

Thuật toán xác định bao đóng và khóa theo tiếp cận hợp giải trong lớp các phụ thuộc logic

Unification Algorithms for Closures and Keys in Relation Schemas with Class of Logic Dependencies

Trương Thị Thu Hà, Nguyễn Thị Vân, Nguyễn Xuân Huy

Abstract: The algorithms for closures and keys in relation schemas with functional dependencies are well-known in theory of relational databases. However, the problems of closures and keys in relation schemas with positive Boolean dependencies are still opened. This paper proposes a solution to these problems. The results are presented by unification method which is a new technique to construct the basic algorithms for logic dependencies in data and knowledge bases.

Keywords: Positive Boolean dependencies, unification algorithm, membership problem, key algorithm, closure algorithm.

I. GIỚI THIỆU

Khóa của lược đồ quan hệ là tập tối thiểu các thuộc tính nhằm xác định đơn trị một bộ trong cơ sở dữ liệu quan hệ. Khóa giữ vai trò quan trọng trong các bài toán tìm kiếm và suy diễn. Chính vì vậy mà bài toán tìm khóa luôn được đề cập như một đối tượng cơ bản trong nghiên cứu về các loại phụ thuộc như phụ thuộc hàm, phụ thuộc đa trị, phụ thuộc sai khác [12, 13, 14], v.v.. Khái niệm khóa lại được dẫn xuất từ khái niệm bao đóng của một tập thuộc tính. Do đó bài toán tìm bao đóng và tìm khóa trong lược đồ quan hệ có trang bị phụ thuộc Boole dương và phụ thuộc Boole dương tổng quát [10, 11, 14] đang là vấn đề mở. Bài báo này trình bày các thuật toán tìm bao đóng và khóa theo tiếp cận của phép hợp giải trong logic hình thức [1].

Sau phần giới thiệu của bài báo, phần 2 và 3 trình bày vắn tắt các khái niệm và kết quả nghiên cứu trước

đó về phụ thuộc Boole dương. Phần 4 là một số hướng nghiên cứu của các nhóm tác giả về các loại phụ thuộc logic trong cơ sở dữ liệu. Phần 5 đề xuất thuật toán tìm khóa và tìm bao đóng trong lớp các phụ thuộc logic dựa trên thuật toán hợp giải. Phần 6 là kết luận của bài báo.

II. CÔNG THỨC BOOLE DƯƠNG

Cho $U = \{x_1, \dots, x_n\}$ là tập hữu hạn các biến Boole nhận giá trị trong tập $B = \{0, 1\}$. Tập các công thức Boole (CTB), kí hiệu $L(U)$, bao gồm các biểu thức được xây dựng từ các biến trong U , các hằng 0/1 và các phép toán logic $\wedge, \vee, \neg, \rightarrow$. Mỗi vector 0/1, $v = (v_1, \dots, v_n)$ trong không gian B^n được gọi là *phép gán trị*. Khi đó với mỗi CTB $f \in L(U)$, $f(v)$ là trị của công thức f đối với phép gán trị v . Kí hiệu e là phép gán trị đơn vị, $e = (1, 1, \dots, 1)$. Công thức $f \in L(U)$ gọi là *công thức Boole dương* (CTBD) nếu $f(e) = 1$.

Kí hiệu $P(U)$ là tập các công thức Boole dương trên U . Với mỗi công thức Boole $f \in L(U)$, kí hiệu $T_f = \{v \in B^n \mid f(v) = 1\}$ là *bảng chân lí* của f . Mỗi tập công thức $F \subseteq L(U)$ được hiểu là một hội logic của các công thức thành phần, $\bigwedge \{f \mid f \in F\}$. Khi đó, $T_F = \bigcap \{T_f \mid f \in F\}$ là *bảng chân lí* của tập công thức F . Ta đã biết $f \rightarrow g$ ($F \rightarrow G$) khi và chỉ khi $T_f \subseteq T_g$ ($T_F \subseteq T_G$).

Theo qui ước của lí thuyết cơ sở dữ liệu và tri thức, dấu phép hội thường được bỏ qua, giống như phép nhân trong đại số, dấu phép tuyển có thể được viết là “+”, dấu phép phủ định “ $\neg$ ” được thay bằng “ $\neg$ ”. Tập các thuộc tính (biến logic) được viết liền nhau như một dãy kí tự.

