Tạp chí Tin học và Điều khiển học, T.17, S.4 (2001), 17-22

VỀ MỘT LỚP CÔNG THỨC LOGIC SUY DẪN

NGUYỄN XUÂN HUY, ĐÀM GIA MẠNH,
VŨ THỊ THANH XUÂN, KIM LAN HƯƠNG

Abstract. This paper refers to the representation of a set of inference formulas by their truth value table. A necessary and sufficient conditions for representing a given logical formula by a set of inference formulas are proved. An algorithm for finding a set of inference formulas by their truth value is presented. Some applications of inference formulas to Armstrong's relations are discussed.

Tóm tắt. Bài viết đề cập đến một lớp công thức logic suy dẫn dạng $X \to Y$, trong đó X và Y là các tích logic của hữu hạn biến. Trong phần thứ hai của bài phát biểu và chứng minh điều kiện cần và đủ để một công thức logic có thể biểu diễn dưới dạng hội của các công thức suy dẫn. Phần này cũng trình bày và đánh giá thuật toán xây dựng hội suy dẫn theo bảng giá trị cho trước. Phần thứ ba mô tả vài ứng dụng lớp các công thức suy dẫn để khảo sát các quan hệ Amstrong trong lý thuyết cơ sở dữ liệu.

1. MỞ ĐẦU

Các công thức logic suy dẫn dạng $X \to Y$, trong đó X và Y là các tích logic của hữu hạn biến được sử dụng khá rộng rãi trong tin học. Chúng đóng vai trò chủ yếu trong các motor suy diễn của các hệ chuyên gia, trong việc thể hiện các ràng buộc dữ liệu của các cơ sở dữ liệu cũng như trong các thuật toán trích chọn luật từ cơ sở dữ liệu thuộc lĩnh vực khai thác tri thức. Sagiv et al. đã chứng minh sự tương đương giữa các phụ thuộc hàm trong cơ sở dữ liệu quan hệ với một tập các công thức của đại số mệnh đề [9]. Berman đã chỉ ra rằng chỉ có các công thức dương mới bảo toàn sự tương đương giữa ba loại hình suy dẫn: suy dẫn theo logic, suy dẫn theo mọi quan hệ và duy dẫn theo các quan hệ hai bộ [2]. Nhóm nghiên cứu của Berman cũng đề xuất và phát triển khái niệm về các phụ thuộc Boolean dương [2]. Nguyễn Xuân Huy và Lê Thị Thanh đã khảo sát lớp các phụ thuộc Boolean dương tổng quát theo nghĩa mở rộng phép so sánh giữa các trị trong mỗi thuộc tính và chỉ ra rằng với các phụ thuộc loại này thì định lý tương đương vẫn bảo toàn [10]. Theo tiếp cận đại số và logic một số tác giả khác cũng nhận được những kết quả thú vị về biểu diễn khóa, phản khóa và các tập đóng trong lý thuyết cơ sở dữ liệu và các hệ thống suy dẫn [4, 5, 6, 9]. Phần thứ hai của bài viết phát biểu và chứng minhh điều kiện cần và đủ để một công thức logic có thể biểu diễn dưới dạng hội của các công thức suy dẫn. Phần này cũng trình bày và đánh giá thuật toán xây dựng hội suy dẫn theo bảng giá trị cho trước. Phần thứ ba chỉ ra một vài ứng dụng lớp các công thức suy dẫn để khảo sát các quan hệ Amstrong [1, 8, 12] trong lý thuyết cơ sở dữ liệu.

Trong bài này khái niệm cơ bản về logic được tham khảo trong [7], về cơ sở dữ liệu được tham khảo trong [1, 3, 8, 12].

