*Tạp chí Tin học và Điều khiển học, T. 16, S. 3 (2000), 7–15*

# TÍNH ĐÚNG ĐẮN CỦA LƯỢC ĐỒ CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG

ĐOÀN VĂN BAN

**Abstract.** Inheritance, in object-oriented systems, allows compact, open and readable software to be constructed [2], [4], [5], [6]. Powerful object models adopt multiple inheritance, allowing a type (or class) to inherit from more than one supertype. Unfortunately, this powerful modeling mechanism can generate inheritance conflicts, which arise when the same property (or operation) is defined in more than one supertype, or a property (or operation) already present in one supertype, is locally redefined in subtype (overriding) [1]. In this article, a graph-theoretic approach for object-oriented database schema correctness checking has been presented. Schema correctness, which depends on the inheritance hierarchy, is twofold: it requires the absence of unsolvable inheritance conflicts and the termination of the inhertance process.

## 1. TÓM TẮT

Trong cách tiếp cận hướng đối tượng, cơ chế thừa kế cho phép chúng ta xây dựng được những phần mềm cô đọng, rõ ràng và có tính mở hơn [2, 4, 5, 6]. Những mô hình mạnh đều chấp nhận cơ chế thừa kế bội, cho phép một kiểu (hay lớp) thừa kế từ nhiều hơn một kiểu cơ sở. Nhưng cũng chính từ cơ chế mạnh này lại có thể tạo ra những xung đột giữa các kiểu thừa kế, khi một tính chất (thuộc tính) được thừa kế từ một số (nhiều hơn một) kiểu cha mà những kiểu này lại không tương thích với nhau [1]. Tiếp theo những kết quả đã nêu ở [3], và dựa vào các tính chất cơ bản của quá trình thừa kế, chúng tôi giới thiệu một cách tiếp cận theo lý thuyết đồ thị để kiểm tra tính đúng đắn của lược đồ cơ sở dữ liệu. Tính đúng đắn của lược đồ phụ thuộc vào tính phi mâu thuẫn và sự kết thúc của quá trình thừa kế kiểu trong hệ thống.

## 2. GIỚI THIỆU

Trong bài báo này chúng ta sử dụng một số ký hiệu, khái niệm và nhiều tính chất thừa kế kiểu trong hệ thống đối tượng đã được giới thiệu trong [3]. Để tiện theo dõi chúng ta hãy nhắc lại một số khái niệm chính.

Trong hệ thống kiểu người ta thường sử dụng một số kiểu cơ bản mà chúng ta gọi là những kiểu nguyên thủy như kiểu integer, real, string, v.v.. Ký hiệu $L$ là tập các kiểu nguyên thủy và $T$ là tập tên các kiểu được tạo lập trong hệ thống.

**Định nghĩa 1.** *Quan hệ Sub thừa kế kiểu*
Cho trước tập các kiểu $T$, $\tau_i \in T$, $i = 1, 2, \ldots, n$ và $\delta_m \in T \cup L$, $m = 1, 2, \ldots, k$. Kiểu $\tau$ thừa kế (trực tiếp) từ những kiểu $\tau_i$ và được bổ sung một số thuộc tính trong các kiểu $\delta_m$ sẽ được định nghĩa như sau:

$$type \quad \tau = \tau_1, \tau_2, \ldots, \tau_n \,\{p_1 : \delta_1; p_2 : \delta_2; \ldots; p_k : \delta_k\}. \tag{1}$$

và được ký hiệu $\tau\,\mathrm{Sub}\,\tau_i$ $(i = 1, 2, \ldots, n)$, kiểu $\tau_i$ gọi là cha của $\tau$ hay $\tau$ là con của $\tau_i$.
Kiểu chuẩn có thể là kiểu nguyên thủy hoặc kiểu được tạo lập ra như sau:

$$type \quad \tau = \{p_1 : \delta_1; p_2 : \delta_2; \ldots; p_k : \delta_k\}. \tag{2}$$