III. PHÉP SÁNH TRỊ

Ta quy ước mỗi miền trị d_x của biến x trong U chứa ít nhất hai phần tử. Với mỗi miền trị d_x , xét ánh xạ $\alpha_x: d_x \times d_x \rightarrow B$ thoả ba tính chất sau [11, 14]:

- $\forall a, b \in d_x$
- $\alpha_x(a, a) = 1$
- $\alpha_x(a, b) = \alpha_x(b, a)$
- $\exists c \in d_x: \alpha_x(a, c) = 0$

α_x chính là quan hệ (hai ngôi) bộ phận thực sự, thoả các tính chất phản xạ và đối xứng trên miền trị d_x . Việc xác định α_x được hiểu là thiết lập một *phép sánh trị* trên miền trị d_x cho x .

Quan hệ bằng $=_x$ được định nghĩa: $\forall a, b \in d_x: =_x(a, b) = 1$, khi và chỉ khi $a = b$ là trường hợp riêng của phép sánh trị và được ngầm định trong trường hợp không định nghĩa tường minh phép sánh trị cho thuộc tính này.

Cho quan hệ r trên tập thuộc tính U , với mỗi bộ $v \in r$, thuộc tính $x \in U$ và tập con thuộc tính $X \subseteq U$, kí hiệu $u.x$ ($u.X$) là trị của thuộc tính u (của tập con thuộc tính X) trong bộ u . Với mỗi cặp bộ $u, v \in r$, $u = (u_1, u_2, \dots, u_n)$, $v = (v_1, v_2, \dots, v_n)$, ta đặt tương ứng một vector $t \in B^n$, $t = (t_1, t_2, \dots, t_n)$ và kí hiệu là $\alpha(u, v)$, trong đó thành phần ứng với thuộc tính x trong U chính là ảnh của ánh xạ $\alpha_x(u.x, v.x)$. Khi đó mỗi quan hệ r sẽ được đặt tương ứng với tập các vector 0/1:

$$T_r = \{ \alpha(u, v) \mid u, v \in r \}$$

và được gọi là *bảng trị* của quan hệ r [10, 14, 15].

IV. QUAN HỆ GIỮA CÁC LOẠI PHỤ THUỘC LOGIC TRONG CỐ SỞ DỮ LIỆU

Phụ thuộc hàm đã là cách thức truyền thống để thiết kế lược đồ, ràng buộc toàn vẹn, tối ưu hoá truy vấn,... Với đề xuất đầu tiên cho khía cạnh hướng lược đồ, là định nghĩa cơ sở trên phép suy dẫn thông thường ($\rightarrow$) và phép sánh trị đẳng thức ($=$). Tuy nhiên, trong hướng dữ liệu thực tiễn không phải lúc nào cũng đồng nhất. Do đó, rất nhiều nhóm nghiên cứu gần đây đã đề xuất các loại phụ thuộc khác nhau để phù hợp hơn với đặc trưng của dữ liệu, ví dụ như việc lấy được

dữ liệu ngược nhau, phục hồi dữ liệu, loại bỏ dữ liệu trùng lặp... [4, 16, 17].

Phụ thuộc hàm có điều kiện [2, 3, 5]. Phụ thuộc hàm có điều kiện là mở rộng các phụ thuộc hàm bằng cách củng cố các mẫu của các hằng số có quan hệ về ngữ nghĩa. Các phụ thuộc hàm có điều kiện đã được chứng minh là hiệu quả hơn so với phụ thuộc hàm trong việc phát hiện và sửa chữa các điểm không nhất quán (tình trạng không sạch) của dữ liệu.

Một phụ thuộc hàm có điều kiện φ trên quan hệ r là một cặp $(X \rightarrow x, T_P)$ thoả mãn:

- $X \subseteq U$ và $x \in U$, với U là tập các thuộc tính
- $X \rightarrow x$ là một phụ thuộc hàm
- T_P là tập bộ mẫu trong X và x tương ứng.

Phụ thuộc hàm mềm [8]. Ilyas nghiên cứu phụ thuộc hàm mềm với các giá trị của thuộc tính được dự đoán bởi các giá trị của thuộc tính khác. Trong phụ thuộc hàm, giá trị của X xác định đầy đủ giá trị của Y . Trong phụ thuộc hàm mềm, giá trị của X xác định giá trị của Y nhưng không đầy đủ, chỉ với xác suất cao. Ví dụ, trong một cơ sở dữ liệu của xe ô tô, một phụ thuộc mềm có thể: $model \rightarrow make$. Với $model = 323$, ta suy ra $make = Mazda$ với xác suất cao, nhưng xác suất nhỏ có thể là BMV . Do đó phụ thuộc hàm mềm được sử dụng trong việc cải tiến sự đánh giá chọn lọc trong tối ưu hoá truy vấn và làm nên chỉ số so sánh tối thiểu.

