Tạp chí Tin học và Điều khiển học, T. 17, S. 2 (2001), 65–74

MỘT SỐ VẤN ĐỀ PHỦ DẠNG VÀNH VÀ CÁC KHÁI NIỆM LIÊN QUAN

PHẠM QUANG TRUNG

Abstract. Maier [2] gave concept about *annular cover* in 1983, and applied it in algorithm SYNTHESIZE, which only was an orientation. In this paper we present new results on annular cover and related concepts. These results are applied in algorithm THV.

Tóm tắt. Maier [2] đã đưa ra khái niệm *phủ dạng vành* từ năm 1983 và khái niệm này đã được ứng dụng trong Thuật toán SYNTHESIZE với những vấn đề còn để mở. Trong bài báo này chúng tôi đưa ra một số kết quả mới về phủ dạng vành và các khái niệm liên quan. Những kết quả này là cơ sở của Thuật toán THV.

1. MỞ ĐẦU

Trong lý thuyết cơ sở dữ liệu, khái niệm *phủ dạng vành* (annular cover) được Maier [2] nêu ra từ năm 1983, tuy nhiên khái niệm này còn ít được quan tâm sử dụng vì là một khái niệm khá phức tạp, ít quen thuộc và việc ứng dụng bước đầu chỉ được trình bày trong Thuật toán SYNTHESIZE với những vấn đề còn để mở. Chúng tôi đã chứng minh một số kết quả về phủ dạng vành và các khái niệm liên quan, những kết quả này là cơ sở của Thuật toán THV [3] do chúng tôi đề xuất.

Kí hiệu: Quan hệ R trên tập thuộc tính U được kí hiệu là $R(U)$; hợp của hai tập thuộc tính X, Y được viết là XY. Các thuật toán được viết dưới dạng ngôn ngữ Pascal.

Mục này chỉ nêu một số khái niệm và kết quả liên quan, bạn đọc nếu cần quan tâm chi tiết hơn thì xem [1, 2, 4].

Định nghĩa 1. Cho $R(A_1, A_2, \dots, A_n)$ là một lược đồ quan hệ, cho X và Y là các tập con của $\{A_1, A_2, \dots, A_n\}$. Chúng ta nói $X \to Y$ (đọc là X *xác định hàm* Y" hay "Y *phụ thuộc hàm vào* X") nếu với mọi quan hệ r là thể hiện của R, thì trong r không thể có hai bộ trùng nhau trên các thành phần của mọi thuộc tính trong tập X mà lại không trùng nhau trên một hay nhiều hơn các thành phần của các thuộc tính của tập Y.

- Quan hệ r *thỏa* phụ thuộc hàm (function dependency - FD) $X \to Y$, nếu với mọi cặp bộ μ, ν trong r sao cho $\mu[X] = \nu[X]$ thì $\mu[Y] = \nu[Y]$ cũng đúng. Nếu r không thỏa $X \to Y$, thì r *vi phạm* phụ thuộc đó.

- Cho F là tập phụ thuộc hàm của lược đồ quan hệ R, và cho $X \to Y$ là một phụ thuộc hàm. Chúng ta nói F *suy diễn logic* ra $X \to Y$, viết là $F \models X \to Y$, nếu với mọi quan hệ r của R mà thỏa các phụ thuộc hàm trong F thì cũng thỏa $X \to Y$.

Định nghĩa 2. Bao đóng của tập phụ thuộc hàm F, ký hiệu là F^+, là tập các phụ thuộc hàm được suy diễn logic từ F, nghĩa là: $F^+ = \{X \to Y \mid F \models X \to Y\}$.

Định nghĩa 3. Cho lược đồ quan hệ R với tập phụ thuộc hàm F, cho X là một tập con của R.

a) Khi đó X^+, *bao đóng* của X (đối với F) là tập các thuộc tính A, sao cho $X \to A$ có thể được suy dẫn từ F bởi hệ tiên đề Armstrong, tức là: $X^+ = \{A \mid F \models X \to A\}$.

b) Tập X được gọi là *khóa* (key) của lược đồ quan hệ R nếu:

(1) $X \to R \in F^+$.

(2) Với $\forall Y \subset X$ thì $Y \not\to R$.

Tập X nếu chỉ thỏa điều kiện (1) được gọi là một *siêu khóa* (superkey). Các khóa (hay siêu khóa) được liệt kê rõ ràng cùng với lược đồ quan hệ được gọi là các *khóa chỉ định* (designated key).

Định nghĩa 4. Hai tập phụ thuộc hàm F và G trên lược đồ R là *tương đương* (equivalent), ký hiệu là $F \equiv G$, nếu: $F^+ = G^+$. Nếu $F \equiv G$ thì F là một *phủ* (cover) của G.

Phụ thuộc hàm $X \to Y \in F$ là *dư thừa* nếu $F - \{X \to Y\} \models X \to Y$.

Định nghĩa 5. Tập phụ thuộc hàm F là *cực tiểu* (minimum) nếu không có tập phụ thuộc hàm bất kỳ tương đương với F lại có ít hơn số lượng phụ thuộc hàm.

Một tập phụ thuộc hàm F là cực tiểu thì cũng là không dư thừa.

Thuộc tính A được gọi là thuộc tính dư thừa trong phụ thuộc hàm $X \to Y$ thuộc tập phụ thuộc hàm F, nếu A có thể được loại bỏ khỏi vế trái hay vế phải của $X \to Y$ mà không làm thay đổi bao đóng của F.

Ký hiệu: a là số lượng thuộc tính khác nhau trong F, p là số lượng phụ thuộc hàm trong F, thì $n = ap$ là *độ dài của dữ liệu vào*, tức là số lượng ký hiệu cần để viết F.

Trong [1] đã chứng minh sự đúng đắn của thuật toán sau đây:

Thuật toán MINCOVER (tên của thuật toán này do tác giả đặt).

