VỀ MỘT PHÉP BIẾN ĐỔI TIỀN XỬ LÝ HIỆU QUẢ CÁC TẬP PHỤ THUỘC HÀM

Vũ Quốc Tuấn[1*], Hồ Thuần[2]

Tóm tắt: *Trong [1] và [2], Ángel Mora và các cộng sự đã thiết kế một phép biến đổi tiền xử lý sử dụng toán tử thay thế của logic thay thế SL_{FD} để loại bỏ dư thừa trong tập phụ thuộc hàm ban đầu nhằm thu được một tập phụ thuộc hàm tương đương với kích thước nhỏ hơn trong thời gian đa thức. Cơ sở và tính đúng đắn của phép biến đổi tiền xử lý này được chứng minh bởi Định lý 6 trong [1]. Trong bài báo này, chúng tôi sẽ chỉ ra một lỗi sai không chấp nhận được trong chứng minh của Định lý 6 và đưa ra một chứng minh đúng và đơn giản hơn cho định lý đó. Một số nhận xét về phép biến đổi tiền xử lý cũng được đưa ra.*

Từ khóa: Cơ sở dữ liệu quan hệ, Lược đồ quan hệ, Phụ thuộc hàm, Phép biến đổi tiền xử lý.

1. MỞ ĐẦU

Trong [1] và [2], Ángel Mora và các cộng sự đã thiết kế một phép biến đổi tiền xử lý sử dụng toán tử thay thế của logic thay thế SLFD để loại bỏ dư thừa trong tập phụ thuộc hàm ban đầu nhằm thu được một tập phụ thuộc hàm tương đương với kích thước nhỏ hơn trong thời gian đa thức. Cơ sở và tính đúng đắn của phép biến đổi tiền xử lý này được chứng minh bởi định lý 6 trong [1]. Trong bài báo này, chúng tôi sẽ chỉ ra một lỗi sai không chấp nhận được trong chứng minh của Định lý 6 và đưa ra một chứng minh đúng và đơn giản hơn cho định lý đó. Một số nhận xét về phép biến đổi tiền xử lý cũng được đưa ra.

Bài báo được tổ chức như sau: phần thứ hai nhắc lại một số khái niệm và kết quả quan trọng của mô hình quan hệ, giới thiệu về logic Paredaens dựa vào cách trình bày trong [2]. Trong phần này cũng nhắc lại định nghĩa thế nào là một tập F các phụ thuộc hàm có dư thừa, phát biểu lại Định lý 6 trong [1] và chỉ ra chỗ sai trong chứng minh của định lý đó. Chứng minh đúng của Định lý 1 được cho trong phần thứ ba cùng với một số nhận xét. Trong phần thứ ba cũng giới thiệu về thủ tục *removeRedundancy* được trình bày trong [2] với một vài cải tiến cùng một số ví dụ ứng dụng. Kết luận được giới thiệu trong phần thứ tư.

2. MÔ HÌNH QUAN HỆ VÀ LOGIC PAREDAENS

2.1. Mô hình quan hệ

Trong mô hình quan hệ của E.F.Codd, dữ liệu được lưu trữ dưới dạng các quan hệ (các bảng). Mỗi quan hệ được định nghĩa trên một tập hữu hạn các thuộc tính $\Omega = \{A_1, A_2,..., A_n\}$, trong đó mỗi thuộc tính A_i lấy giá trị trong một miền tương ứng $Dom(A_i)$. Như vậy, một quan hệ R xác định trên Ω là một tập con của tích Descartes $Dom(A_1) \times Dom(A_2) \times ... \times Dom(A_n)$. Nói cách khác, R là một tập các bộ t có dạng $t = (a_1, a_2,...,a_n)$ trong đó $a_i \in Dom(A_i)$ với mọi $i = 1, 2, ..., n$.

Cho $X \subseteq \Omega$, $t \in R$. Khi đó, hình chiếu của t trên X, ký hiệu $t[X]$ là bộ sao cho $t[X](a_i) = t(a_i)$, $a_i \in Dom(A_i)$ với mọi $A_i \in X$.