Phụ thuộc hàm độ đo [9]. Koudas nghiên cứu các phụ thuộc với độ đo gần giống trên tập thuộc tính Y khi cho các giá trị được so sánh chính xác trên tập thuộc tính X ; với $X, Y \subseteq U$.

Trước hết, ta xây dựng độ đo cho tập các điểm S trong không gian, khoảng cách giữa hai điểm bất kỳ d : $d_x \times d_x \rightarrow \mathbb{R}$ là độ đo được xác định trên miền của x , thoả ba tính chất:

- $\forall a, b \in d_x:$
- $d(a, b) \geq 0$, $d(a, b) = 0$ khi và chỉ khi $a = b$
- $d(a, b) = d(b, a)$
- $\forall c \in d_x: d(a, c) + d(c, b) \geq d(a, b)$.

Đường kính của tập các điểm S trong không gian độ đo là khoảng cách lớn nhất giữa cặp điểm p, q trong S , $\Delta_d(S) = \max_{p,q \in S} d(p, q)$.

Cho quan hệ r trên tập thuộc tính U ; hai tập con $X, Y \subseteq U$. Cho tập bộ $T \subseteq r$, độ đo d trên Y , tham số $\delta \geq 0$.

Ta nói có phụ thuộc hàm độ đo $X \xrightarrow{\delta} Y$ nếu $\max_{T \in \pi_X} \Delta_d(T.Y) \leq \delta$. Trong đó, $T.Y = \{t.Y \mid t \in T\}$, $\pi_X(r) = \{[t]_X\}$ là phân hoạch của quan hệ r trên tập thuộc tính X .

Phụ thuộc đối sánh [6]. *Phụ thuộc đối sánh* được đề xuất đầu tiên trong Fan (2008) với việc đặc tả các luật đối sánh với biến đối tượng. Trong quan hệ r , với mỗi thuộc tính x có một miền khoảng cách tương ứng d_x .

Một toán tử đồng dạng $\approx$ trên một thuộc tính x được định nghĩa trên miền khoảng cách của x , $\approx: d_x \times d_x \rightarrow \{\text{true}, \text{false}\}$, với $u, v \in d_x$. Cho tập các thuộc tính X , toán tử đồng dạng trên X nhận trị $\approx \text{true}$ nếu các toán tử đồng dạng trên $\forall x \in X$ đều là true. Một toán tử đối sánh $\delta \leftrightarrow$ trên thuộc tính x được định nghĩa trên miền khoảng cách của x . Nó là true nếu hai giá trị đồng dạng.

Một phụ thuộc đối sánh có dạng: $[X \approx] \rightarrow [Y \leftrightarrow]$, trong đó $X, Y \subseteq U$, và $\approx, \leftrightarrow$ biểu thị sự tương ứng đồng dạng các toán tử đối sánh trên tập thuộc tính X và Y theo thứ tự.

Phụ thuộc tuần tự [7]. Golab đề xuất *phụ thuộc tuần tự* có dạng: $X \rightarrow_g Y$, với $X \subseteq U$ là các thuộc tính có thứ tự, $Y \subseteq U$ được trang bị một độ đo nào đó, g là khoảng thời gian. Điều đó nói lên rằng khi các bộ được sắp xếp trên X (thí dụ như theo thuật toán group by X trong SQL), thì khoảng cách giữa các giá trị Y của hai bộ bất kỳ liên tiếp nhau trong khoảng thời gian g .

Một phụ thuộc tuần tự có điều kiện là một cặp: $(X \rightarrow_g Y, t_r)$ với t_r là bộ mẫu. Với mỗi mẫu t_r đặc tả một dãy giá trị của X để đồng nhất một tập các bộ trên r .

Phụ thuộc sai khác [12, 13]. *Phụ thuộc sai khác* trên quan hệ r có dạng $g: \phi_X \rightarrow \phi_Y$, trong đó X, Y là các tập con của tập thuộc tính U ; ϕ_X, ϕ_Y là các hàm sai khác

thỏa độ sai khác $\delta_a: d_a^2 \rightarrow D$ thỏa hai tính chất phản xạ và đối xứng của độ đo.