2. CÁC CÔNG THỨC LOGIC SUY DẪN

Cho $U = \{x_1, \dots, x_n\}$ là tập các biến logic biến thiên trong miền $B = \{0, 1\}$. Mỗi vectơ các phần tử 0/1 $v = (v_1, \dots, v_n)$ trong không gian B^n được gọi là một *phép gán trị*. Khi đó với mỗi công thức logic f trên U ta có $f(v) = f(v_1, \dots, v_n)$ là *trị chân lý* của công thức f đối với phép gán giá trị v. Cho v là một phép gán giá trị, nếu x là một biến trong U ta kí hiệu $v(x)$ là trị (0 hoặc 1) gán cho biến x trong v. Tương tự, với tập biến X trong U, kí hiệu $v(X)$ là giới hạn của phép gán giá trị v trên tập biến X, $v(X) = \{x : v(x) \mid x \in X\}$, trong đó cặp $x : v(x)$ cho biết cụ thể giá trị $v(x)$ ứng với biến x. Kí hiệu $E(v)$ là tập các biến trong U tại đó x nhận giá trị 1, $E(v) = \{x \in X \mid v(x) = 1\}$. Về bản chất, toán tử E cho phép ta diễn tả các khái niệm logic thông qua các khái niệm của lý thuyết tập hợp.

Thí dụ, với $U = \{x_1, x_2, x_3\}$, $v = (1, 0, 1)$, $X = \{x_1, x_2\}$, ta có:

$$v(x_1) = (1),\ v(X) = \{x_1 : 1,\ x_2 : 0\},\ E(v) = \{x_1,\ x_3\}.$$

Với mỗi tập $X = \{x_{i1}, \dots, x_{ik}\} \subseteq U$, ta qui ước viết tích logic ($\wedge$) của các biến trong X như một dãy kí hiệu của X : $X = x_{i1} \dots x_{ik} = x_{i1} \wedge \dots \wedge x_{ik}$. Như vậy, trong trường hợp không gây ra nhầm lẫn, kí hiệu X vừa biểu thị một tập biến logic trong U vừa biểu thị công thức logic được lập bởi tích logic các biến trong X. Khi đó, nếu coi X là một tích logic thì với mỗi phép gán trị v, ta có $X(v) = 1$ *khi và chỉ khi* $E(v) \supseteq X$.

Ta gọi công thức $f : X \to Y$ là một *công thức suy dẫn* (ctsd) trên tập biến U. Với hệ thống kí hiệu trên, nếu $X = \{x_{i1}, x_{i2}, \dots, x_{ik}\}$ và $Y = \{x_{j1}, x_{j2}, \dots, x_{jm}\}$, f sẽ có dạng tường minh $x_{i1} \wedge x_{i2} \wedge \dots \wedge x_{ik} \to x_{j1} \wedge x_{j2} \wedge \dots \wedge x_{jm}$. Cho $f : X \to Y$ là một ctsd trên tập biến U, với mỗi phép gán trị v, ta có $f(v) = 1$ *khi và chỉ khi* $(E(v) \supseteq X) \Rightarrow (E(v) \supseteq Y)$.

Kí hiệu $I(U)$ là tập các ctsd trên tập biến U.

Ta quan tâm hai phép gán đặc biệt là *phép gán trị đơn vị*, $e = (1, 1, \dots, 1)$ và *phép gán trị không*, $z = (0, 0, \dots, 0)$. Với mọi ctsd $f \in I(U)$ ta có $f(e) = 1 \to 1 = 1$ và $f(z) = 0 \to 0 = 1$.

Với mỗi tập hữu hạn các công thức logic trên U, $F = \{f_1, f_2, \dots, f_m\}$ ta xem F như là một công thức dạng $F = f_1 \wedge f_2 \wedge \dots \wedge f_m$ và gọi là *hội suy dẫn* (hsd). Khi đó với mỗi phép gán trị v, giá trị chân lý của hsd F sẽ được tính là $F(v) = f_1(v) \wedge f_2(v) \wedge \dots \wedge f_m(v)$.

Với mỗi công thức f trên U, bảng chân lý của f, kí hiệu là T_f là tập các phép gán v sao cho $f(v)$ nhận giá trị 1, $T_f = \{v \in B^n \mid f(v) = 1\}$. Khi đó bảng chân lý T_F của hsd F trên U, chính là giao của các bảng chân lý của mỗi công thức thành viên trong F. Ta có $v \in T_F$ *khi và chỉ khi* $(\forall f \in F) : (F(v) = 1)$.

Cho V là tập các phép gán trị trên U. Với hai phần tử $u, v \in V$ ta xét phép toán nhân kí hiệu là $u * v$ như là phép nhân logic trên các thành phần tương ứng của u và v. Cụ thể là nếu

$$u = (u_1, u_2, \dots, u_n) \text{ và } v = (v_1, v_2, \dots, v_n) \text{ thì } u * v = (u_1 \wedge v_1,\ u_2 \wedge v_2,\ \dots,\ u_n \wedge v_n).$$

Thí dụ, với $u = (1, 1, 0, 1)$ và $v = (1, 0, 0, 1)$, ta có $u * v = (1, 0, 0, 1)$.