VÀO: Tập phụ thuộc hàm $F = \{X_i \to Y_i \mid i = 1, 2, \ldots, p\}$

RA: Phủ cực tiểu G.

MINCOVER(F)

begin

$G := \{X_i \to X_i^+ \mid i = 1, 2, \ldots, p\}$;

return(NONREDUN(G));

end.

Độ phức tạp tính toán theo thời gian của thuật toán MINCOVER chính là độ phức tạp tính toán theo thời gian của Thuật toán NONREDUN [2], là $O(np)$.

Định nghĩa 6. Hai tập thuộc tính X và Y là *tương đương* với nhau trên tập phụ thuộc hàm F, nếu $F \models X \to Y$ và $F \models Y \to X$ (ký hiệu là $X \leftrightarrow Y$).

Cho F là tập phụ thuộc hàm trên lược đồ R và tập thuộc tính $X \subseteq R$, ký hiệu $E_F(X)$ là tập phụ thuộc hàm trong F có các vế trái tương đương với X. Ký hiệu $\overline{E}_F$ là tập hợp: $\{E_F(X) \mid X \subseteq R$ và $E_F(X) \neq \emptyset\}$. Nếu trong F không tồn tại phụ thuộc hàm có vế trái tương đương với X thì $E_F(X)$ rỗng. Tập $\overline{E}_F(X)$ là một phân hoạch (partition) của tập F.

Định nghĩa 7. *Phụ thuộc hàm phức hợp* (compound functional dependency - CFD) có dạng $(X_1, X_2, \ldots, X_k) \to Y$, trong đó X_1, X_2, ..., X_k và Y là các tập con khác nhau của lược đồ R. Quan hệ $r(R)$ thỏa phụ thuộc hàm phức hợp $(X_1, X_2, \ldots, X_k) \to Y$ nếu nó thỏa các phụ thuộc hàm $X_i \to X_j$ và $X_i \to Y$, với $1 \le i, j \le k$. Trong phụ thuộc hàm phức hợp này, $(X_1, X_2, \ldots, X_k)$ được gọi là vế trái, X_1, X_2, ..., X_k là các tập trái, Y là vế phải.

Phụ thuộc hàm phức hợp là cách viết rút gọn hơn tập các phụ thuộc hàm có các vế trái tương đương. Trong trường hợp nếu $Y = \emptyset$, có dạng đặc biệt của phụ thuộc hàm phức hợp là $(X_1, X_2, \ldots, X_k)$.

Định nghĩa 8. Giả sử G là tập các phụ thuộc hàm phức hợp trên R và F là tập các phụ thuộc hàm hay các phụ thuộc hàm phức hợp trên R. Tập G *tương đương* với tập F, ký hiệu là $G \equiv F$, nếu mỗi quan hệ $r(R)$ thỏa G thì thỏa F và ngược lại.

Định nghĩa 9. Tập F được gọi là *phủ* của G nếu $F \equiv G$, trong đó F và G bao gồm hoặc là tập các phụ thuộc hàm, tập các phụ thuộc hàm phức hợp, hoặc là tập hợp chỉ gồm một loại phụ thuộc.

Định nghĩa 10. Tập phụ thuộc hàm F được gọi là *tập đặc trưng* (characteristic set) đối với phụ thuộc hàm phức hợp $(X_1, X_2, \ldots, X_k) \to Y$, nếu $F \equiv \{(X_1, X_2, \ldots, X_k) \to Y\}$. Nếu mỗi tập hợp trái của phụ thuộc hàm phức hợp được sử dụng với tư cách là vế trái của phụ thuộc hàm đúng một lần (nghĩa là F có dạng $\{X_1 \to Y_1, X_2 \to Y_2, \dots, X_k \to Y_k\}$), thì F được gọi là *tập đặc trưng tự nhiên* (natural characteristic set) đối với phụ thuộc hàm phức hợp đã cho.

Định nghĩa 11. Tập phụ thuộc hàm phức hợp F được gọi là *dạng vành* (annular), nếu không có các tập trái X và Z trong các vế trái khác nhau mà $X \leftrightarrow Z$ trên F.

Định nghĩa 12. Cho G là tập phụ thuộc hàm phức hợp chứa phụ thuộc $X_1, X_2, \dots, X_k) \to Y$. Cho X_i là một trong các tập trái và A là một thuộc tính trong X_i. Thuộc tính A được gọi là *có thể chuyển dịch* (shiftable) nếu A có thể được chuyển từ X_i sang Y mà vẫn bảo toàn sự tương đương. Tập trái X_i là *có thể chuyển dịch*, nếu mọi thuộc tính của X_i là có thể chuyển dịch đồng thời.

Định nghĩa 13. Phủ dạng vành G là *không dư thừa*, nếu không thể loại được một phụ thuộc hàm phức hợp nào khỏi G mà không vi phạm sự tương đương, ngoài ra, không có một phụ thuộc hàm phức hợp nào trong G chứa các tập trái có thể chuyển dịch. Trong trường hợp ngược lại thì G là dư thừa.

Bổ đề 1. *Cho G là tập phụ thuộc hàm phức hợp dạng vành không dư. Sự hợp nhất các tập đặc trưng tự nhiên của tất cả các phụ thuộc hàm phức hợp trong G tạo thành tập phụ thuộc hàm không dư tương đương với G.*

Định nghĩa 14. Cho G là tập dạng vành không dư. Phụ thuộc hàm phức hợp $(X_1, X_2, \dots, X_k) \to Y$ trong G được gọi là *rút gọn*, nếu các tập trái không có thuộc tính có thể chuyển dịch, còn các vế phải không có thuộc tính dư thừa. Tập G là rút gọn nếu mọi phụ thuộc hàm phức hợp trong G là rút gọn.