Trong các biểu thức (1) và (2): $\delta_m \in T \cup L$, $p_m \in P$, $m = 1, 2, \ldots, k$, $P$ là tập tất cả các thuộc tính. Gọi $I$ là tập tất cả các kiểu được tạo ra theo quan hệ thừa kế. Gọi $N$ là tập tất cả các kiểu chuẩn. Với mỗi kiểu $\tau \in T$ chúng ta có thể xác định tập tất cả các kiểu cơ sở để được thừa kế thông qua hàm $\mathcal{A} : T \to \cup L$, được xác định như sau:

$$\mathcal{A}(\tau) = \emptyset \text{ nếu } \tau \in N,$$
$$\mathcal{A}(\tau) = \bigcup_{i=1}^{n} \{\{\tau_j\} \cup \mathcal{A}(\tau_j)\} \text{ cho những } \tau \in I. \tag{3}$$

**Định nghĩa 2.** *Lược đồ đối tượng*
Lược đồ cơ sở dữ liệu hướng đối tượng (gọi tắt là lược đồ đối tượng) là một bộ ba $\Sigma = \langle T, L, P\rangle$ thỏa mãn ba tính chất sau:

(i) Tất cả những kiểu $\tau \in T$ phải được định nghĩa duy nhất, nghĩa là xuất hiện đúng một lần ở vế trái trong các định nghĩa kiểu của $T$.

(ii) Với mọi kiểu $\tau \in T$, $\tau \notin \mathcal{A}(\tau)$, nghĩa là quan hệ thừa kế không tạo thành chu trình và những thuộc tính bổ sung không lặp lại các thuộc tính của ở các kiểu cơ sở.

(iii) Không có các bộ lồng nhau trong định nghĩa kiểu, trong thừa kế và khai báo thuộc tính đều phải thông qua tên gọi kiểu.

**Định nghĩa 3.** Cho trước lược đồ $\Sigma = \langle T,\ L,\ P\rangle$, trên $T$ xác định một quan hệ $\leq$ như sau:

(i) $\forall \in T \cup L$, $\tau \leq \tau$.

(ii) Nếu $\tau_1 = \{a_i : \eta_i;\ c_k : \gamma_k\}$, $\tau 2 = \{a_i : \mu_i\} \in T$ thì
$$\tau_1 \leq \tau_2 \Leftrightarrow \eta_i \leq \mu_i \text{ với } \eta_i,\ \mu_i,\ \gamma_k \in T \cup L. \tag{4}$$

Nếu $\tau_1 \leq \tau_2$ thì chúng ta nói rằng $\tau_1$ là kiểu con của $\tau_2$.

**Định nghĩa 4.** Quan hệ suy diễn $\leq^{(n)}$ trong $\Sigma$ được định nghĩa như sau:

(i) $\tau \leq^{(n)} \tau$, với mọi $\tau \in T \cup L$ và $n \geq 0$.

(ii) Nếu $\tau_1 = \{a_i : \eta_i;\ c_k : \gamma_k\}$ và $\tau_2 = \{a_i : \mu_i$ thì với mọi $n \geq 0$,
$$\tau_1 \leq^{(n)} \tau_2 \Leftrightarrow \forall i,\ \exists n_i,\ 0 \leq n_i < n \text{ sao cho } \eta_i \leq^{(n_i)} \mu_i. \tag{5}$$

Trong [3] đã khẳng định quan hệ $\leq$ là thứ tự bộ phận tương đương của $\leq$ với $\leq^{(n)}$. Để khẳng định được tính phi mâu thuẫn của hệ thống đối tượng chúng ta phải kiểm tra được tính nhất quán dữ liệu của lược đồ đối tượng. Một lược đồ được gọi là nhất quán nếu quá trình phân giải thừa kế để xác định các thành phần tường minh của các kiểu trong hệ thống không dẫn tới xung đột trong số những kiểu cơ sở. Quá trình phân giải thừa kế là quá trình biến đổi những kiểu không ở dạng chuẩn về dạng chuẩn tương đương, hay còn gọi là quá trình chuẩn hóa kiểu. Trong mô hình hệ thống phần mềm, cơ chế thừa kế được sử dụng để hỗ trợ cho việc sử dụng lại những kiểu đã được xây dựng tốt và tránh khỏi phải viết lại các thành phần (thuộc tính) từ những kiểu cơ sở trong các kiểu thừa kế. Mặt khác chúng ta cũng đã biết, trong hệ thống không cho phép lặp lại vì sự lặp lại không chỉ dẫn tới dư thừa mà còn gây ra sự không nhất quán dữ liệu. Như vậy, nếu trong quá trình chuẩn hóa kiểu mà một kiểu được thừa kế bội (nhiều hơn 1) kiểu cha và có cùng chung một số thuộc tính thì có thể dẫn tới xung đột hay không nhất quán về kiểu. Khi những thuộc tính chung đó cùng qui chiếu tới một kiểu nguyên thủy thi không xảy ra xung đột. Quá trình thay thế tên các kiểu cha (cơ sở) bằng những thuộc tính tương ứng của chúng trong kiểu thừa kế có thể thực hiện thông qua phép hội nhập kiểu $\downarrow$.