Bảng 1. So sánh các loại phụ thuộc logic

Loại phụ thuộc logic	Phép toán logic	Phép sánh trị	Tập trị
Phụ thuộc có điều kiện	$\rightarrow$	Thỏa tập mẫu T_p	$\{0,1\}$
Phụ thuộc hàm mềm	$\rightarrow$	$=$ (Với xác suất cao)	$[0;1]$
Phụ thuộc hàm độ đo	$\rightarrow$	Độ đo khoảng cách δ	$[0;1]$
Phụ thuộc hàm đối sánh	$\rightarrow$	Đồng dạng trên toán tử đối sánh $\approx$ cho về trái và $\leftrightarrow$ cho về phải	$\{0,1\}$
Phụ thuộc tuần tự	$\rightarrow$	Thời khoảng g	$[a..b]$
Phụ thuộc sai khác	$\rightarrow$	Hàm sai khác ϕ thỏa độ sai khác δ_a với hai tính chất không âm và đối xứng	$[0;1]$
Phụ thuộc Boole dương	$\wedge, \vee, \neg, \rightarrow$	Phép sánh đẳng thức ($=$)	$\{0,1\}$
Phụ thuộc Boole dương tổng quát	$\wedge, \vee, \neg, \rightarrow$	Thỏa phép sánh trị α với ba tính chất phản xạ, đối xứng và bộ phận.	$\{0,1\}$
Phụ thuộc Boole dương đa trị	$\wedge, \vee, \neg, \rightarrow$	Thỏa phép sánh trị α với ba tính chất phản xạ, đối xứng và bộ phận.	$[0;1]$

Phụ thuộc Boole dương tổng quát [14]. Mỗi công thức Boole dương f trong $P(U)$ với phép sánh trị α là một *phụ thuộc Boole dương tổng quát* (PTBDTQ). Quan hệ r trên tập thuộc tính U thỏa PTBDTQ f (tập PTBDTQ F) và viết $r(f)$ ($r(F)$) nếu $T_r \subseteq T_f$ ($T_r \subseteq T_F$).

Phụ thuộc Boole dương đa trị [14]. Tập trị Boole $\mathfrak{B} = \{b_1, \dots, b_k\}$ gồm k giá trị thực trong đoạn $[0; 1]$, $k \geq 2$ được sắp tăng và thỏa các điều kiện sau:

- $0 \in \mathfrak{B}$,
- $\forall b \in \mathfrak{B} \Rightarrow 1 - b \in \mathfrak{B}$.

Với mỗi trị $b \in \mathfrak{B}$ ta định nghĩa hàm I_b

$$\forall x \in \mathfrak{B} : I_b(x) = 1 \text{ nếu } x = b, \text{ ngoài ra } I_b(x) = 0$$

Cho $U = \{x_1, \dots, x_n\}$ là tập hữu hạn các biến Boole, $\mathfrak{B}$ là tập trị Boole. Khi đó các công thức Boole đa trị (CTBDT) là các công thức được xây dựng trên các biến của U , các trị trong $\mathfrak{B}$, các hàm I_b với $b \in \mathfrak{B}$ và các phép toán $\wedge, \vee, \neg, \rightarrow$. Mỗi công thức Boole dương đa trị f với $f(e) = 1$ thỏa phép sánh trị α trên tập trị $\mathfrak{B}$ là *phụ thuộc Boole dương đa trị* (PTBDĐT).

V. LỚP CÁC PHỤ THUỘC LOGIC

Trong [10, 14, 15] đã chỉ ra mối quan hệ giữa một số loại phụ thuộc trong cơ sở dữ liệu, ... và gọi chung là *lớp các phụ thuộc logic* (PTLG). Mỗi phụ thuộc trong lớp này chính là một biểu thức Boole trên tập hữu hạn các thuộc tính. Thí dụ, phụ thuộc hàm là trường hợp riêng của PTBDTQ khi các biểu thức logic có dạng hội suy dẫn và các phép sánh trị $=$.

V.1. Định lý tương đương

Cho tập các PTBD F và một PTBD f trên tập thuộc tính U . Cho quan hệ r trên tập thuộc tính U . Quan hệ $r(U)$ thỏa PTBD f và viết $r(f)$ nếu $T_r \subseteq T_f$, quan hệ $r(U)$ thỏa tập PTB F , $R(F)$, nếu $T_r \subseteq T_F$. Ta nói tập PTBD F suy ra PTBD f theo quan hệ và viết $F \vdash f$, nếu với mọi quan hệ $r(U)$, r thỏa F kéo theo r thỏa f :

$$F \vdash f \Leftrightarrow \forall r(U): r(F) \rightarrow r(f)$$

Ta nói F suy ra f theo quan hệ có không quá hai bộ và viết $F \vdash_2 f$, nếu với mọi quan hệ r trên tập thuộc tính U và có không quá hai bộ, r thỏa F thì r thỏa f :

$$F \vdash_2 f \Leftrightarrow \forall r(U), \|r\| \leq 2 : r(F) \rightarrow r(f)$$