Định nghĩa 15. Cho G là tập dạng vành không dư. Tập G là *cực tiểu* nếu không có tập dạng vành bất kỳ tương đương lại có ít hơn số lượng tập trái.

2. MỘT SỐ KẾT QUẢ

Thí dụ 1. Cho tập phụ thuộc hàm $F = \{A \to AB,\ B \to ACD,\ AE \to IJ\}$. Tập $G = \{(A, AB, B) \to CD,\ (AE) \to IJ\}$ là phủ dạng vành đối với tập F. Tập $G' = \{(A, B) \to ABCD,\ (AE) \to IJ\}$ cũng là phủ dạng vành đối với tập F.

Như vậy, có thể có nhiều tập dạng vành tương đương đối với 1 tập phụ thuộc hàm cho trước.

Định nghĩa 16. Cho F là tập phụ thuộc hàm. Cho G là tập dạng vành tương đương với F và các vế trái của F tương ứng một-một là các tập trái của G, thì G là *phủ dạng vành đầy đủ* (completely annular cover) đối với tập F.

Trong Thí dụ 1 có tập G' là phủ dạng vành đầy đủ đối với F, còn tập G không phải là phủ dạng đầy đủ đối với F.

Khái niệm phủ dạng vành đầy đủ đối với một tập phụ thuộc hàm nhằm mục đích xác lập được một lớp phủ dạng vành có sự đồng nhất nguyên vẹn các tập trái với các vế trái của tập phụ thuộc hàm cho trước. Do đó trường hợp phủ dạng vành đầy đủ đối với phủ không dư, phủ tối thiểu, phủ cực tiểu, phủ rút gọn trái, hay phủ đã gộp các phụ thuộc hàm có vế trái giống nhau của tập phụ thuộc hàm cho trước..., trong những trường hợp cụ thể, sẽ được nêu rõ ràng. Những trường hợp này phân biệt với các trường hợp: phủ dạng vành đầy đủ và (có tính chất) không dư, phủ dạng vành đầy đủ và (có tính chất) tối thiểu,...

Thuật toán tạo phủ dạng vành đầy đủ đối với tập phụ thuộc hàm cho trước là sự thực hiện so sánh các bao đóng của các vế trái (của các phụ thuộc hàm) có độ phức tạp tính toán theo thời gian căn cứ theo thuật toán tính bao đóng của p tập thuộc tính vế trái nên là $O(np)$ (việc tính bao đóng sử dụng Thuật toán LINCLOSURE [2] có độ phức tạp tính toán theo thời gian là $O(n)$.

Thuật toán COANCOVER

VÀO: Tập phụ thuộc hàm $F = \{X_i \to Y_i \mid i = 1, 2, \dots, p\}$.

RA: Tập G là phủ dạng vành đầy đủ đối với F.

COANCOVER(F)

begin

for mỗi phụ thuộc hàm $X_i \rightarrow Y_i \in F$ **do**

$E_F(X_i) := \{X_j \rightarrow Y_j \mid X_i \leftrightarrow X_j, \forall X_j \rightarrow Y_j \in F\}$;

$\overline{E}_F := \{E_F(X_i) \mid i = 1, 2, \ldots, p\}$;

//ký hiệu CF^t là phụ thuộc hàm phức hợp thứ t.

//CF^t có dạng: - Vế trái gồm các tập trái là các X_j thuộc $E_F(X_i)$.

// - Vế phải là hợp của các Y_j thuộc $E_F(X_i)$.

$G := \{CF^t \mid t = 1, 2, \ldots, |\overline{E}_F|\}$;

return(G);

end.

Dễ dàng khẳng định được hai kết quả (các bổ đề 2 và 3) sau đây.

Bổ đề 2. *Thuật toán COANCOVER xác định đúng phủ dạng vành đầy đủ đối với tập phụ thuộc hàm cho trước.*

Bổ đề 3. *Thuật toán COANCOVER có độ phức tạp tính toán theo thời gian là $O(np)$.*

Định lý 1. *Cho G là phủ dạng vành đầy đủ đối với tập phụ thuộc hàm F, thì G là cực tiểu nếu và chỉ nếu F là cực tiểu.*

Chứng minh. a) (*Điều kiện cần*). Theo Định nghĩa 16 thì số lượng tập trái của G bằng số lượng phụ thuộc hàm của F. Giả sử F không phải là cực tiểu.

Ký hiệu F' là tập phụ thuộc hàm cực tiểu tương đương với F, thì F' có số lượng phụ thuộc hàm ít hơn F. Gọi G' là tập dạng vành đầy đủ đối với F', thì số lượng tập trái của G' bằng số lượng phụ thuộc hàm của F' (theo Định nghĩa 16) nên ít hơn số lượng tập trái của G. Đây là điều mâu thuẫn vì như thế thì G không phải là tập dạng vành cực tiểu.

b) (*Điều kiện đủ*). Giả sử G không là phủ dạng vành cực tiểu, ký hiệu G' là phủ dạng vành cực tiểu tương đương với F. Ký hiệu tập phụ thuộc hàm F' là hợp nhất các tập đặc trưng tự nhiên của tất cả các phụ thuộc hàm phức hợp trong phủ dạng vành cực tiểu G', theo Bổ đề 1 thì F' là không dư và tương đương với G', theo Định nghĩa 10 thì số lượng phụ thuộc hàm của F' bằng số lượng tập trái của G'. Nhưng vì số lượng vế trái của F bằng số lượng tập trái của G theo cách xây dựng G và G' có số lượng tập trái ít hơn G, nên F' có số lượng phụ thuộc hàm ít hơn. Đây là điều mâu thuẫn, vì thế F không phải là tập phụ thuộc hàm cực tiểu. □

Qui ước: Để ngắn gọn, thuật ngữ "phủ dạng vành đầy đủ và cực tiểu đối với tập phụ thuộc hàm F là để chỉ "phủ dạng vành đầy đủ và (có tính chất) cực tiểu đối với phủ cực tiểu của tập phụ thuộc hàm F".