Định nghĩa 1 (*Phụ thuộc hàm*). Cho R là một quan hệ trên Ω. Mọi khẳng định có dạng $X \rightarrow Y$, trong đó, X, $Y \subseteq \Omega$ được gọi là một phụ thuộc hàm trên R. Ta nói R thỏa $X \rightarrow Y$ nếu với mọi $t_1, t_2 \in R$ có $t_1[X] = t_2[X]$ kéo theo $t_1[Y] = t_2[Y]$.

Ký hiệu FD_R là tập sau:

$$FD_R = \{X \rightarrow Y \mid X, Y \subseteq \Omega, R \text{ thỏa } X \rightarrow Y\}$$

Trong [3], W.W. Armstrong đã chứng minh ngữ nghĩa của phụ thuộc hàm, đặc trưng các tính chất thỏa mãn tập FD_R, được phát biểu dưới dạng các tiên đề Armstrong dưới đây.

Cho R là một quan hệ trên Ω, khi đó:

1. Nếu $Y \subseteq X \subseteq \Omega$ thì $X \to Y \in \mathrm{FD}_R$
2. Nếu $X \to Y \in \mathrm{FD}_R$ thì $X \to XY \in \mathrm{FD}_R$
3. Nếu $X \to Y,\ Y \to Z \in \mathrm{FD}_R$ thì $X \to Z \in \mathrm{FD}_R$
4. Nếu $X \to Y,\ X \to Z \in \mathrm{FD}_R$ thì $X \to YZ \in \mathrm{FD}_R$
5. Nếu $X \to Y \in \mathrm{FD}_R$ thì $X \to Y - X \in \mathrm{FD}_R$
6. Nếu $X \to Y \in \mathrm{FD}_R$, $X \subseteq U \subseteq \Omega$ và $V \subseteq XY$ thì $U \to V \in \mathrm{FD}_R$
7. Nếu $X \to Y,\ X' \to Z \in \mathrm{FD}_R$, $X' \subseteq XY$, $X \subseteq U \subseteq A$ và $V \subseteq ZU$ thì $U \to V \in \mathrm{FD}_R$

2.2. Logic Paredaens

Logic Paredaens được gọi là $\mathbf{L_{Par}}$ cho phép đặc tả hình thức và thao tác các phụ thuộc hàm.

Định nghĩa 2 (*Ngôn ngữ* $\mathcal{L}_{\mathbf{Par}}$). Cho Ω là tập vô hạn đếm được các nguyên tử (atoms) và $\to$ là một liên kết nhị phân (binary connective), ta định nghĩa ngôn ngữ:

$$\mathcal{L}_{\mathbf{Par}} = \{X \to Y \mid X, Y \in 2^{\Omega} \text{ và } X \neq \varnothing\}$$

Bây giờ, hệ tiên đề S_{Par} được đưa vào như sau:

Định nghĩa 3. $\mathbf{L_{Par}}$ là logic được cho bởi cặp ($\mathcal{L}_{\mathbf{Par}}$, $\boldsymbol{S_{Par}}$) trong đó $\boldsymbol{S_{Par}}$ là một lược đồ tiên đề ⌊Ax_{Par}⌋: $\vdash_{S_{Par}} X \to Y$ nếu $Y \subseteq X$ và các quy tắc suy diễn sau:

(Kí hiệu $F \vdash_{S_{Par}} F'$ nghĩa là tập phụ thuộc hàm F' được suy diễn logic từ tập phụ thuộc hàm F theo hệ tiên đề S_{Par})

⌊Trans⌋ $X \to Y,\ Y \to Z \vdash_{S_{Par}} X \to Z$ (quy tắc bắc cầu)

⌊Augm⌋ $X \to Y \vdash_{S_{Par}} X \to XY$ (quy tắc gia tăng)

Trong $\mathbf{S_{Par}}$ ta có các quy tắc suy diễn sau:

⌊Union⌋ $X \to Y,\ X \to Z \vdash_{S_{Par}} X \to YZ$ (quy tắc hợp)

⌊Comp⌋ $X \to Y,\ W \to Z \vdash_{S_{Par}} XW \to YZ$ (quy tắc hợp thành)