Định lý 5.1. (Định lý tương đương) [10, 11, 15]

Cho tập PTBD F và một PTBD f trên U . Ba mệnh đề sau là tương đương:

- $F \rightarrow f$ (suy dẫn logic)
- $F \vdash f$ (suy dẫn theo quan hệ)
- $F \vdash_2 f$ (suy dẫn theo quan hệ có không quá 2 bộ)

V.2. Bài toán thành viên

Kí hiệu $F^+ = \{f \in P(U) \mid F \vdash f\}$ là tập toàn bộ các PTBD được suy dẫn theo quan hệ từ tập PTBD F . Nếu $F \vdash f$, có nghĩa $f \in F^+$ thì f là thành viên của F^+ . Hiển nhiên, nếu $f \in F$ thì $F \vdash f$. Vấn đề đặt ra là ngoài F thì F^+ còn chứa PTBD nào khác?

Bài toán thành viên [18]: Cho F là tập PTBD và f là một PTBD trên U . Hãy xác định $F \vdash f$? Nói cách khác là xác định $f \in F^+$?

Định lý 5.1 cho thấy việc kiểm tra $F \vdash f$ tương đương với việc kiểm tra suy dẫn logic $F \rightarrow f$ và cũng tương đương với việc kiểm tra trên các quan hệ có không quá hai bộ.

Định lý tương đương cũng cho biết bài toán thành viên tương đương với bài toán suy dẫn của logic mệnh đề trong lớp các phụ thuộc Boole dương. Nếu chọn cách kiểm tra $F \vdash f$ theo quan hệ thì ta phải xây dựng 2^m quan hệ, trong đó m là tích các lực lượng của miền trị của các thuộc tính. Để xây dựng bảng chân lí T cho mỗi quan hệ gồm k bộ ta phải xét k^2 cặp bộ. Trong trường hợp các thuộc tính có vô hạn giá trị thì tiếp cận theo quan hệ là không thể. Do đó, theo định lý tương đương, thay vì kiểm tra $F \vdash f$ theo quan hệ, ta có thể chứng minh $F \rightarrow f$ theo logic, cụ thể là vận dụng *thuật toán hợp giải* để chứng minh $F \rightarrow f$.

Thuật toán 5.1. (Thuật toán hợp giải)

Trong logic, bài toán suy dẫn $F \rightarrow f$ đối với tập công thức Boole F và công thức Boole f trên tập biến Boole U được giải theo *thuật toán hợp giải* như sau [1]

Để chứng minh $F \rightarrow f$ ta xét phủ định $g = (F \rightarrow f)' \equiv Ff'$. Nếu chứng minh được Ff' là sai thì $F \rightarrow f$, ngược lại F không suy dẫn được f .

Bước 1. *Chuẩn hóa CNF*. Đưa g về dạng chuẩn hội (CNF) h , tức là viết g dưới dạng hội của các tuyến cơ sở, trong đó mỗi hạng tử của mọi tuyến cơ sở đều là các biến đơn hoặc phủ định của biến đơn trong g . Gọi thủ tục này là CNF, ta có $h = \text{CNF}(g)$.

Bước 2. *Hợp giải*. Lần lượt thay cặp nhân tử $(a+B)(a'+C)$ trong h bằng $(B+C)$ cho đến khi không còn thay được nữa, hoặc gặp hai nhân tử phủ định nhau là x và x' . Ta gọi thủ tục này là Unif(h).

Lưu ý $x \equiv x+0 \equiv 0+x$, nên ta có thể thay $(a+B)a'$ hoặc $(a'+B)a$ bằng B trong quá trình hợp giải.

Bước 3. *Kết luận*. Nếu h bị xóa hết (kết quả nhận được là tập rỗng $\emptyset$), tức là *hợp giải thành công* thì kết luận $F \rightarrow f$; ngược lại, với mọi khả năng thay thế như trên, h không bị xóa hết, tức là *hợp giải không thành công* thì kết luận F không suy dẫn ra f .

Tổng hợp các bước trên ta thu được thuật toán Member_PBD(F, f) kiểm tra tập PTBD f có là thành viên trong tập F^+ ?