Căn cứ vào Định lý 1 và Thuật toán MINCOVER hoàn toàn khẳng định được sự đúng đắn của Thuật toán MINCOANCOVER sau đây để tìm phủ dạng vành đầy đủ, cực tiểu đối với tập phụ thuộc hàm cho trước, là sự phối hợp của Thuật toán COANCOVER và Thuật toán MINCOVER.

Thuật toán MINCOANCOVER

VÀO: Tập phụ thuộc hàm $F = \{X_i \rightarrow Y_i \mid i = 1, 2, \ldots, p\}$.

RA: Tập G là phủ dạng vành đầy đủ, cực tiểu đối với F.

MINCOANCOVER(F)

begin

$G :=$ COANCOVER(MINCOVER(F);

return(G);

end.

Bổ đề 4. *Thuật toán MINCOANCOVER xác định đúng phủ dạng vành đầy đủ, cực tiểu đối với tập phụ thuộc hàm cho trước.*

Bổ đề 5. *Thuật toán MINCOANCOVER có độ phức tạp tính toán theo thời gian là* $O(np)$.

Chứng minh. Độ phức tạp tính toán theo thời gian của Thuật toán MINCOANCOVER là tổng độ phức tạp tính toán theo thời gian của Thuật toán MINCOVER (là $O(np)$) và độ phức tạp tính toán theo thời gian của Thuật toán COANCOVER (là $O(np)$), nên là $O(np)$. □

Vấn đề xác định được phủ dạng vành rút gọn, cực tiểu đối với tập phụ thuộc hàm cho trước không có thuật toán trong [2] và đây là việc không đơn giản, như thí dụ dưới đây chứng tỏ: không thể bằng cách nhóm các phụ thuộc hàm trong tập cực tiểu và rút gọn để có thể nhận được tập dạng vành cực tiểu và rút gọn được.

Thí dụ 2. Cho tập phụ thuộc hàm: $F = \{B_1B_2 \to A,\ D_1D_2 \to B_1B_2,\ B_1 \to C_1,\ B_2 \to C_2,\ D_1 \to A,\ D_2 \to A,\ AB_1C_2 \to D_2,\ AB_2C_1 \to D_1\}$. Tập F là cực tiểu và rút gọn. Các tập trái tương đương là B_1B_2 và D_1D_2. Nhóm các phụ thuộc hàm thành các phụ thuộc hàm phức hợp, nhận được $G = \{(B_1B_2, D_1D_2) \to A,\ (B_1) \to C_1,\ (B_2) \to C_2,\ (D_1) \to A,\ (D_2) \to A,\ (AB_1C_2) \to D_2,\ (AB_2C_1) \to D_1\}$. Ta thấy phụ thuộc hàm phức hợp đầu tiên có thuộc tính A trong vế phải là dư thừa, tức là G không phải là tập dạng vành cực tiểu và rút gọn.

Tác giả Maier D. [2] nêu thí dụ này và khẳng định đây là vấn đề phức tạp nhất của Thuật toán SYNTHESIZE.

Định nghĩa 17. Phụ thuộc hàm phức hợp trong phủ dạng vành G được gọi là *dư thừa* nếu có thể loại bỏ khỏi G mà không vi phạm sự tương đương. Phụ thuộc hàm phức hợp trong G được gọi là *rút gọn trái* (*rút gọn phải*), nếu các tập trái không có thuộc tính có thể dịch chuyển (tương ứng, nếu các tập phải không có thuộc tính dư thừa). Cho G là tập dạng vành không có phụ thuộc hàm phức hợp dư thừa, tập G được gọi là *rút gọn trái* (*rút gọn phải*), nếu mọi phụ thuộc hàm phức hợp trong G là rút gọn trái (tương ứng, là rút gọn phải).

Trong [2] không có khái niệm phụ thuộc hàm phức hợp dư thừa, nếu sử dụng khái niệm trong Định nghĩa 17 thì Định nghĩa 13 trong [2] được phát biểu thành: "Phủ dạng vành G là *không dư thừa*, nếu G không có phụ thuộc hàm phức hợp dư thừa, ngoài ra, không có một phụ thuộc hàm phức hợp nào trong G chứa các tập trái có thể chuyển dịch. Trong trường hợp ngược lại thì G là dư thừa".

Các khái niệm trong Định nghĩa 17 đều định nghĩa trên cơ sở G là tập dạng vành không có phụ thuộc hàm phức hợp dư thừa, nên hoàn toàn phân biệt với "phủ dạng vành đầy đủ đối với phủ rút gọn trái (phải) của tập phụ thuộc hàm" - hai loại phủ này có thể là dư thừa. Khái niệm phủ dạng vành không dư thừa (Định nghĩa 13 trong [2]) có thể có tập trái chứa thuộc tính có thể dịch chuyển nên xác định lớp dạng vành rộng hơn so với khái niệm phủ dạng vành rút gọn trái (Định nghĩa 17), việc xây dựng khái niệm này phầm phân biệt với khái niệm phủ dạng vành rút gọn phải và khái niệm phụ thuộc hàm phức hợp thu hẹp phải (Định nghĩa 19). Khái niệm phủ dạng vành rút gọn (Định nghĩa 14 trong [2]) đồng nhất với khái niệm phủ dạng vành vừa là rút gọn trái vừa là rút gọn phải (Định nghĩa 17).

Hoàn toàn không đơn giản khi cần tìm phủ dạng vành đầy đủ cực tiểu, rút gọn phải đối với tập phụ thuộc hàm cho trước.