⌊Inters⌋ $X \to Y,\ X \to Z \vdash_{S_{Par}} X \to Y \cap Z$ trong đó $Y \cap Z \neq \varnothing$ (quy tắc giao)

⌊Reduc⌋ $X \to Y \vdash_{S_{Par}} X \to Y - Z$ trong đó $Y - Z \neq \varnothing$ (quy tắc rút gọn)

⌊Frag⌋ $X \to YZ \vdash_{S_{Par}} X \to Y$ (quy tắc phân mảnh)

⌊gAug⌋ $X \to Y \vdash_{S_{Par}} U \to V$ trong đó $X \subseteq U$ và $V \subseteq XY$ (quy tắc gia tăng suy rộng)

⌊gTrans⌋ $X \to Y,\ Z \to U \vdash_{S_{Par}} V \to W$ trong đó $Z \subseteq XY$, $X \subseteq V$ và $W \subseteq UV$ (quy tắc bắc cầu suy rộng)

Nhận xét 1. Có thể xem logic Paredaens là sự mô tả chi tiết hơn của hệ tiên đề Armstrong, với việc bổ sung các quy tắc hợp thành, quy tắc giao và quy tắc phân mảnh.

Nhận xét 2. Dễ thấy rằng logic Paredaens cũng như các logic khác cho phụ thuộc hàm ([3]. [4], [5], [6]) đều có cùng cấu trúc trong cú pháp, ngữ nghĩa và hệ tiên đề và do đó chúng tương đương nhau.

Nhận xét 3. Để đơn giản trong thao tác với các phụ thuộc hàm cũng như suy diễn ra các phụ thuộc hàm mới từ một tập các phụ thuộc hàm cho trước, ta có thể sử dụng một hệ tiên đề tương đương với hệ tiên đề Armstrong như đã làm trong [7] bao gồm 3 tiên đề sau với X, Y, Z là các tập con của Ω:

A_1. Nếu $Y \subseteq X$ thì $X \to Y$ (quy tắc phản xạ)

A_2. Nếu $X \to Y$ thì $XZ \to YZ$ (quy tắc gia tăng)

A_3. Nếu $X \to Y$ và $Y \to Z$ thì $X \to Z$ (quy tắc bắc cầu)

với việc ngầm hiểu các quy tắc suy diễn khác là những quy tắc suy diễn dễ dàng được suy từ hệ tiên đề Armstrong với ba tiên đề A_1, A_2, A_3.

Trong phần dưới đây, ta hình thức hóa khái niệm dư thừa liên quan tới một tập F các phụ thuộc hàm cho trước trên Ω.

Định nghĩa 4. Cho $F \subseteq \mathcal{L}_{\mathbf{Par}}$ và $f = X \to Y \in F$.

Ta nói f là dư thừa (không cần thiết) trong F nếu $F \setminus \{f\} \vdash_{S_{Par}} f$

Ta nói f là l-dư thừa trong F nếu tồn tại $Z \neq \varnothing$, $Z \subsetneq X$ sao cho

$$(F \setminus \{f\}) \cup \{(X - Z) \to Y\} \vdash_{S_{Par}} f$$

Ta nói f là r-dư thừa trong F nếu tồn tại $U \neq \varnothing$, $U \subsetneq Y$ sao cho

$$(F \setminus \{f\}) \cup \{X \to (Y - U)\} \vdash_{S_{Par}} f$$

Ta nói F có dư thừa nếu nó có chứa một phần tử hoặc là dư thừa hoặc là l-dư thừa hoặc là r-dư thứa trong F.

Sau đây, không giảm tổng quát, ta chỉ xét các tập phụ thuộc hàm F, trong đó, các phụ thuộc hàm thuộc F đều có vế trái và vế phải rời nhau, có nghĩa với mọi phụ thuộc hàm $X \to Y \in F$ ta có $X \cap Y = \varnothing$.

Một tập các phụ thuộc hàm có tính chất như vậy được gọi là một tập phụ thuộc hàm được thu gọn (reduced functional dependencies set).