Nếu $f \in F^+$ thuật toán cho giá trị 1 (true), ngược lại thuật toán cho ra giá trị 0 (false).

Thuật toán 5.2. [18] (Thuật toán thành viên)

Algorithm Member_PBD(F, f)

Input: Tập PTBD F ; PTBD f .

Output: true, if $F \rightarrow f$; else false

Method

$g \leftarrow Ff'$; // gán Ff' cho g

$h \leftarrow \text{CNF}(g)$;

return Unif(h) = $\emptyset$;

End Member_PBD.

Mệnh đề 5.1. Bài toán thành viên trong lớp các phụ thuộc Boole dương thuộc lớp NP đầy đủ (NPC) [18]

V.3. Thuật toán bao đóng

Gọi U là tập các thuộc tính và F là tập PTBD trên U , $X \subseteq U$. Ta định nghĩa bao đóng X^+ của X là tập các thuộc tính sau:

$$X^+ = \{a \in U \mid X \rightarrow a \in F^+\}$$

Để tìm bao đóng X^+ của X , ta khởi tạo $X^+ = X$ sau đó vận dụng thuật toán thành viên để xét cho từng thuộc tính $a \in U-X$, nếu $X \rightarrow a \in F^+$ thì bổ sung thêm a vào X^+ . Tiếp tục quá trình cho đến khi không mở rộng thêm X^+ được nữa.

Để ý rằng nếu F đã được đưa về dạng CNF thì điều kiện $X \rightarrow a \in F^+$ sẽ cho ta $\text{CNF}(F(X \rightarrow a)) = FXa'$, do đó sẽ tương đương với điều kiện $\text{Unif}(FXa') = \emptyset$.

Thuật toán 5.3. (Thuật toán tìm bao đóng)

Algorithm Closure_PBD(X, F)

Input: Tập PTBD F dạng CNF

Tập thuộc tính $X \subseteq U$

Output: $X^+ = \{a \in U \mid X \rightarrow a \in F^+\}$

Method

$Y \leftarrow X$; // Y chứa kết quả

$V \leftarrow U-X$; // phần bù của X

repeat

$Z \leftarrow Y$;

for each attribute a in V do

if $\text{Unif}(FYa') = \emptyset$ then

add a to Y ;

endif;

endfor;

until $Y = Z$;

return Y ;

End Closure_PBD.

Theo mệnh đề 5.1 thì thuật toán tìm bao đóng trong lớp các phụ thuộc Boole dương là NP đầy đủ (NPC).

Thí dụ 5.1. Cho tập thuộc tính $U = abcd$, tập PTBD $F = \{b'+c, a'+b\}$, $X = ab$, tìm bao đóng X^+ ?

Ta có $F = (b'+c)(a'+b)$ hiện ở dạng CNF.

Đặt $Y = X = ab$, $V = U - X = cd$. Xét các thuộc tính trong V .

Với $c \in V$, ta có $\text{Unif}(FXc') = (\underline{b}'+c)(a'+\underline{b})abc' = (c+\underline{a}')\underline{a}bc' = bcc' = \emptyset$. Vậy $Y = abc$.

Với $d \in V$, ta có $\text{Unif}(FXd') = (\underline{b}'+c)(a'+\underline{b})abcd' = (\underline{a}'+c)\underline{a}bcd' = cbcd' \neq \emptyset$ nên Y không thay đổi. Cuối cùng ta thu được $X^+ = (ab)^+ = abc$.

V.4. Thuật toán tìm khóa

Cho F là tập PTBD trên U . Tập $K \subseteq U$ được gọi là khóa nếu

- $K^+ = U$
- $\forall a \in K: (K-a)^+ \neq U$

Để tìm tập khóa K , trước tiên đặt $K = U$. Sau đó rút gọn K bằng cách xét từng thuộc tính a trong K , nếu $(K-a)^+ = U$ thì loại a ra khỏi K .

Do ta luôn bảo toàn bất biến $K^+ = U$ nên điều kiện $(K-a)^+ = U$ tương đương với điều kiện $(K-a) \rightarrow a$. Điều kiện này lại tương đương với điều kiện $\text{Unif}(F(K-a)a') = \emptyset$.

Thuật toán 5.4. (Thuật toán tìm khóa)

Algorithm Key_PBD(U, F)

Input: Tập thuộc tính U ; Tập PTBD F dạng CNF

Output: $K \subseteq U$:

- $K^+ = U$
- $\forall a \in K: \text{Unif}(F(K-a)a') = \emptyset$

Method

```

K ← U;
for each attribute a in K do
    if Unif(F(K-a)a') = ∅ then
        delete a from K;
    endif
endfor
return K;

```

End Key_PBD.