Thí dụ 3. Cho tập phụ thuộc hàm $F = \{A \to B,\ B \to ACD,\ AE \to IJ\}$ là cực tiểu và rút gọn phải. Tập $G = \{(A, B) \to ABCD,\ (AE) \to IJ\}$ là phủ dạng vành đầy đủ, cực tiểu đối với tập F, nhưng dư thừa vế phải. Tập $G' = \{(A, B) \to CD,\ (AE) \to IJ\}$ phủ dạng vành đầy đủ, cực tiểu, rút gọn phải đối với tập F.

Định lý 2. *Cho G là tập dạng vành cực tiểu. Sự hợp nhất các tập đặc trưng tự nhiên của tất cả các phụ thuộc hàm phức hợp trong G tạo thành tập phụ thuộc hàm cực tiểu tương đương với G.*

Chứng minh. Ký hiệu tập phụ thuộc hàm F là hợp nhất các tập đặc trưng tự nhiên của tất cả các phụ thuộc hàm phức hợp trong tập dạng vành cực tiểu G, theo Bổ đề 1 thì F là không dư và tương đương với G, theo Định nghĩa 10 thì số lượng phụ thuộc hàm của F bằng số lượng tập trái của G. Nếu F không là cực tiểu, thì ký hiệu F' là tập cực tiểu tương đương với F. Tạo phủ dạng vành G' tương đương và đầy đủ đối với F'. Theo điều kiện đủ của Định lý 1 thì G' là phủ dạng vành cực tiểu, có số lượng tập trái bằng số vế trái của F' là ít hơn F, tức là ít hơn số lượng tập trái của G, nghĩa tà tập G không là cực tiểu. Đây là điều mâu thuẫn. □

Cho G là phủ dạng vành đầy đủ đối với tập phụ thuộc hàm F, thì có thể nói: F là tập phụ thuộc hàm đặc trưng tự nhiên của G. Nên Định lý 2 là sự mở rộng kết quả điều kiện cần của Định lý 1 cho khái niệm phủ dạng vành nói chung.

Có nhiều cách để thể hiện tập phụ thuộc hàm đặc trưng tự nhiên đối với phụ thuộc hàm phức hợp cho trước, sau đây là định nghĩa một cách thể hiện đặc biệt.

Định nghĩa 18. Tập phụ thuộc hàm F được gọi là *tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ* (completely natural characteristic set) đối với phụ thuộc hàm phức hợp $(X_1, X_2, \dots, X_k) \rightarrow Y$, nếu F là tập phụ thuộc hàm phụ thuộc hàm đặc trưng tự nhiên đối với phụ thuộc hàm phức hợp đã cho và F có dạng:

$$F = \{X_i \rightarrow (\textstyle\bigcup_{j=1;j\neq i}^{k} X_j)Y \mid i = 1, 2, \dots, k\}.$$

Với khái niệm này, một tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ đối với một tập dạng vành sẽ thể hiện được sự tương đương của các vế trái trong tập phụ thuộc hàm này một cách trực tiếp (do có sự tương đương của các tập trái thuộc cùng một phụ thuộc hàm phức hợp thuộc tập dạng vành đã cho) mà không phải dựa vào sự suy dẫn (hay vào bao đóng của tập phụ thuộc hàm đó) mới thấy được.

Thí dụ 4. Xét tập dạng vành trong Thí dụ 2:

$$G = \{(B_1B_2, D_1D_2) \rightarrow A,\ (B_1) \rightarrow C_1,\ (B_2) \rightarrow C_2,\ (D_1) \rightarrow A,$$
$$(D_2) \rightarrow A,\ (AB_1C_2) \rightarrow D_2,\ (AB_2C_1) \rightarrow D_1\}$$

- Tập F_1 là tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ đối với G:

$$F_1 = \{B_1B_2 \rightarrow D_1D_2A,\ D_1D_2 \rightarrow B_1B_2A,\ B_1 \rightarrow C_1,\ B_2 \rightarrow C_2,$$
$$D_1 \rightarrow A,\ D_2 \rightarrow A,\ AB_1C_2 \rightarrow D_2,\ AB_2C_1 \rightarrow D_1\}$$

- Tập F_2 là tập phụ thuộc hàm đặc trưng tự nhiên đối với G:

$$F_2 = \{B_1B_2 \rightarrow A,\ D_1D_2 \rightarrow B_1B_2,\ B_1 \rightarrow C_1,\ B_2 \rightarrow C_2,$$
$$D_1 \rightarrow A,\ D_2 \rightarrow A,\ AB_1C_2 \rightarrow D_2,\ AB_2C_1 \rightarrow D_1\}$$

Dễ dàng thấy là trong F_2 thì sự tương đương của các vế trái $B_1B_2 \leftrightarrow D_1D_2$ không thể hiện trực tiếp như trong F_1.

Thuật toán CONACHASET

VÀO: Phụ thuộc hàm phức hợp $CF = (X_1, X_2, \dots, X_k) \rightarrow Y$.

RA: Tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ F.

CONACHASET(CF)

begin

$F := \{X_i \rightarrow (\bigcup_{j=1;j\neq i}^{k} X_j)Y \mid i = 1, 2, \dots, k\}$;

return(F);

end.

Hai kết quả (các bổ đề 6 và 7) sau đây là khẳng định trực tiếp được từ Định nghĩa 18.

Bổ đề 6. *Thuật toán CONACHASET xác định đúng tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ đối với phụ thuộc hàm phức hợp cho trước.*

Bổ đề 7. *Thuật toán CONACHASET có độ phức tạp tính toán theo thời gian là $O(k)$, trong đó k là số lượng tập trái của phụ thuộc hàm phức hợp cho trước.*

Định nghĩa 19. Phụ thuộc hàm phức hợp có dạng $CF = (X_1, X_2, \ldots, X_k) \to Y - (\bigcup_{j=1}^{k} X_j)$ được gọi là *phụ thuộc hàm phức hợp thu hẹp phải* (right restricted compound functional dependency).