Trong [1] có phát biểu và chứng minh định lý sau:

Định lý 1 (*Định lý 6 trong [1] và là Định lý 1 trong [2]*)

Cho $X \to Y$, $U \to V \in \mathrm{L_{FD}}$ với $X \cap Y = \varnothing$.

(a). Nếu $X \subseteq U$ thì $\{X \to Y, U \to V\} \equiv_{S_{Par}} \{X \to Y, (U - Y) \to (V - Y)\}$ (1)

Do đó, nếu $U \cap Y \neq \varnothing$ hay $V \cap Y = \varnothing$ thì $U \to V$ theo thứ tự là l-dư thừa hay r-dư thừa trong $\{X \to Y, U \to V\}$.

(b). Nếu $X \nsubseteq U$ và $X \subseteq UV$ thì $\{X \to Y, U \to V\} \equiv_{S_{Par}} \{X \to Y, U \to (V - Y)\}$ (2)

Do đó, nếu $V \cap Y \neq \varnothing$ thì $U \to V$ là r-dư thừa trong $\{X \to Y, U \to V\}$.

(Kí hiệu $F \equiv_{S_{Par}} F'$ nghĩa là tập phụ thuộc hàm F' tương đương với tập phụ thuộc hàm F theo hệ tiên đề S_{Par}, từ F có thể suy ra F' và ngược lại)

Chứng minh của Định lý 1 (*là Định lý 6 trong [1]*) được trình bày lại đầy đủ như sau:

Chứng minh.

(a). $\Rightarrow$

1. $X \to Y$ (giả thiết)
2. $(U - Y) \to Y$ (1, gia tăng suy rộng)
3. $(U - Y) \to (U - Y)$ Ax_{FD} (tiên đề phản xạ)
4. $(U - Y) \to UY$ (2, 3, quy tắc hợp)
5. $(U - Y) \to U$ (4, gia tăng suy rộng)
6. $U \to V$ (giả thiết)
7. $(U - Y) \to V$ (5, 6, bắc cầu suy rộng)
8. $(U - Y) \to (V - Y)$ (7, gia tăng suy rộng)

(a). $\Leftarrow$

1. $U \to X$ Ax_{FD} (tiên đề phản xạ)
2. $X \to Y$ (giả thiết)
3. $U \to Y$ (1, 2, bắc cầu suy rộng)
4. $(U - Y) \to (V - Y)$ (giả thiết)
5. $U \to VY$ (3, 4, quy tắc hợp)
6. $U \to V$ (2, 5, gia tăng suy rộng)

(b). $\Rightarrow$

1. $U \to V$ (giả thiết)
2. $U \to (V - Y)$ (1, gia tăng suy rộng)

(b). $\Leftarrow$

1. $U \to X$ Ax_{FD} (tiên đề phản xạ)
2. $X \to Y$ (giả thiết)
3. $U \to Y$ (1, 2, bắc cầu suy rộng)
4. $U \to (V - Y)$ (giả thiết)
5. $U \to VY$ (3, 4, quy tắc hợp)
6. $U \to V$ (2, 5, gia tăng suy rộng)

Cái hay và mới của Định lý 1 là nó cho phép đưa vào hai quy tắc thay thế quan trọng được ký hiệu theo thứ tự là ⌊Subst⌋ và ⌊rSubst⌋

⌊Subst⌋ $X \to Y, U \to V \vdash_{S_{Par}} (U - Y) \to (V - Y)$ nếu $X \subseteq U, X \cap Y = \varnothing$

⌊rSubst⌋ $X \to Y, U \to V \vdash_{S_{Par}} U \to (V - Y)$ nếu $X \nsubseteq U, X \subseteq UV, X \cap Y = \varnothing$

Rõ ràng là không có hệ tiên đề nào cho phụ thuộc hàm có những quy tắc thay thế nói trên, có khả năng phát hiện và loại bỏ dư thừa trong một tập phụ thuộc hàm một cách hiệu quả như vậy.

Dưới đây là một số nhận xét về phần chứng minh của Định lý 1 nêu ở trên.