Theo mệnh đề 5.1 thì thuật toán tìm khóa trong lớp các phụ thuộc Boole dương là NP đầy đủ (NPC).

Thí dụ 5.2. Cho tập các thuộc tính $U = \{a, b, c, d\}$, tập phụ thuộc Boole dương $F = \{b'+c, a'+b\}$, hãy xác định một khóa K trong quan hệ r ?

Trước hết ta đưa F về dạng CNF, $F = (b'+c)(a'+b)$.

Áp dụng thuật toán tìm khóa, đặt $K = U = abcd$. Lần lượt rút gọn khóa K bằng cách xét từng thuộc tính a, b, c và d trong K :

Xét a : $K-a = bcd$. Ta có: $\text{Unif}(Fbcd'a') = (\underline{b}'+c)(a'+\underline{b})bcd'a' = (a'+c)bcd'a' \neq \emptyset$. Vậy, K không thay đổi, $K = abcd$.

Xét b : $K-b = acd$. Ta có: $\text{Unif}(Facdb') = (\underline{b}'+c)(\underline{a}'+b)\underline{a}cdb' = (b'+c)bcdb' = \emptyset$. Vậy, $K = acd$.

Xét c : $K-c = ad$. Ta có $\text{Unif}(Fadc') = (b'+c)(\underline{a}'+b)\underline{a}dc' = (\underline{b}'+c)\underline{b}dc' = cdc' = \emptyset$. Vậy, $K = ad$.

Xét d : $K-d = a$. Ta có $\text{Unif}(Fad') = (b'+c)(\underline{a}'+b)\underline{a}d' = (b'+c)bd' = cd' \neq \emptyset$. Vậy, K không thay đổi. Cuối cùng ta thu được khóa $k = ad$.

VI. KẾT LUẬN

Thuật toán hợp giải có thể được cài đặt như một tiện ích trong các thư viện của các ngôn ngữ lập trình logic như Prolog, P#, Lisp. Nghiên cứu các phụ thuộc Boole dương theo tiếp cận của logic hình thức cho phép ta thiết kế và quản lý các cơ sở dữ liệu và tri thức với những phụ thuộc phức tạp và đa dạng một cách thống nhất. Các kết quả thu được trong bài này có thể được vận dụng trong lĩnh vực khai thác tri thức từ các tập dữ liệu lớn.

TÀI LIỆU THAM KHẢO

- [1] BAADER F., SNYDER W., “*Hanbook of Automated Reasoning*”. Ed. Alan Robinson and Andrei Voronkov, Elsevier Science Publishers B.V, 2001.
- [2] BOHANNON P., FAN W., GEERTS F., JIA X., KEMENTSIETSIDIS A., (2007), “*Conditional functional dependencies for data cleaning*”, In *ICDE*, pp.746-755.
- [3] BRAVO L., FAN W., MA S., (2007), “*Extending dependencies with condition*”, In *VLDB*, pp.243-254.