## 3. PHÉP HỘI NHẬP KIỂU VÀ QUÁ TRÌNH THỪA KẾ

**Định nghĩa 5.** Phép tụ tập kiểu $\downarrow$ xác định quá trình thừa kế, được định nghĩa như sau:

(i) $\tau \downarrow \tau = \tau$, $\tau \in T \cup L$.

(ii) Nếu $\tau_1 \neq \tau_2$ và $\tau_1$ hoặc $\tau_2$ là nguyên thủy thì $\tau_1 \downarrow \tau_2 = \perp$.

(iii) Nếu $\tau_1 = \{a_i : \eta_i;\ b_j : \mu_j\}$, $\tau_2 = \{a_i : \eta_i';\ c_k : \gamma_k\}$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$, $k = 1, 2, ..., l$, và $\forall j, k : b_j \neq c_k$ thì khi đó
$$\tau_1 \downarrow \tau_2 = \perp \Leftrightarrow \exists i : \eta_i \downarrow \eta_i' = \perp,$$

$$\tau_1 \downarrow \tau_2 = \{a_i : \eta_i \downarrow \eta_i'; \, b_j : \mu_j; \, c_k : \gamma_k\} \Leftrightarrow \forall i : \eta_i \downarrow \eta_i' \neq \perp . \tag{6}$$

Trong đó $\perp$ được dùng để ký hiệu cho sự xung đột giữa các kiểu. $\tau_1 \downarrow \tau_2 = \perp$ nghĩa là phép tụ tập $\downarrow$ của $\tau_1$ với $\tau_2$ là không xác định. Hay nói cách khác trong quan hệ thừa kế từ $\tau_1$ và $\tau_2$ sẽ sử dụng chung những thuộc tính liên quan tới $\tau_1$ và $\tau_2$ mà bản thân chúng lại không tương thích nên dẫn tới mâu thuẫn.

*Ví dụ 1.* Cho trước hệ thống kiểu

type Nguoi_lon = {Tuoi: integer; Con: Sinh_vien};

type Sinh_vien = {Ho_ten: string; Truong: string; Ban: Nguoi_lon};

type Giao_vien= {Ho_ten: string; Luong: integer; Ban: Giao_vien};

type Nhan_vien = Sinh_vien, Giao_vien{ };

Để xác định được các thuộc tính (một cách tường minh) của Nhan_vien chúng ta phải thay thế Sinh_vien và Giao_vien bằng những thuộc tính của chúng sao cho không xảy ra mâu thuẫn. Hai kiểu Sinh_vien, Giao_vien có 2 thuộc tính chung là Ho_ten cùng kiểu string còn Ban có kiểu Nguoi_lon ở Sinh_vien, kiểu Giao_vien ở Giao_vien. Vậy:

Nhan_vien = Sinh_vien $\downarrow$ Giao_vien =

{Ho_ten: string; Ban: Nguoi_lon $\downarrow$ Giao_vien; Truong: string; Luong: integer}.

Trong đó

Nguoi_lon $\downarrow$ Giao_vien = {Ho_ten: string; Tuoi: integer; Con: Sinh_vien; Ban: Giao_vien}.

Như vậy nếu chúng ta đặt Nguoi_lon $\downarrow$ Giao_vien là kiểu mới để đảm bảo không có các bộ lồng nhau, ví dụ Vien_chuc thì kiểu Nhan_vien trong hệ thống trên không ở dạng chuẩn chuyển được về dạng chuẩn như sau:

type Nhan_vien = {Ho_ten: string; Ban: Vien_chuc; Truong: string; Luong: integer};

type Vien_chuc = {Ho_ten: string; Tuoi: integer; Con: Sinh_vien; Ban: Giao_vien}.