Nhận xét 4. Trong chứng minh chiều $\Leftarrow$, phần (a) của Định lý 1, các dòng 5. và 6. được viết lại là

5. $U \to VY$ (3, 4, Quy tắc hợp)
6. $U \to V$ (2, 5, Quy tắc gia tăng suy rộng)

Rõ ràng dòng 6. phải sửa lại là

6. $U \to V$ (5, Quy tắc gia tăng suy rộng)

Nhận xét 5. Trong chứng minh chiều $\Leftarrow$, phần (b) của Định lý 1, ta hãy xem dòng 1.

1. $U \to X$ Ax_{FD} (tiên đề phản xạ)

Khẳng định này rõ ràng là sai vì trong phát biểu phần (b) của Định lý 1, ta có các giả thiết $X \not\subseteq U, X \subseteq UV, X \cap Y = \varnothing$.

Do đó, chiều $\{X \to Y, U \to (V - Y)\} \vdash_{S_{Par}} \{X \to Y, U \to V\}$ với các giả thiết nêu trên chưa được chứng minh.

3. MỘT CHỨNG MINH MỚI CHO ĐỊNH LÝ 1

Để đơn giản cách chứng minh Định lý 1, ta sử dụng hệ ba tiên đề tương đương với hệ tiên đề Armstrong với $X, Y, Z \subseteq \Omega$, trong đó, Ω là vũ trụ các thuộc tính.

A_1. Nếu $Y \subseteq X$ thì $X \to Y$ (Tiên đề phản xạ)

A_2. Nếu $X \to Y$ thì $XZ \to YZ$ (Tiên đề gia tăng)

A_3. Nếu $X \to Y$ và $Y \to Z$ thì $X \to Z$ (Tiên đề bắc cầu)

cùng với các quy tắc suy diễn quen thuộc, dễ dàng được suy ra từ hệ ba tiên đề A_1, A_2, A_3 như:

Nếu $X \to Y$ và $U \to V$ thì $XU \to YV$ (Quy tắc hợp)

Nếu $X \to Y$ thì $X \to Z$ với mọi $Z \subseteq Y$ (Quy tắc tách hay phân mảnh)

Sau đây là một chứng minh mới cho Định lý 1.

Trước hết, Định lý 1 được phát biểu lại như sau:

(a). Nếu $X \subseteq U$, $X \cap Y = \varnothing$ thì hai tập phụ thuộc hàm

$$\{X \to Y, U \to V\} \equiv \{X \to Y, (U - Y) \to (V - Y)\}$$

trong đó, $\equiv$ là tương đương theo nghĩa sử dụng hệ quy tắc suy diễn Armstrong, hệ phụ thuộc hàm thứ nhất có thể suy ra hệ phụ thuộc hàm thứ hai và ngược lại.

(b). Nếu $X \not\subseteq U, X \subseteq UV$ thì

$$\{X \to Y, U \to V\} \equiv \{X \to Y, U \to (V - Y\}$$

Chứng minh.

(a). $\Rightarrow$

Vì $X \subseteq U$ nên $X - Y \subseteq U - Y$. Vì $X \cap Y = \varnothing$ nên $X - Y = X \subseteq U - Y$. Từ đó ta có dãy suy diễn sau:

1. $(U - Y) \to X$ (A_1)
2. $X \to Y$ (Giả thiết)
3. $(U - Y) \to Y$ (1, 2, A_3)
4. $(U - Y) \to (U - Y)$ (A_1)
5. $(U - Y) \to UY$ (3, 4, Quy tắc hợp)
6. $(U - Y) \to U$ (5, Quy tắc tách)
7. $U \to V$ (Giả thiết)
8. $(U - Y) \to V$ (6, 7, A_3)
9. $(U - Y) \to (V - Y)$ (8, Quy tắc tách do $(V - Y) \subseteq V$)

(a). $\Leftarrow$

1. $X \to Y$ (Giả thiết)

2. $(U - Y) \to (V - Y)$ (Giả thiết)
3. $U \to X$ (A_1, vì $X \subseteq U$)
4. $U \to Y$ (3, 1, A_3)
5. $U \to VY$ (2, 4, Quy tắc hợp)
6. $U \to V$ (5, Quy tắc tách)