Ta qui ước tích của tập rỗng các phần tử trong V chính là phép gán trị đơn vị $e = (1, \dots, 1)$.

Tập các phép gán trị V được gọi là đóng đối với phép nhân $*$ nếu V chứa tích của mọi cặp phần tử trong V, tức là $(\forall u, v \in V) : (u * v \in V)$.

Dễ thấy $E(u * v) = E(u) \cap E(v)$.

Các công thức logic f thỏa tính chất $f(x) = 1$ được gọi là các công thức dương. Post [7] đã chứng minh rằng mọi công thức dương đều có thể biểu diễn thông qua các phép toán $\wedge$, $\vee$, $\to$ và hằng 1. Ta cũng biết mọi công thức logic đều có thể biểu diễn dưới dạng chuẩn tuyển (hội) [7]. Nói cách khác, mỗi bảng $T \subseteq B^n$ đều ứng với một công thức logic dạng chuẩn tuyển (hội). Vấn đề biểu diễn một công thức logic qua một tập các phép toán và hằng logic cho trước chưa có lời giải tổng quát [7]. Các phần trình bày dưới đây liên quan đến bài toán sau:

Bài toán 2.1. *Xác định điều kiện cần và đủ để có thể biểu diễn một công thức logic dưới dạng hội suy dẫn.*

Định lý 2.1. *Với mỗi ctsd $f \in I(U)$, T_f chứa các phép gán trị đơn vị e, không z và đóng với phép nhân $*$.*

Chứng minh. Cho ctsd $f : X \to Y$. Dễ thấy $f(e) = f(z) = 1$, do đó $e, z \in T_f$. Giả sử $u, v \in T_f$. Đặt $t = u * v$, ta sẽ chứng minh $t \in T_f$. Giả sử $E(t) \supseteq X$. Vì $E(t) = E(u * v) = E(u) \cap E(v)$ nên $E(u) \supseteq X$ và $E(v) \supseteq X$. Vì $f(u) = f(v) = 1$ nên $E(u) \supseteq Y$ và $E(v) \supseteq Y$ và do đó $E(t) = E(u) \cap E(v) \supseteq Y$. Vậy $f(t) = 1$, và do đó $t \in T_f$. Định lý được chứng minh.

Với mỗi hsd F trong $I(U)$, vì bảng chân lý T_f của F là giao của các bảng chân lý của các công thức thành viên nên ta có hệ quả sau đây.

Hệ quả 2.1. *Với mỗi hsd F trong $I(U), T_F$ chứa các phép gán trị đơn vị e, không z và đóng với phép nhân $*$.*

Bài toán 2.2. *Cho bảng T trên tập biến U, T chứa các phép gán trị đơn vị e, không z và đóng với phép $*$. Hãy xây dựng hsd F trên U nhận T làm bảng chân lý.*

Thuật toán DF dưới đây giải Bài toán 2.2.

Algorithm DF

Input Bảng $T \subseteq B^n$ đóng với phép $*$, chứa e và z..

Output Hội suy dẫn F trên U thỏa tính chất $T_F = T$.

Method

$F := \emptyset$;

for each u *in* $B^n \setminus T$ *do*

$X := E(u)$;

$$Y := \bigcap_{\substack{v \in T \\ E(v) \supseteq X}} E(v) \setminus X;$$

$F := F \cup \{X \to Y\}$;

endfor;

return F;

end.

Chúng ta minh họa thuật toán trên qua thí dụ sau: *Thiết lập hsd cho bảng T sau:*

T

x_1	x_2	x_3	x_4	E
0	0	0	0	$\emptyset$
1	0	0	0	$\{x_1\}$
0	1	0	0	$\{x_2\}$
0	0	1	0	$\{x_3\}$
1	0	1	0	$\{x_1, x_3\}$
0	1	1	0	$\{x_2, x_3\}$
0	0	0	1	$\{x_4\}$
0	1	0	1	$\{x_2, x_4\}$
1	1	0	1	$\{x_1, x_2, x_4\}$
1	1	1	1	$\{x_1, x_2, x_3, x_4\}$