Hai kiểu được gọi là không tương thích nếu kết quả của phép hội nhập kiểu $\downarrow$ của chúng là không xác định.

*Ví dụ 2.*

type Nguoi_lon = {Tuoi: integer; Con: Sinh_vien};

type Sinh_vien = {Ho_ten: string; Truong: string; Ban: Nguoi_lon};

type Giao_vien = {Ho_ten: Hovaten; Luong: integer; Ban: Giao_vien};

type Hovaten = {Ho: string; Ten: string};

type Nhan_vien = Sinh_vien, Giao_vien{Dia_chi: string};

Bởi vì string và Hovaten không tương thích, string là nguyên thủy, Hovaten không phải nguyên thủy nên theo Định nghĩa 5 thì string $\downarrow$ Hovaten = $\perp$. Do vậy Sinh_vien $\downarrow$ Giao_vien = $\perp$, nghĩa là kiểu Nhan_vien định nghĩa thừa kế từ Sinh_vien, Giao_vien sẽ dẫn tới mâu thuẫn. Nói cách khác kiểu Nhan_viên định nghĩa trong ví dụ 2 là không xác định, vì vậy hệ thống này là có mâu thuẫn.

**Định nghĩa 6.** Quá trình thừa kế của $\tau \in I$ là kết thúc nếu số lần thực hiện phép hội nhập kiểu $\downarrow$ kết thúc sau hữu hạn lần áp dụng điều kiện (6).

Trước khi xét điều kiện kiểm tra tính đúng của lược đồ chúng ta hãy nghiên cứu một số tính chất quan trọng của quá trình thừa kế thông qua phép hội nhập kiểu $\uparrow$ và quan hệ thứ tự kiểu con $\leq$.

Sử dụng phương pháp tương tự như trong [3], các tính chất của phép hội nhập kiểu $\downarrow$ có thể khẳng định thông qua phép hội nhập kiểu suy diễn $\downarrow^{(n)}$.

**Định nghĩa 7.** Phép hội nhập kiểu suy diễn $\downarrow^{(n)}$ xác định quá trình thừa kế suy diễn được định nghĩa như sau:

(i) $\forall n \geq 0 : \tau \downarrow^{(n)} \tau = \tau$,

(ii) Nếu $\tau_1 \neq \tau_2$ và $\tau_1$ hoặc $\tau_2$ là nguyên thủy thì $\forall n \geq 0 : \tau_1 \downarrow^{(n)} \tau_2 = \perp$,

(iii) Nếu $\tau_1 = \{a_1 : \eta_i;\, b_j : \mu_j\}$, $\tau_2 = \{a_i : \eta'_i;\, c_k : \gamma_k\}$, $i = 1, \dots, n$; $j = 1, \dots, m$; $k = 1, \dots,$ và $\forall j, k : b_j \neq c_k$ khi đó $\forall n > 0\ \tau_1 \downarrow^{(n)} \tau_2 = \Leftrightarrow \exists i\, \exists n_i < n : \eta_i \downarrow^{(n_i)} \eta'_i = \perp$

$$\tau_1 \downarrow \tau_2 = \{a_i : \eta_i \downarrow \eta'_i;\, b_j u : \mu_j;\, c_k : \gamma_k\} \Leftrightarrow \forall i\, \forall 0 \leq n_i < n : \eta_i \downarrow^{(n_i)} \eta'_i \neq \perp . \tag{7}$$

**Định lý 1.** *Phép $\downarrow$ và $\downarrow^{(n)}$ là tương đương*

$$\forall \tau_1, \tau_2 \in T,\ \tau_1 \downarrow \tau_2 \neq \perp \Leftrightarrow \exists n \geq 0 : \tau_1 \downarrow^{(n)} \tau_2 \neq \perp . \tag{8}$$

*Chứng minh. Điều kiện cần:* Qui nạp theo số lần $m$ áp sụng tính chất (6).
*Điều kiện đủ:* Qui nạp theo $n$.

Sau đây chúng ta xét tiếp một số tính chất của quá trình thừa kế và quan hệ $\leq$.