(b). $\Rightarrow$

1. $U \to V$ (Giả thiết)
2. $U \to (V - Y)$ (1, Quy tắc tách)

(b). $\Leftarrow$

1. $X \to Y$ (Giả thiết)
2. $U \to (V - Y)$ (Giả thiết)
3. $U \to U(V - Y)$ (2, A_2)
4. $U(V - Y) \to (UV - Y)$ (A_1)

do có $U \cup (V - Y) \supseteq (U - Y) \cup (V - Y)$ và $(U - Y) \cup (V - Y) = UV - Y$

5. $U \to (UV - Y)$ (3, 4, A_3)
6. $(UV - Y) \to X$ (A_1)

do có $X \subseteq UV$ và $X \cap Y = \varnothing$ nên $X = (X - Y) \subseteq UV - Y$

7. $(UV - Y) \to Y$ (6, 1, A_3)
8. $U \to Y$ (5, 7, A_3)
9. $U \to UVY$ (5, 8, A_2)
10. $U \to V$ (9, Quy tắc tách)

Nhận xét 6. Trong chứng minh mới của Định lý 1, việc chứng minh phần (a) về cơ bản là giống với chứng minh phần (a) trong [1]. Cái khác nhau là ở chỗ cách thức giải thích các bước suy diễn. Trong [1], các tác giả dùng các tiên đề và các quy tắc suy diễn trong logic Paredaens, còn trong chứng minh mới, chúng tôi sử dụng hệ tiên đề quen thuộc của Armstrong, nên việc giải thích các bước suy diễn là đơn giản, rõ ràng hơn.

Để khắc phục lỗi sai trong chứng minh phần (b) của Định lý 1 trong [1], chứng minh phần (b) của chúng tôi ở đây là hoàn toàn mới. Nó khiến cho Định lý 1 trong [1], một Định lý rất hay, là nền tảng cho phép biến đổi tiền xử lý loại bỏ hiệu quả các dư thừa trong một tập phụ thuộc hàm cho trước, đứng vững và sử dụng được.

Nhận xét 7. Trong thực hành, trong nhiều trường hợp, để đơn giản hơn, ta có thể dùng quy tắc thay thế sau:

Cho hệ hai phụ thuộc hàm $\{X \to Y, U \to V\}$. Nếu $X \not\subseteq U$, $X \subseteq V$ và $X \cap Y = \varnothing$ thì, do tương đương, có thể thay thế $\{X \to Y, U \to V\}$ bằng hệ hai phụ thuộc hàm $\{X \to Y, U \to (V - Y)\}$ nói chung đơn giản hơn. Nói cách khác, nếu $X \not\subseteq U$, $X \subseteq V$ và $X \cap Y = \varnothing$ thì

$$\{X \to Y, U \to V\} \equiv \{X \to Y, U \to (V - Y)\} \qquad (3)$$

Điều này hiển nhiên đúng vì nếu $X \not\subseteq U$, $X \subseteq V$ và $X \cap Y = \varnothing$ thì đương nhiên $X \not\subseteq U$, $X \subseteq UV$ và $X \cap Y = \varnothing$ và ta rơi vào trường hợp (b) của Định lý 1.

Nhận xét 8. Trên cơ sở các phép thay thế (1), (2), (3), ta có thể làm đơn giản hơn thủ tục **removeRedundancy** trong [1] bằng thủ tục ***Loại bỏ dư thừa*** cho các tập phụ thuộc hàm F ở dạng thu gọn gồm các bước sau:

Procedure ***Loại bỏ dư thừa***
Input: F (Một tập phụ thuộc hàm ở dạng thu gọn)

Output: *F'* (Một tập phụ thuộc hàm tương đương với *F* với dư thừa ít hơn)
Begin
Repeat
B1. Thực hiện các phép hợp cho các phụ thuộc hàm có cùng vế trái;
B2. Thực hiện các phép thay thế (1), (2), (3);
Until (không thực hiện các thao tác B1 và B2 thêm được nữa);
B3. Kiểm tra xem trong tập phụ thuộc hàm thu được, có phụ thuộc hàm nào được suy ra từ hai phụ thuộc hàm khác từ việc áp dụng (A_3). Nếu có thì loại bỏ nó.
End;