$B^n \setminus T$

x_1	x_2	x_3	x_4	E
1	1	0	0	$\{x_1, x_2\}$
1	1	1	0	$\{x_1, x_2, x_3\}$
1	0	0	1	$\{x_1, x_4\}$
0	0	1	1	$\{x_3, x_4\}$
1	0	1	1	$\{x_1, x_3, x_4\}$
0	1	1	1	$\{x_2, x_3, x_4\}$

Vì T sẽ là bảng chân lý cho hsd cần tìm F nên F phải thỏa hai điều kiện (i) và (ii) sau đây:

(i) $(\forall t \in T) : (F(t) = 1)$, và

(ii) $(\forall t \in B^n \setminus T) : F(t) = 0$.

Để tiện theo dõi, ta sử dụng cột E để ghi các giá trị của $E(t)$ với mỗi dòng t của bảng.

Với dòng thứ nhất $u = (1, 1, 0, 0)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_1, x_2\}$ do đó

$Y = (\{x_1, x_2, x_4\} \cap \{x_1, x_2, x_3, x_4\}) \setminus \{x_1, x_2\} = \{x_4\}$.

Ta thu được $F = \{x_1x_2 \to x_4\}$.

Với dòng thứ hai $u = (1,1,1,0)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_1, x_2, x_3\}$ do đó
$Y = \{x_1, x_2, x_3, x_4\} \setminus \{x_1, x_2, x_3\} = \{x_4\}$.
Ta thu được $F = \{x_1x_2 \to x_4, x_1x_2x_3 \to x_4\}$.
Với dòng thứ ba $u = (1,0,0,1)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_1, x_4\}$ do đó
$Y = (\{x_1, x_2, x_4\} \cap \{x_1, x_2, x_3, x_4\}) \setminus \{x_1, x_4\} = \{x_2\}$.
Ta thu được $F = \{x_1x_2 \to x_4, x_1x_2x_3 \to x_4, x_1x_4 \to x_2\}$
Với dòng thứ tư $u = (0,0,1,1)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_3, x_4\}$ do đó
$Y = (\{x_1, x_2, x_3, x_4\} \setminus \{x_3, x_4\}) = \{x_1, x_2\}$.
Ta thu được $F = \{x_1x_2 \to x_4, x_1x_2x_3 \to x_4, x_1x_4 \to x_2, x_3x_4 \to x_1x_2\}$.
Với dòng thứ năm $u = (1,0,1,1)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_1, x_3, x_4\}$ do đó
$Y = (\{x_1, x_2, x_3, x_4\} \setminus \{x_1, x_3, x_4\}) = \{x_2\}$.
Ta thu được $F = \{x_1x_2 \to x_4, x_1x_2x_3 \to x_4, x_1x_4 \to x_2, x_3x_4 \to x_1x_2, x_1x_3x_4 \to x_2\}$.
Với dòng thứ sáu $u = (0,1,1,1)$ trong $B^n \setminus T$ ta có $X = E(u) = \{x_2, x_3, x_4\}$ do đó
$Y = (\{x_1, x_2, x_3, x_4\} \setminus \{x_2, x_3, x_4\} = \{x_1\}$.
Ta thu được đầu ra của thuật toán:
$F = \{x_1x_2 \to x_4, x_1x_2x_3 \to x_4, x_1x_4 \to x_2, x_3x_4 \to x_1x_2, x_1x_3x_4 \to x_2, x_2x_3x_4 \to x_1\}$.

Định lý 2.2. *Với bảng T trên U chứa các phép gán trị đơn vị e, không z và đóng với phép $*$, thuật toán DF tính đúng tập công thức suy dẫn F nhận T làm bảng chân lý.*