**Bổ đề 1.** *Tính giao hoán của phép hội nhập kiểu*

$$\forall \tau_1, \tau_2 \in T,\ \tau_1 \downarrow \tau_2 \neq \perp \Rightarrow \tau_2 \downarrow \tau_1 \neq \perp \ \text{ và } \tau_1 \downarrow \tau_2 = \tau_2 \downarrow \tau_1. \tag{9}$$

*Chứng minh.* Để có (9) chúng ta chỉ cần chứng minh

$$\tau_1 \downarrow \tau_2 \neq \perp \Rightarrow \tau_1 \downarrow^{(n)} \tau_2 \neq \perp \ \text{ và } \tau_1 \downarrow^{(n)} \tau_2 = \tau_2 \downarrow^{(n)} \tau_1,\ n \geq 0. \tag{10}$$

Tính chất (10) lại dễ dàng được chứng minh qui nạp theo $n$. □

**Bổ đề 2.**

$$\forall \tau_1, \tau_2 \in T,\ \tau_1 \leq \tau_2 \Leftrightarrow \tau_1 \downarrow \tau_2 = \tau_1. \tag{11}$$

*Chứng minh.* Từ Định lý 1 suy ra (11) tương đương với

$$\tau_1 \leq^{(n)} \tau_2 \Leftrightarrow \tau_1 \downarrow^{(n)} \tau_2 = \tau_1,\ \forall n \geq 0. \tag{12}$$

Khi $n = 0$ thì khẳng định (12) là hiển nhiên. Giả thiết khẳng định trên đúng với $k < n$, cần chứng minh tiếp nó đúng với $n$.

*Điều kiện cần:* $\tau_1 = \{a_i : \eta_i;\, b_k : \mu_i\}$, $\tau_2 = \{a_1 : \eta'_i\}$, $\tau_1 \leq^{(n)} \tau_2 \Rightarrow \eta_i \leq^{(n_i)} \eta'_i$ với $n_i < n$. Theo giả thiết qui nạp chúng ta có $\eta_i \downarrow^{(n_i)} \eta'_i = \eta_i$. Vậy $\tau_1 \downarrow^{(n)} \tau_2 = \{a_i : \eta_i;\, b_k : \mu_k\} = \tau_1$.

*Điều kiện đủ:* Nếu $\tau_1 = \{a_i : \eta_i;\, b_j : \mu_j\}$, $\tau_2 = \{a_i : \eta'_i;\, c_k : \gamma_k\}$ và $\exists n \geq 0 : \tau_1 \downarrow^{(n)} \tau_2 = \tau_1 = \{a_i : \eta_i;\, b_j : \mu_j\}$. Theo định nghĩa của $\downarrow^{(n)}$ thì $\tau_1 \downarrow^{(n)} \tau_2 = \{a_i : \eta_i \downarrow^{(n_i)} \eta'_i;\, b_j : \mu_j;\, c_k : \gamma_k\} = \{a_i : \eta_i;\, b_j : \mu_j\}$ với $n_i < n$. Từ đó ta có $\eta_i \downarrow^{(n_i)} \eta'_i = \eta_i \Rightarrow \eta_i \leq^{(n_i)} \eta'_i$, nghĩa là $\tau_1 \leq^{(n)} \tau_2$.

**Bổ đề 3.**

$$\forall \tau, \tau_1, \tau_2 \in T,\ \tau \leq \tau_1,\ \tau \leq \tau_2 \Rightarrow \tau_1 \downarrow \tau_2 \neq \perp \ \text{ và } \tau \leq \tau_1 \downarrow \tau_2. \tag{13}$$

*Chứng minh.* Trong [3] đã khẳng định sự tương đương của $\leq$ với $\leq^{(n)}$, vậy $\tau \leq \tau_1 \Rightarrow \exists n \geq 0 : \tau \leq^{(n)} \tau_1$ và $\tau \leq \tau_2 \Rightarrow \exists m \geq 0 : \tau \leq^{(m)} \tau_2$. Đặt $r = \max\{m, n\}$, chúng ta sẽ thấy ngay rằng khẳng định (13) tương đương với

$$\tau_1 \downarrow^{(r)} \tau_2 \neq \perp \ \text{ và } \tau \leq^{(r)} \tau_1 \downarrow^{(r)} \tau_2. \tag{14}$$

