cẩm nang tin cậy cho những người lập trình.

Nội dung của hai cuốn sách trên không chỉ bó hẹp trong việc truyền bá ngôn ngữ lập trình PASCAL. Những tư tưởng và triết lý nêu trong hai cuốn đó xứng đáng giữ vai trò chỉ đạo cho những bạn đọc nào muốn trang bị một học vấn lập trình cho bản thân.

Do những đóng góp có giá trị trong việc phát triển tin học, tác giả của cuốn sách — Niklaux Wirth đã được tặng giải thưởng Turing năm 1984.

C.J.Date. NHẬP MÔN CÁC HỆ CƠ SỞ DỮ LIỆU

Hồ Thuần, Nguyễn Quang Vinh, Nguyễn Xuân Huy dịch.

Hồ Thuần hiệu đính và giới thiệu. NXB THỐNG KÊ Hà Nội, 1986. Tập I: 300 tr, Tập II: 368 tr.

Nguyên bản: C.J.Date. An Introduction to Databese Systems (second edition).

Đây là cuốn sách nằm trong lô sách «Lập trình hệ thống» của công ty IBM (thuộc chương trình mức 2).

Nội dung của cuốn sách đáp ứng được nhu cầu cấp thiết của đông đảo bạn đọc: các nhà nghiên cứu tin học, các kỹ sư có liên quan tới việc ứng dụng tin học, các giảng viên, nghiên cứu sinh và sinh viên các trường đại học và cao đẳng kỹ thuật muốn tìm hiểu một cách nghiêm túc và có hệ thống lý thuyết các hệ cơ sở dữ liệu để trên cơ sở đó tích lũy khả năng thiết kế, cài đặt và sử dụng chúng một cách có hiệu quả cho công tác của ngành mình.

Cơ sở dữ liệu là một thành phần hữu cơ nhất thiết phải có trong hầu hết các hệ tin học. Chỉ riêng việc bảo trì một cơ sở dữ liệu cỡ vừa, chừng dăm nghìn bản ghi, cũng nảy sinh nhiều vấn đề mà nếu người dùng không nắm được, ở một mức độ nào đó, thì chắc chắn sẽ khó có thể đảm bảo được chất lượng của dữ liệu đang quản lý trong tay. Chất lượng của dữ liệu trong một cơ sở dữ liệu được thể hiện qua các tính chất tác động trực tiếp hoặc gián tiếp lên các tập dữ liệu trong quá trình khai thác chúng như tính toàn vẹn dữ liệu, tính an toàn dữ liệu, tính bảo mật, tính không phụ thuộc, v.v.... Những tính chất đó được đảm bảo nhờ những công cụ toán học khá phức tạp như lý thuyết về các phụ thuộc hàm, các dạng chuẩn, logic toán, lý thuyết đồ thị. v.v... Những vấn đề cấp thiết nói trên được tác giả minh chứng một cách sáng sủa, dễ hiểu với nhiều mức độ khác nhau.

Cuốn sách đề cập đến các khía cạnh lý thuyết, thiết kế, cài đặt và vận hành các hệ cơ sở dữ liệu thuộc 3 loại hình — 3 cách tiếp cận: mạng, phân cấp và quan hệ. Với mỗi cách tiếp cận tác giả minh họa qua một hệ cơ sở dữ liệu cụ thể phổ dụng nhất (xét tại thời điểm viết sách: 1974—1977) trong lĩnh vực tin học ứng dụng. Phản ánh đúng vận động của lĩnh vực cơ sở dữ liệu, tác giả đã phân tích khá tỉ mỉ 3 mô hình ứng với 3 cách tiếp cận nói trên và giới thiệu khá chi tiết nội dung của nhiều bài báo quan trọng. Chỉ tính riêng phần thư mục của cuốn sách này ta đã thấy một nội dung khá trọn vẹn cho những bạn đọc nào muốn tìm hiểu sâu sắc lý thuyết và các kỹ thuật cài đặt một hệ cơ sở dữ liệu.

Cuốn sách đặc biệt bổ ích trong giai đoạn hiện nay, khi mà nhiều hệ cơ sở dữ liệu lớn và nhỏ cho các loại máy tính micro, mini, macro do chúng ta nhập từ nước ngoài vào hoặc tự thiết kế và cài đặt đang được sử dụng rộng rãi trong nghiên cứu khoa học và quản lý kinh tế.

N.X.H.

MỤC LỤC

(СОРДЕЖЕНИЕ – CONTENTS)

MỤC LỤC TẬP I

Người gửi đăng bài cần chú ý

1. Bài gửi đăng phải viết rõ ràng, có đánh số từng trang. Nếu là công trình nghiên cứu, không dài quá 8 trang viết tay, có phần tóm tắt nội dung không dài quá nửa trang bằng một trong hai thứ tiếng: Nga, Anh. Nếu là bài giới thiệu tổng quan, dài không quá 16 trang viết tay. Dịch đầu đề các bài ra tiếng Nga và tiếng Anh.

2. Các công thức, kí hiệu trong bài cần viết ngay ngắn, chân phương, đúng vị trí và tỉ lệ to nhỏ. Số hiệu đánh số các công thức thống nhất ghi ở cuối dòng bên phải các công thức.

3. Các hình nên vẽ riêng trên giấy trắng. Trong bài có ghi chỗ đặt hình và chú thích cho từng hình.

4. Những từ dùng nên thống nhất với các quyển từ điển toán học do Ủy ban Khoa học và kỹ thuật nhà nước xuất bản. Những từ mới cần được ghi chú thích dưới trang viết các từ tương ứng với chúng bằng tiếng nước ngoài.

5. Tên tác giả, nhà xuất bản, địa danh thuộc hệ chữ La tinh thì dùng nguyên gốc, nếu thuộc hệ chữ Nga thì chuyển sang hệ chữ Việt như sau: (А) → A, (Б) → B, (В) → V, (Г) → G, (Д) → Đ, (Е, Э) → E, (Ё) → YO, (Ж) → J, (З) → Z, (И, Й) → I, (К) → K, (Л) → L, (М) → M, (Н) → N, (О) → O, (П) → P, (Р) → R, (С) → X, (Т) → T, (У) → U, (Я) → YA, (Ю) → YU, (Ы) → Ư, (Ц) → TX, (Ш, Щ) → S, (Ч) → CH, (Ф) → F, (Х) → KH. Bỏ các dấu Ъ, Ь.

6. Tài liệu tham khảo và trích dẫn ghi theo thứ tự sau: Nếu là bài báo: Tên tác giả, tên bài báo, tên báo, tập, số, năm xuất bản, từ trang ... đến trang. Nếu là sách: Tên tác giả, tên sách, nhà xuất bản, nơi xuất bản, năm xuất bản, từ trang... đến trang.

7. Bài gửi phải có ghi họ tên, địa chỉ cơ quan và số điện thoại liên lạc với tác giả.

8. Bài gửi đăng và thư gửi Tòa soạn gửi theo địa chỉ: Tạp chí Khoa học tính toán và điều khiển, 70 Trần Hưng Đạo, Hà Nội. Dây nói: 52825.

9. Bài không đăng không trả lại bản thảo.

TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN

In 500 cuốn, khổ 19 × 26,5. Số in 109/T2 tại Nhà máy in Diên Hồng, Hà Nội. In xong và nộp lưu chiểu tháng 3-1987.

Giá 5,00đ