Chứng minh. Gọi F là tập công thức suy dẫn thu được qua thuật toán DF. Điều kiện T chứa e và z là hiển nhiên. Ta chứng minh với mỗi $t \in T$, $F(t) = 1$ và với mỗi $t \in B^n \setminus T$, $F(t) = 0$. Thật vậy, giả sử $t \in T$, $f = X \to Y \in F$ và $E(t) \supseteq X$. Ta có, theo thuật toán DF phải tồn tại một $u \in B^n \setminus T$ để $X = E(u)$ và

$$Y = \bigcap_{\substack{v \in T \\ E(v) \supseteq X}} E(v) \setminus X.$$

Vì $t \in T$ và $E(t) \supseteq X$ nên $E(t) \supseteq Y$, do đó $f(t) = 1$. Giả sử $t \in B^n \setminus T$ ta chỉ ra rằng trong F tồn tại một công thức f để $f(t) = 0$. Xét công thức $f = X \to Y$ xây dựng từ t theo thuật toán DF. Ta có $X = E(t)$ và

$$Y = \bigcap_{\substack{v \in T \\ E(v) \supseteq X}} E(v) \setminus X.$$

Từ biểu thức tính Y ta thấy X và Y không giao nhau. Kết hợp với điều kiện $X = E(t)$ ta suy ra $f(t) = 0$. Định lý được chứng minh.

Định lý 2.3. *Độ phức tạp của thuật toán DF là $t = O(k.m.n) = O(n.4^{n-1})$, trong đó n là số biến trong U, m là số dòng của bảng T, k là số dòng của bảng $B^n \setminus T$, $k + m = 2^n$.*

Chứng minh. Thuật toán lập k công thức suy dẫn. Để lập mỗi công thức ta phải thực hiện m phép duyệt, $m - 1$ phép lấy giao hai tập hợp và một phép lấy hiệu hai tập hợp. Mỗi phép toán tập hợp trên n phần tử của U đều đòi hỏi độ phức tạp n. Tổng hợp lại ta có $t = O(k.m.n)$.
Vì $k + m = 2^n$ nên tích $k.m$ đạt trị lớn nhất khi $k = m = 2^n/2 = 2^{n-1}$.
Khi đó $t = O(n.2^{n-1}.2^{n-1}) = O(n.4^{n-1})$.
Định lý được chứng minh.

Kết hợp Định lý 2.1 và Thuật toán DF ta thu được kết quả sau:

Định lý 2.4. *Bảng T trên U là bảng chân lý của một hsd khi và chỉ khi T chứa các phép gán trị đơn vị e, không z và đóng với phép $*$.*

Giảm độ phức tạp tính toán thông qua tổ chức dữ liệu

Nếu biểu diễn mỗi phần tử của bảng T như một số tự nhiên và sử dụng kỹ thuật đánh dấu ta dễ dàng tìm được các phần tử thuộc phần bù của T là $B^n \setminus T$. Khi đó các phép toán tập hợp sẽ được tổ chức thông qua các phép toán *thao tác bit* trên các số tự nhiên đã được cài đặt sẵn trong các bộ xử lí của máy tính. Thí dụ, với hai số tự nhiên x và y biểu diễn cho hai tập hợp thì ta có

$x \cap y = x \wedge y$

$x \cup y = x \vee y$

$x \setminus y = x \wedge (\text{not}\, y)$

$x \subseteq y$ khi và chỉ khi $x \wedge y = x$

Với thí dụ đã cho ta có

$T = \{0, 8, 4, 2, 10, 6, 1, 5, 13, 15\}$

$B^n \setminus T = \{0, 1, \ldots, 2^n - 1\} \setminus T = \{12, 14, 9, 3, 11, 7\}$

Tập hợp các kết quả đã trình bày ở phần trên ta thu được lời giải cho Bài toán 2.1.

Định lý 2.5. *Công thức logic f có thể biểu diễn qua một hsd khi và chỉ khi bảng chân lý của f chứa các phép gán trị đơn vị e, không z và đóng với phép $*$.*

Đối với hàm logic có dạng tuyển chuẩn tắc đã có những kết quả về biểu diễn tối thiểu như thủ tục Quine–McCluskey hay phương pháp Blake–Poreski [7]. Đối với một hsd, bài toán tìm dạng biểu diễn tối ưu, tức là dạng biểu diễn một hsd cho trước dưới dạng một hsd tương đương và chứa ít kí hiệu đơn nguyên nhất là thuộc lớp NPC [8].

3. QUAN HỆ ARMSTRONG

Cho tập hữu hạn các phần tử gọi là thuộc tính $U = \{x_1, x_2, \ldots, x_n\}$ trong đó mỗi x_i biến thiên trong một miền trị $d_i = \text{dom}\,(x_i)$. Đặt

$$D = \bigcup_{i=1}^{n} d_i \,.$$

Một *quan hệ* R với tập thuộc tính U là tập hữu hạn các ánh xạ $t : U \to D$ thỏa tính chất: $(\forall x_i \in U) : t(x_i) \in d_i$. Mỗi phần tử của R được gọi là một bộ của quan hệ.