Khẳng định (14) có thể chứng minh qui nạp theo $r$ tương tự như trên. □

**Bổ đề 4.**

$$\forall \tau_1, \tau_2 \in T,\ \tau_1 \downarrow \tau_2 \neq \perp \Rightarrow \tau_1 \downarrow \tau_2 \leq \tau_1 \ \text{ và } \tau_1 \downarrow \tau_2 \leq \tau_2. \tag{15}$$

*Chứng minh.* Bởi vì $\downarrow$ có tính hoán vị nên chỉ cần chỉ ra rằng $\forall \tau_1, \tau_2 \in T,\ \tau_1 \downarrow \tau_1 \neq \perp \Rightarrow \tau_1 \downarrow \tau_2 \leq \tau_1$. Mặt khác $\tau_1 \downarrow \tau_2 \neq \perp \Rightarrow \forall n \geq 0 : \tau_1 \downarrow^{(n)} \tau_2$ là xác định, nên chỉ cần khẳng định tiếp $\tau_1 \downarrow^{(n)} \tau_2 \leq \tau_1$. Điều này có thể chứng minh qui nạp khá đơn giản theo $n$. □

Từ các bổ đề trên chúng ta có kết quả khá lý thú như sau:

**Định lý 2.** *$\forall \tau_1, \tau_2 \in T$, nếu $\tau_1 \downarrow \tau_2 \neq \perp$ (xác định) thì nó là kiểu con chung lớn nhất của $\tau_1$ và $\tau_2$ theo quan hệ kiểu con $\leq$.*

*Chứng minh.* Suy trực tiếp từ hai bổ đề 3 và 4. □

Ngoài ra chúng ta thấy phép $\downarrow$ còn có tính chất kết hợp.

**Bổ đề 5.**

$$\forall \tau_1, \tau_2, \tau_3 \in T, (\tau_1 \downarrow \tau_2) \downarrow \tau_3 \neq \perp \Rightarrow \tau_1 \downarrow (\tau_2 \downarrow \tau_3) \neq \perp \text{ và } (\tau_1 \downarrow \tau_2) \downarrow \tau_3 = \tau_1 \downarrow (\tau_2 \downarrow \tau_3). \quad (16)$$

*Chứng minh.* Từ Định lý 2 suy ra $(\tau_1 \downarrow \tau_2) \downarrow \tau_3$ là kiểu con chung của $\tau_1, \tau_2, \tau_3$. Vậy $(\tau_1 \downarrow \tau_2) \downarrow \tau_3 \leq \tau_1$ và $(\tau_1 \downarrow \tau_2) \downarrow \tau_3 \leq \tau_2 \downarrow \tau_3 \Rightarrow (\tau_1 \downarrow \tau_2) \downarrow \tau_3 \leq \tau_1 \downarrow (\tau_2 \downarrow \tau_3)$. Tương tự chúng ta có $\tau_1 \downarrow (\tau_2 \downarrow \tau_3) \leq (\tau_1 \downarrow \tau_2) \downarrow \tau_3$.
Phép $\downarrow$ có tính kết hợp và thứ tự thực hiện từ trái qua phải nên khi tính với nhiều kiểu chúng ta có thể bỏ đi các dấu ngoặc. $\tau_1 \downarrow \tau_2 \downarrow \ldots \downarrow \tau_n = (\ldots(\tau_1 \downarrow \tau_2) \downarrow \ldots \downarrow \tau_n)$.

**Định lý 3.**

$$\forall \tau_1, \tau_2, \ldots, \tau_n \in T, \tau_1 \downarrow \tau_2 \downarrow \ldots \downarrow \tau_n \neq \perp \Leftrightarrow \forall 1 \leq i < k \leq n : \tau_i \downarrow \tau_k \neq \perp\,. \quad (17)$$

*Chứng minh.* Chỉ cần chứng minh định lý trên với $n = 3$, sau đó chúng ta có thể tổng quát hóa cho $n$ bất kỳ.

*Điều kiện cần:* $\tau_1, \tau_2, \tau_3 \in T$, $\tau_1 \downarrow \tau_2 \downarrow \tau_3 \neq \perp \Rightarrow$ tồn tại kiểu con chung của $\tau_1, \tau_2, \tau_3$. Dựa tiếp vào các bổ đề 3, 4 chúng ta khẳng định được từng cặp kiểu $\tau_1, \tau_2, \tau_3$ là xác định.