Nhận xét 9. Trong [1] và [2], các tác giả đã cho chạy thủ tục **removeRedundancy** trên nhiều tập phụ thuộc hàm với số lượng và kích thước khác nhau và đã thấy rằng tỷ lệ phần trăm số lần áp dụng các quy tắc thay thế là rất cao và tăng đáng kể với độ phức tạp của các tập phụ thuộc hàm.

Ngoài ra, các tác giả của [1] và [2] đã rút ra các kết luận tổng quát sau:

- Đối với 28,25% các tập phụ thuộc hàm, không cần thiết áp dụng quy tắc bắc cầu (A_3) và phép biến đổi tiền xử lý loại bỏ dư thừa một cách hiệu quả.
- Kích thước của các tập phụ thuộc hàm được rút gọn tới 52,89%
- Khi số các thuộc tính tăng lên thì số trường hợp trong đó không cần áp dụng quy tắc bắc cầu (A_3) cũng tăng lên. Điều này chứng tỏ quy tắc thay thế đặc biệt thích hợp để làm việc với các lược đồ cơ sở dữ liệu lớn.
- Số phần trăm các áp dụng của quy tắc thay thế không phụ thuộc vào số thuộc tính và độ dài của phụ thuộc hàm.

Nhận xét 10. Để thấy được ý nghĩa và ưu việt của các quy tắc thay thế (tức phép biến đổi tiền xử lý các tập phụ thuộc hàm), ta xét hai ví dụ sau, trong đó, ví dụ 1 được lấy lại từ ví dụ 1 trong [2] với việc chỉnh sửa lại một sai sót nhỏ.

Ví dụ 1 ([2]). Cho $F = \{a \to bc, ab \to ce, bd \to ac, af \to b, cd \to ba\}$. Ta có thể áp dụng các phép thay thế để thu được một tập phụ thuộc hàm với dư thừa ít hơn.

Như vậy, sau khi thực hiện phép biến đổi tiền xử lý, ta thu được tập *F'* tương đương với *F* nhưng chứa ít dư thừa hơn.

$$F' = \{a \to bce, bd \to a, cd \to b\}.$$

Ví dụ 2. Áp dụng các phép thay thế đối với tập phụ thuộc hàm $F = \{b \to a, b \to gh, d \to a, bi \to h, ab \to de, ab \to fg, abc \to dj, abc \to k\}$.

Quy tắc áp dụng	*F*
⌊Quy tắc hợp⌋: $b \to a, b \to gh \vdash_{S_{Par}} b \to agh$	$b \to agh$, $d \to a$, $bi \to h$, $ab \to de$, $ab \to fg$, $abc \to dj$, $abc \to k$
⌊Quy tắc hợp⌋: $ab \to de, ab \to fg \vdash_{S_{Par}} ab \to defg$	$b \to agh$, $d \to a$, $bi \to h$, $ab \to defg$, $abc \to dj$, $abc \to k$
⌊Quy tắc hợp⌋: $abc \to dj, abc \to k \vdash_{S_{Par}} abc \to djkh$	$b \to agh$, $d \to a$, $bi \to h$, $ab \to defg$, $abc \to djk$
⌊Subst⌋:	$b \to agh$, $d \to a$, $bi \to \varnothing$, $ab \to defg$,

$b \to agh,\ bi \to h \vdash_{S_{Par}} bi \to \varnothing$	$abc \to djk$
⌊A_1⌋: $\vdash_{S_{Par}} bi \to \varnothing$ (sẽ được loại bỏ)	$b \to agh,\ d \to a,\ ab \to defg,\ abc \to djk$
⌊Subst⌋: $b \to agh,\ ab \to defg \vdash_{S_{Par}} b \to def$	$b \to agh,\ d \to a,\ b \to def,\ abc \to djk$
⌊Quy tắc hợp⌋: $b \to agh,\ b \to def \vdash_{S_{Par}} b \to adefgh$	$b \to adefgh,\ d \to a,\ abc \to djk$
⌊Subst⌋: $b \to adefgh,\ abc \to djk \vdash_{S_{Par}} bc \to jk$	$b \to adefgh,\ d \to a,\ bc \to jk$
⌊rSubst⌋: $d \to a,\ b \to adefgh \vdash_{S_{Par}} b \to defgh$	$b \to defgh,\ d \to a,\ bc \to jk$