Một *phụ thuộc hàm* (pth) f trên tập thuộc tính U là một mệnh đề dạng $f : X \to Y;\ XY \subseteq U$. Cho quan hệ R và một pth: $f : X \to Y$ trên cùng một tập thuộc tính U. Ta nói quan hệ R thỏa pth f, kí hiệu R/f nếu $(\forall u, v \in R) : (u(X) = v(X) \Rightarrow u(Y) = v(Y))$ trong đó $u(X)$ là hạn chế của ánh xạ u trên miền X. Quan hệ R thỏa tập pth F, R/F, nếu $(\forall f \in F) : (R/f)$. Nếu R là một quan hệ trên U, ta kí hiệu F_R là tập toàn thể các pth trên U thỏa trong quan hệ R, $F_R = \{f \mid R/f\}$.

Với mỗi quan hệ R trên tập thuộc tính U ta xây dựng bảng T_R có các cột là các phần tử trong U và các dòng $t(u,v)$ chứa các trị 0/1 được tạo từ các cặp bộ $u = (u_1, u_2, \ldots, u_n)$ $v = (v_1, v_2, \ldots, v_n)$ như sau: $t(u,v) = (t_1, t_2, \ldots, t_n)$, trong đó $t_i = 1$ nếu $u_i = v_i$, $t_i = 0$ nếu $u_i \neq v_i$.

Nếu xem mỗi pth f là một ctsd thì $R/f \Leftrightarrow T_R \subseteq T_f$.

Quan hệ R trên tập thuộc tính U được gọi là *quan hệ Armstrong* cho tập pth F nếu $T_R = T_F$ [1, 10, 11].

Bài toán 3.1. [1, 5, 6, 9] *Cho tập pth F trên U. Xây dựng quan hệ Armstrong R của F.*

Bài toán 3.2. *Cho quan hệ R trên U. Xác định xem R có phải là quan hệ Amstrong của một tập pth F nào đó hay không?*

TÀI LIỆU THAM KHẢO

[1] Armstrong W. W., Dependency structure of data-base relationship, *Information Processing 74*, North Holland, Amsterdam, 1974, 580–583.

[2] Berman J., Blok W. J., Positive Boolean dependencies, *Inf. Processing Letters* **27** (1988) 147–150.

[3] Codd E. F., Futher normalization of the database relational model, Database Systems, *Courant Compt. Sci., Symp.* **6** (1971) 65–98.

[4] Demetrovics J., Ho Thuan, Nguyen Xuan Huy, Le Van Bao, Translation of relation scheme, balanced relation schemes and the problem of key representation, *J. Inf. Process* **23** (2-3) (1987) 81–97.

[5] Demetrovics J., Nguyen Xuan Huy, Closed sets and translations of relations schemes, *Computers Math. Applic.* **21** (1) (1991) 13–23.

[6] Demetrovics J., Vu Duc Thi, Some results about normal forms for functional dependencies in the relational datamodel, *Discrete Applied Mathematics* **69** (1996) 61–74.

[7] Iablonski S. V., *Introduction to Discrete Mathematics*, Nauka, Moscow, 1979, (Russian).

[8] Maier D., *The Theory of Relation Database System*, Computer Science Press, 1982.

[9] Mannila H. and Raiha K. J., Design by example: An application of Armstrong relations, *Journal of Computer and System Science*, **33** (1986) 126–141.

[10] Nguyen Xuan Huy, Le Thi Thanh, Generalized positive Boolean dependencies, *J. Inform. Process Cybernet., EIK* **28** (6) (1992) 363–370.

[11] Sagiv Y., Delobel C., Parker D. S., and Fagin R., An equivalence between relational database dependencies and fragment of propositional logic, *J. ACM* **28** (1981) 425–453, *Corrigendum J. ACM* **34** (1987) 1016–1101.

[12] Ullman J. D., *Principles of Data-bases and Knowledge-bases Systems* (second edition), Computer Science Press, 1982.

Nhận bài ngày 20 - 4 - 2001
Nhận lại sau khi sửa ngày 13 - 11 - 2001

Viện Công nghệ thông tin