*Điều kiện đủ:* $\tau_1 \downarrow \tau_2 \neq \perp \Rightarrow \exists n_1 : \tau_1 \downarrow^{(n_1)} \tau_2 \neq \perp$, $\tau_1 \downarrow \tau_3 \neq \perp \Rightarrow \exists n_2 : \tau_1 \downarrow^{(n_2} \tau_3 \neq \perp$, $\tau_2 \downarrow \tau_3 \neq \perp \Rightarrow \exists n_3 : \tau_1 \downarrow^{(n_3)} \tau_2 \neq \perp$. Đặt $r = \max\{n_1, n_2, n_3\}$ và qui nạp theo $r$ để chứng minh $\tau_1 \downarrow \tau_2 \downarrow \tau_3 \neq \perp$.
□

Định lý 3 khẳng định rằng sự xung đột của các kiểu chỉ có thể xảy ra khi có ít nhất một cặp kiểu cha của một kiểu được thừa kế bội là không tương thích, nghĩa là phép hội nhập của hai kiểu cha đó là không xác định.

Dựa vào những kết quả nêu trên chúng ta có thể thực hiện biến đổi những kiểu không dạng chuẩn $\tau \in I$ thành dạng chuẩn $\tau \in N$ như sau:

(i) Đối với thừa kế đơn: $\tau_1 = \{a_i :\in \eta_i\} \in N$, $\tau \in I$, $\tau = \tau_1\ \{b_j : \mu_j\}$, $a_i \neq b_j$, sau khi thế các bộ phận của $\tau_1$ vào chúng ta có $\tau = \{a_i :\in \eta_i;\ b_j : \mu_j\}$.

(ii) Đối với thừa kế bội: $\tau = \tau_1, \tau_2, \ldots, \tau_n\ \{b_j : \mu_j\} \in I$, $n > 1$.
Nếu quá trình tính $\tau_1 \downarrow \tau_2 \downarrow \ldots \downarrow \tau_n$ kết thúc và $\tau_1 \downarrow \tau_2 \downarrow \ldots \downarrow \tau_n = \{a_i :\in \eta_i\}$ (xác định) thì $\tau = \{a_i :\in \eta_i;\ b_j : \mu_j\}$. Ngược lại nếu $\tau_1 \downarrow \tau_2 \downarrow \ldots \downarrow \tau_n = \perp$ thì $\tau = \perp$.

Vấn đề nảy sinh ở đây là kiểm tra tính kết thúc của quá trình thừa kế. Chúng ta nhận thấy là nếu trong quan hệ thừa kế có xuất hiện đệ qui thì quá trình phân giải thừa kế có thể bị lặp lại vô hạn. Trong [1] các tác giả cũng đã xét một số tính chất của đệ qui các kiểu con và điều kiện để phát hiện sự kết thúc của quan hệ thừa kế khi có đệ qui.

*Ví dụ 3.* Xét lược đồ sau

type Nguoi_lon = {Ho_ten: string; Tuoi: integer; Ban: Cong_nhan};

type Cong_nhan = {Ho_ten: string; Co_quan: string; Ban: Nguoi_lon};

type Can_bo = {Ho_ten: string; Luong: integer; Ban: Can_bo};

type Nhan_vien = Cong_nhan, Can_bo {Dia_chi: string};

Để chuyên Nhan_vien về dạng chuẩn thì chúng ta tính Cong_nhan $\downarrow$ Can_bo. Vấn đề chính là có thể xác định được kiểu cho thuộc tính Ban hay không. Ở Cong_nhan thuộc tính Ban qui chiếu tới Nguoi_lon và tiếp theo ở Nguoi_lon thuộc tính Ban lại qui chiếu đệ qui về Cong_nhan. Quá trình trên có thể lặp lại nhiều lần và rất khó khẳng định tính nhất quán của các kiểu đó khi các phép qui chiếu lặp lại và không kết thúc.

Sau đây ta dựa vào tính chất của đồ thị đặc trưng cho quan hệ thừa kế giữa các kiểu của lược đồ (gọi là $s$-đồ thị) để kiểm tra tính dừng của quá trình thừa kế và tính phi mẫu của hệ thống kiểu.