Định nghĩa 20. Cho F là tập phụ thuộc hàm. Cho G là phủ dạng vành đầy đủ đối với F và nếu G gồm các phụ thuộc hàm phức hợp thu hẹp phải thì G là *phủ dạng vành đầy đủ và thu hẹp phải* (right restricted complexity annular cover) đối với F.

Thí dụ 5. Tập $G'' = \{(A, B) \to CD, (AE) \to IJ\}$ là phủ dạng vành đầy đủ và thu hẹp phải đối với tập F trong Thí dụ 1.

So sánh Định nghĩa 17 và Định nghĩa 20 dễ dàng nhận thấy: phủ dạng vành rút gọn phải đối với tập phụ thuộc hàm F không đồng thời là phủ dạng vành đầy đủ thu hẹp phải đối với F, và ngược lại. Một phản thí dụ minh họa là phủ dạng vành G trong Thí dụ 2 là phủ dạng vành đầy đủ và thu hẹp phải đối với tập F nhưng không là phủ dạng vành rút gọn phải đối với tập F. Trong những trường hợp đặc biệt thì phủ dạng vành đầy đủ thu hẹp phải đồng thời cũng là phủ dạng vành rút gọn phải.

Thuật toán tạo phủ dạng vành đầy đủ và thu hẹp phải đối với tập phụ hàm cho trước cũng như Thuật toán COANCOVER là thực hiện so sánh các bao đóng của các vế trái (của các phụ thuộc hàm) có độ phức tạp tính toán theo thời gian căn cứ theo thuật toán tính bao đóng của p tập thuộc tính vế trái nên là $O(np)$ (việc tính bao đóng sử dụng Thuật toán LINCLOSURE).

Thuật toán RRCOANCOVER

VÀO: Tập phụ thuộc hàm $F = \{X_i \to Y_i \mid i = 1, 2, \ldots, p\}$.

RA: Tập G là phủ dạng vành đầy đủ và thu hẹp phải đối với F.

RRCOANCOPVER

begin

  for mỗi phụ thuộc hàm $X_i \to Y_i \in F$ **do**

    $E_F(X_i) := \{X_j \to Y_j \mid X_i \leftrightarrow X_j, \forall X_j \to Y_j \in F\}$;

  $\overline{E}_F := \{E_F(X_i) \mid i = 1, 2, \ldots, p\}$;

   //ký hiệu RRCF^t là phụ thuộc hàm phức hợp thu hẹp phải thứ t.

   //RRCF^t có dạng: - Vế trái gồm các tập trái là các X_j thuộc $E_F(X_i)$.

   //   - Vế phải là hợp của các Y_j trừ hợp của các X_j (thuộc $E_F(X_i)$).

  $G := \{\text{RRCF}^t \mid t = 1, 2, \ldots, |\overline{E}_F|\}$;

  return(G);

end.

Dễ dàng khẳng định được hai kết quả (các bổ đề 8 và 9) sau đây.

Bổ đề 8. *Thuật toán RRCOANCOVER xác định đúng phủ dạng vành đầy đủ đối với tập phụ thuộc hàm cho trước.*

Bổ đề 9. *Thuật toán RRCOANCOVER có độ phức tạp tính toán theo thời gian là $O(np)$.*

Định lý 3. *Không làm mất tính tổng quát, giả sử có V_1 là tập dạng vành cực tiểu và rút gọn trái. Cho F_1 là hợp nhất các tập đặc trưng tự nhiên đầy đủ của tất cả các phụ thuộc hàm phức hợp trong V_1. Cho F_2 là phủ rút gọn phải của F_1. Cho V_2 là phủ dạng vành đầy đủ và thu hẹp phải đối với F_2, thì V_2 là phủ dạng vành đầy đủ, cực tiểu và rút gọn đối với F_2.*

Chứng minh. Vì V_1 là tập dạng vành cực tiểu nên theo Định lý 2 thì F_1 là phụ thuộc hàm cực tiểu tương đương với V_1. Vì V_1 là rút gọn trái và F_1 là phụ thuộc hàm đặc trưng tự nhiên đầy đủ đối V_1 nên F_1 là đầy đủ, cực tiểu, rút gọn trái đối với V_1. Tiếp tục vì F_2 là tập phụ thuộc hàm cực tiểu nên theo Định lý 1 thì V_2 là phủ dạng vành cực tiểu đối với F_2, lại vì F_2 là rút gọn trái nên V_2 là phủ cực tiểu và rút gọn trái đối F_2.

Ký hiệu phụ thuộc hàm phức hợp thứ i trong tập dạng vành cực tiểu V_1 là:

$$CF_1^i = (X_1^i, X_2^i, \ldots, X_t^i) \to Y_1^i \in V_1.$$

Ký hiệu phụ thuộc hàm thứ i đặc trưng tự nhiên đầy đủ đối với CF_1^i là:

$$FD_1^i = \{X_h^i \to (\bigcup_{j=1;j\neq h}^t X_j^i)Y_1^i \mid h = 1, 2, \ldots, t\} \subseteq F_1.$$

Giả sử tồn tại thuộc tính A là dư thừa trong một phụ thuộc hàm nào đó thuộc FD_1^i. Vì F_1 là rút gọn trái, nên thuộc tính A chỉ có thể là thuộc tính dư thừa thuộc vế phải trong một phụ thuộc hàm nào đó thuộc FD_1^i. Ký hiệu Z_h^i tương ứng là tập thuộc tính dư thừa vế phải của phụ thuộc hàm chỉ số h thuộc FD_1^i. Tập F_2 là phủ rút gọn phải sủa F_1 nên tương ứng với ký hiệu tập FD_1^i ta có ký hiệu tập $FD_2^i \subseteq F_2$:

$$FD_2^i = \{X_h^i \to (\bigcup_{j=1;j\neq h}^t X_j^i Y_1^i) - Z_h^i \mid h = 1, 2, \ldots, t\} \subseteq F_2.$$

Tức là tập FD_2^i là tập phụ thuộc hàm rút gọn.