- [4] DAVID MAIER, V. M MEGLER, KRISTIN TUFTE (2014), Challenges for Dataset Search, “*Database System for Advanced Applications*”, Lecture Notes in Computer Science Volum 8421, 1-15.
- [5] FAN W., GEERTS F., LAKSHMANAN L. V. S., XIONG M., (2009), “*Discovering conditional functional dependencies*”, In *ICDE*, pp. 1231-1234.
- [6] FAN W., LI J., JIA X., MA S., (2009), “*Reasoning about record matching rules*”, *PVLDB*.
- [7] GOLAB L., KARLOFF H., KORN F., SAHA A., SRIVASTAVA D., (2009), “*Sequential dependencies*”, *PVLDB*, 2(1), pp.574-585.
- [8] ILYAS I. F., MARKL V., HAAS P. J., BROWN P., ABOULNAGA A., (2004), “*Cords: Automatic discovery of correlations and soft functional dependencies*”, In *Sigmod Conference*, pp.647-658.
- [9] KOUDAS N., SAHA A., SRIVASTAVA D., VENKATASUBRAMANIAN S., (2009), “*Metric functional dependencies*”, In *ICDE*, pp.1275-1278.
- [10] NGUYEN XUAN HUY, LE THI THANH, “*Generalized Positive Boolean Dependencies*”, *J. Inf. Process. Cybern. EIK*, 28, 1992, 6, 363-370.
- [11] SAGIV Y., DELOBEL C., PARKER D. S., FAGIN R., “*An equivalence between Relational Database Dependencies and a Fragment of Propositional Logic*”, *J.ACM*, 28, 1981, 435 - 453. Corrigendum *J. ACM*, 34, 1987, 1016 -1018.
- [12] SONG S., CHEN L. and CHENG H., “*Efficient Determination of Distance Thresholds for Differential Dependencies*”, *IEEE Transactions on knowledge and data engineering*, Vol. 26, No. 9, 2014, 2179-2192.
- [13] SONG S., CHEN L., “*Differential Dependencies: Reasoning and Discovery*”, *ACM Trans. Datab. Syst.* 9, 4, Article 39, (March 2011), 42 pages.
- [14] NGUYỄN XUÂN HUY, “*Các phụ thuộc logic trong cơ sở dữ liệu*”, NXB Thống Kê, 2006.
- [15] NGUYỄN XUÂN HUY, CAO TÙNG ANH, TRƯỜNG THỊ THU HÀ, LƯƠNG NGUYỄN HOÀNG HOA, BÙI ĐỨC MINH (2012), “*Các biến thể của phụ thuộc sai khác trong cơ sở dữ liệu quan hệ*”, Kỷ yếu Hội thảo quốc gia lần thứ XV: “Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông”, NXB Khoa học và Kỹ thuật, ISBN 978-604-67-0251-137- 41.
- [16] NGUYỄN XUÂN HUY, ĐÀM GIA MẠNH, VŨ THỊ THANH XUÂN, KIM LAN HƯƠNG (2001), “*Về một lớp công thức logic suy dẫn*”, Tạp chí Tin học và điều khiển học, 17 (4), tr. 17-22.
- [17] NGUYỄN XUÂN HUY, ĐOÀN VĂN BAN, ĐÀM GIA MẠNH, NGUYỄN THỂ DŨNG (2001), “*Về mối liên hệ giữa suy diễn phụ thuộc hàm và suy diễn logic*”, Tạp chí Tin học và điều khiển học, 17(4), tr.11-16.
- [18] TRƯỜNG THỊ THU HÀ, “*Độ phức tạp của các thuật toán chuẩn hóa phụ thuộc Boole dương*”, Tạp chí Công nghệ thông tin và truyền thông, ISSN 1859 – 3526, Tập V-1, Số 12(32), 12-2014.

Nhận bài ngày: 29/02/2016

SƠ LƯỢC VỀ TÁC GIẢ

TRƯƠNG THỊ THU HÀ



Sinh năm 1979 tại Nghệ An.

Tốt nghiệp Cử nhân CNTT tại Trường ĐH Sư phạm Hà Nội năm 2000, Thạc sĩ CNTT tại Trường ĐH Công nghệ, ĐH Quốc gia Hà Nội năm 2006. Đang là nghiên cứu sinh tại Khoa CNTT, Học viện Kỹ thuật Quân sự.

Hiện công tác tại Khoa CNTT, trường Đại học Kinh doanh và Công nghệ Hà Nội.

Lĩnh vực nghiên cứu: Cơ sở dữ liệu, Công nghệ phần mềm.

Email: thuha.bh@gmail.com

NGUYỄN XUÂN HUY



Sinh năm 1944, Hải Phòng.

Tốt nghiệp Cử nhân Toán, ĐH Sư phạm Leningrad (Liên Xô) năm 1973, Tiến sĩ CNTT năm 1982, Tiến sĩ khoa học CNTT, Viện Hàn lâm Khoa học Liên Xô năm 1990.

Là Trưởng Phòng Cơ sở dữ liệu và Lập trình, Viện CNTT, Viện Hàn lâm KH&CN Việt Nam từ năm 1997-2009.

Hướng nghiên cứu: Cơ sở dữ liệu và Công nghệ phần mềm. Email: nxhuy564@gmail.com

NGUYỄN THỊ VÂN



Sinh năm 1985 tại Hà Tĩnh.

Tốt nghiệp Cử nhân CNTT tại Trường ĐH Kinh doanh và Công nghệ Hà Nội năm 2011, Thạc sĩ ngành Khoa học máy tính tại Trường ĐH CNTT và Truyền thông năm 2014.

Hiện công tác tại Khoa CNTT, Trường Cao đẳng Cộng đồng Hà Nội.

Lĩnh vực nghiên cứu: Các phụ thuộc logic trong cơ sở dữ liệu, mô hình dữ liệu và cơ sở dữ liệu.

Email: van.cdcd@gmail.com