Như vậy, cuối cùng ta thu được tập $F' = \{b \to defgh,\ d \to a,\ bc \to jk\}$ tương đương với F nhưng chứa ít dư thừa hơn.

4. KẾT LUẬN

Phép biến đổi tiền xử lý để loại bỏ dư thừa trong các tập phụ thuộc hàm được trình bày trong [1] và [2] là mới và tỏ ra rất hiệu quả. Cơ sở của phép biến đổi tiền xử lý này là Định lý 1 (trong [1]) và cũng là Định lý 6 (trong [2]).

Đáng tiếc là chứng minh phần (b) của Định lý 1 là sai và không chấp nhận được. Trong bài báo này, chúng tôi đã đưa ra một chứng minh mới cho Định lý 1, cũng như đưa ra một quy tắc thay thế mới đơn giản và dễ áp dụng trong thực hành. Điều này khiến cho Định lý 1 ([1], [2]) đứng vững và áp dụng được.

Xây dựng thêm các quy tắc thay thế mới cho việc tiền xử lý các tập phụ thuộc hàm cũng là một hướng nghiên cứu đáng quan tâm.

TÀI LIỆU THAM KHẢO

[1]. P. Cordero et al., "*SL_{FD} Logic: Elimination of data redundancy in Knowledge Representation*", Advances in Artificial Intelligence, IBERAMIA 2002, LNAI 2527, pp.141-150, 2002.

[2]. Ángel Mora et al., "*An Efficient Preprocessing Transformation for Functional Dependencies Sets Based on the Substitution Paradigm*", R. Conejo et al. (Eds.): CAEPIA - TTIA 2003, LNAI 3040, pp. 136-146, 2004.

[3]. P. Atzeni and V.D.Antonellis, "*Relational Database Theory*", The Benjamin/Cummings Publishing Company Inc, 1993.

[4]. R. Fagin, "*Functional dependencies in a Relational Database and Propositional Logic*", IBM Journal of Research and Development 21(6), pp. 534-544, 1977.

[5]. T. Ibaraki et al, "*Functional dependencies in Horn theories*", Artificial Intelligence 108(1-2), pp. 1-30, 1999.

[6]. J. D. Ullman, "*Database and Knowledge-base Systems*", Computer Science Press, 1988.

[7]. Ho Thuan and Le Van Bao, "*Some results about keys of relational schemas*", Acta Cybernetica, Tom 7, Fasc. 1, Szeged, pp. 99-113, 1985.

ABSTRACT

ABOUT AN EFFICIENT PREPROCESSING TRANSFORMATION FOR FUNCTIONAL DEPENDENCIES SETS

In [1] and [2], Ángel Mora et al designed a preprocessing transformation using the substitution paradigm of SL_{FD} logic to eliminate data redundancy in a given set of functional dependencies in order to obtain an equivalent set of functional dependencies with a smaller size in polynomial time. The basis and correctness of this preprocessing transformation have been proved by Theorem 6 in [1]. In this paper, we show a serious error in the proof of Theorem 6 and give a simpler and correct proof for that theorem. Some remarks about the preprocessing transformation are also given.

Keywords: Relational database, Relation schema, Functional dependency, Preprocessing transformation.

Nhận bài ngày 01 tháng 6 năm 2017
Hoàn thiện ngày 24 tháng 7 năm 2017
Chấp nhận đăng ngày 18 tháng 8 năm 2017

Địa chỉ: [1] Trường Cao đẳng Hải Dương;
[2] Viện Công nghệ thông tin - Viện HLKH-CNVN.
[*] Email: vqtuanhd@gmail.com.