Tương ứng với phụ thuộc hàm phức hợp rút gọn trái $CF_1^i \in V_1$ ta có phụ thuộc hàm phức hợp đầy đủ và thu hẹp phải đối với tập FD_2^i là: $RRCF_2^i = (X_1^i, X_2^i, \ldots, X_t^i) \to \bigcup_{h=1}^t((\bigcup_{j=1;j\neq h}^t X_j^i Y_1^i) - Z_h^i) - (\bigcup_{j=1}^t X_j^i) \in V_2$ (bởi vì có $X_1^i \leftrightarrow X_2^i \leftrightarrow \cdots \leftrightarrow X_t^i$).

Giả sử V_2 không là rút gọn và có thuộc tính dư thừa B ở phụ thuộc hàm phức hợp $RRCF_2^i$ với chỉ số i nào đó, vì V_2 là rút gọn trái nên B chỉ có thể là thuộc tính dư thừa ở vế phải: $B \in \bigcup_{h=1}^t((\bigcup_{j=1;j\neq h}^t X_j^i Y_1^i) - Z_h^i) - (\bigcup_{j=1}^t X_j^i)$, rõ ràng là $B \notin (\bigcup_{j=1}^t X_j^i)$ và chỉ có thể là $B \in (\bigcup_{j=1;j\neq h}^t X_j^i Y_1^i) - Z_h^i$ với chỉ số h nào đó, tức B là thuộc tính dư thừa vế phải của phụ thuộc hàm chỉ số h thuộc tập $FD_2^i \subseteq F_2$, là mâu thuẫn vì F_2 là phủ rút gọn của tập F_1, mà tập F_1 là đặc trưng tự nhiên đầy đủ, rút gọn đối với V_1. □

Như vậy, đến đây đã có giải pháp để nhận được phủ dạng vành, rút gọn cho trường hợp Thí dụ 2.

Thí dụ 6. Xét tập dạng vành trong Thí dụ 2:

$$V_1 = G = \{(B_1B_2, D_1D_2) \to A,\ (B_1) \to C_1,\ (B_2) \to C_2,\ (D_1) \to A,$$
$$(D_2) \to A,\ (AB_1C_2) \to D_2,\ (AB_2C_1) \to D_1\}.$$

Tập V_1 là cực tiểu và rút gọn trái, có thuộc tính A là dư thừa ở vế phải. Tạo phủ F_1 là tập phụ thuộc hàm đặc trưng tự nhiên đầy đủ đối với V_1:

$$F_1 = \{B_1B_2 \to D_1D_2A,\ D_1D_2 \to B_1B_2A,\ B_1 \to C_1,\ B_2 \to C_2,$$
$$D_1 \to A,\ D_2 \to A,\ AB_1C_2 \to D_2,\ AB_2C_1 \to D_1\}.$$

F_1 là cực tiểu và rút gọn trái, có thuộc tính A là dư thừa ở vế phải. Rút gọn vế phải của F_1 được tập F_2:

$$F_2 = \{B_1B_2 \to D_1D_2,\ D_1D_2 \to B_1B_2,\ B_1 \to C_1,\ B_2 \to C_2,$$
$$D_1 \to A,\ D_2 \to A,\ AB_1C_2 \to D_2,\ AB_2C_1 \to D_1\}.$$

Tập phụ thuộc hàm F_2 là cực tiểu và rút gọn. Tiếp tục ta có phủ dạng vành V_2 đầy đủ thu hẹp phải đối với F_2 là:

$$V_2 = \{(B_1B_2, D_1D_2),\ (B_1) \to C_1,\ (B_2) \to C_2,\ (D_1) \to A,$$
$$(D_2) \to A,\ (AB_1C_2) \to D_2,\ (AB_2C_1) \to D_1\}.$$

Phủ dạng vành V_2 là cực tiểu và rút gọn.

Nhận xét: 1) Nếu F_1 là phụ thuộc hàm đặc trưng tự nhiên (không phải là phụ thuộc hàm đặc trưng tự nhiên đầy đủ) đối với V_1 thì có thể F_1 là:

$$F_1 = \{B_1B_2 \to A,\ D_1D_2 \to B_1B_2,\ B_1 \to C_1,\ B_2 \to C_2,$$
$$D_1 \to A,\ D_2 \to A,\ AB_1C_2 \to D_2,\ AB_2C_1 \to D_1\}.$$

F_1 là cực tiểu và rút gọn, nên tập F_2 chính là F_1. Tiếp tục nếu:

1.1) V_2 là phủ dạng vành đầy đủ, thu hẹp phải đối với F_2 thì có thể là:

$$V_2 = \{(B_1B_2, D_1D_2) \to A,\ (B_1) \to C_1,\ (B_2) \to C_2,$$
$$(D_1) \to A,\ (D_2) \to A,\ (AB_1C_2) \to D_2,\ (AB_2C_1) \to D_1\}.$$

Ta thấy có thuộc tính A dư thừa vế phải trong phụ thuộc hàm phức hợp đầu tiên (giống như trong Thí dụ 2).

1.2) V_2 là phủ dạng vành (mà không là phủ dạng vành đầy đủ, thu hẹp phải) đối với F_2 thì có thể là:

$$V_2 = \{(B_1B_2, D_1D_2) \to AB_1B_2D_1D_2,\ (B_1) \to B_1C_1,\ (B_2) \to C_2,$$
$$(D_1) \to A,\ (D_2) \to A,\ (AB_1C_2) \to D_2,\ (AB_2C_1) \to D_1\}.$$

Ta thấy có những thuộc tính dư thừa vế phải trong phụ thuộc hàm phức hợp thứ nhất và thứ hai.

2) Nếu F_1 và F_2 là như trong Thí dụ 6, và V_2 là phủ dạng vành (không là phủ dạng vành đầy đủ thu hẹp phải) đối với F_2 thì có thể V_2 lại là như trường hợp 1.2).

Như vậy, qua Thí dụ 6, ta thấy rõ hơn ý nghĩa của những khái niệm được nêu trong phần này.

Căn cứ vào Định lý 1 và Định lý 3 khẳng định được sự đúng đắn của thuật toán sau đây.

Thuật toán REDMINCOANCOVER

VÀO: Tập phụ thuộc hàm $F = \{X_i \to Y_i \,|\, i = 1, 2, \dots, p\}$.

RA: Tập G là phủ dạng vành đầy đủ, cực tiểu, rút gọn đối với F.

REDMINCOANCOVER(F)

```
begin
    V1 := MINCOANCOVER(LEFTRED(F));
    F1 := {CONACHASET(CF^i) | CF^i ∈ V1; ∀i};
          //CF^i là ký hiệu phụ thuộc hàm phức hợp thứ i thuộc V1.
    F2 := RIGHTRED(F1);
    G := RRCOANCOVER(F2);
    return(G);
end.
```

Lưu ý: Các thuật toán trong [2]: LEFTRED - rút gọn vế trái của tập phụ thuộc hàm, RIGHTRED - rút gọn vế phải của tập phụ thuộc hàm đều có độ phức tạp tính toán theo thời gian là $O(n^2)$.

Bổ đề 10. *Thuật toán REDMINCOANCOVER xác định đúng phủ dạng vành đầy đủ, cực tiểu, rút gọn đối với tập phụ thuộc hàm cho trước.*

Bổ đề 11. *Thuật toán REDMINCOANCOVER có độ phức tạp tính toán theo thời gian là $O(n^2)$.*

Chứng minh. Độ phức tạp tính toán theo thời gian của Thuật toán REDMIMCOANCOVER là tổng độ phức tạp tính toán theo thời gian cua các thuật toán: Thuật toán LEFTRED (là $O(n^2)$), Thuật toán MINCOANCOVER (là $O(np)$), Thuật toán CONACHASET (là $O(p)$), Thuật toán RIGHTRED (là $O(n^2)$) và Thuật toán RRCOANCOVER (là $O(np)$), tức là: $O(n^2)+O(np)+O(p)+O(n^2)+O(np)$, nên là $O(n^2)$. □

Việc sửa đổi hợp lý Thuật toán REDMINCOANCOVER sẽ nhận được thuật toán tìm phủ dạng vành đầy đủ, cực tiểu, rút gọn phải đối với tập phụ thuộc hàm cho trước.

Thuật toán RMINCOANCOVER

VÀO: Tập phụ thuộc hàm $F = \{X_i \to Y_i \,|\, i = 1, 2, \dots, p\}$.

RA: Tập G là phủ dạng vành đầy đủ, cực tiểu, rút gọn phải đối với F.

RMINCOANCOVER(F)

begin

V_1 := MINCOANCOVER(F);

F_1 := {CONACHASET(CF^i) | $CF^i \in V_1$, $\forall i$};

//CF^i là ký hiệu phụ thuộc hàm phức hợp thứ i thuộc V_1.

F_2 := RIGHTRED(F_1);

G := RRCOANCOVER(F_2);

return(G);

end.

Dễ dàng nhận thấy việc tìm phủ dạng vành đầy đủ, cực tiểu, rút gọn phải G_1 là phức tạp hơn nhiều so với việc tìm phủ dạng vành đầy đủ, cực tiểu, rút gọn trái G_2 đối với tập phụ thuộc hàm F cho trước, vì trường hợp rút gọn trái chỉ cần thực hiện G_2 := MINCOANCOVER(LEFTRED)(F)).

Chứng minh tương tự như Định lý 3, với giả thiết xuất phát từ tập dạng vành đầy đủ, cực tiểu sẽ khẳng định được tính đúng đắn của Thuật toán RMINCOANCOVER.

Bổ đề 12. *Thuật toán RMINCOANCOVER xác định đúng phủ dạng vành đầy đủ, cực tiểu, rút gọn phải đối với tập phụ thuộc hàm cho trước.*

Bổ đề 13. *Thuật toán RMINCOANCOVER có độ phức tạp tính toán theo thời gian là* $O(n^2)$.

Chứng minh. Độ phức tạp tính toán theo thời gian của của Thuật toán RMINCOANCOVER là tổng độ phức tạp tính toán theo thời gian của bốn thuật toán: Thuật toán MINCOANCOVER (là $O(np)$), Thuật toán CONACHASET (là $O(p)$), Thuật toán RIGHTRED (là $O(n^2)$) và Thuật toán RRCOANCOVER (là $O(np)$), tức là: $O(np) + O(p) + O(n^2) + O(np)$, nên là $O(n^2)$. □

TÀI LIỆU THAM KHẢO

[1] Atzeni P., De Antonellis V., *Relational Database Theory*, The Benjamin/Cummings Publishing Company, 1983.

[2] Maier D., *The Theory of Relational Databases*, Computer Science Press, 1983.

[3] Phạm Quang Trung, Thuật toán tổng hợp THV và so sánh với Thuật toán SYNTHESIZE, *Tạp chí Bưu chính Viễn thông: Các công trình nghiên cứu và triển khai Công nghệ thông tin và Viễn thông*, Tổng cục Bưu điện, Hà nội, số 5, tháng 3 (2001).

[4] Ullman J. D., *Principles of Database Systems*, 2$^{\text{nd}}$ edition, Computer Science Press, 1983.

Nhận bài ngày 23 tháng 10 năm 2000

Phòng Công nghệ thông tin
Viện Kiểm sát nhân dân tối cao.