## 4. $s$-ĐỒ THỊ

**Định nghĩa 8.** Cho trước lược đồ $\Sigma = \langle T, L, P\rangle$. $s$-đồ thị của lược đồ $\Sigma$ là một đồ thị định hướng có gắn nhãn $G = (V, E)$, trong đó

+ Tập các đỉnh $V = T \cup L$;

+ Tập cung $E = \{(\tau_1, l, \tau_2) \subseteq T \times P \cup \{h\} \times T \cup L; (\tau_1, l, \tau_2) \in E \Leftrightarrow \tau_1$ thừa kế từ $\tau_2$ ($\tau_1$ Sub $\tau_2$) và $l = h$ hoặc $\tau_1 = \{\dots, a : \tau_2, \dots\}$ và $l = a$, nhãn $l = h$ hoặc $l \in P$;

$s$-đồ thị mô tả mối quan hệ giữa các kiểu trong hệ thống. Những cung có nhãn là $h$ trong đồ thị mô tả quan hệ thừa kế cha/con của 2 đỉnh còn những cung có nhãn khác là những thuộc tính của một kiểu (đỉnh xuất phát) có nhận các giá trị thuộc kiểu mô tả ở đỉnh đến theo đường chỉ của mũi tên. Chúng ta nhận thấy trong $s$-đồ thị, những kiểu không ở dạng chuẩn sẽ tương ứng với những đỉnh có các cung đi tới các đỉnh ứng với kiểu được thừa kế và có nhãn là $h$.

*Ví dụ 4.*

type Cong_nhan = {Ho_ten: string; Luong: real; Phan_xuong: string};

type Nhan_vien = {Ho_ten: string; Tuoi: integer; Thu_truong: string};

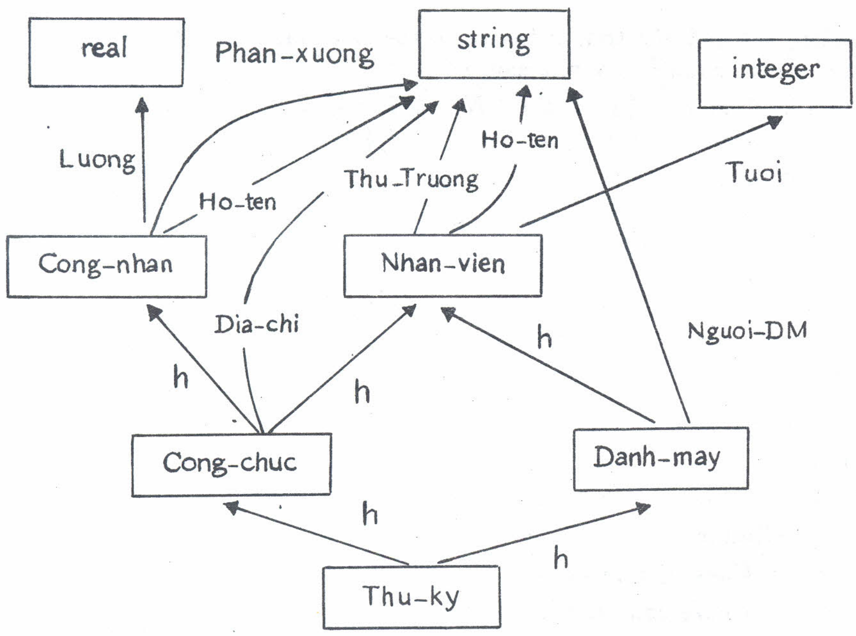
type Danh_may = Nhan_vien {Nguoi_DM: string};

type Cong_chuc = Cong_nhan, Nhan_vien {Dia_chi: string};

type Thu_ky = Cong_chuc, Danh_may{ };

$T$ = {Cong_nhan, Nhan_vien, Danh_may, Cong_chuc, Thu_ky}, $L$ = {integer, real, string} và $P$ = {Ho_ten, Tuoi, Luong, Thu_truong, Nguoi_DM, Dia_chi}.

$s$-đồ thị tương ứng của lược đồ trên là đồ thị mô tả như sau:




Nói chung $s$-đồ thị không phải là đồ thị đơn, nghĩa là có thể có những cung song song và có vòng khuyên (cung tự trỏ). Trực tiếp từ định nghĩa chúng ta suy ra $s$-đồ thị của lược đồ dạng chuẩn là đồ thị không có cung có nhãn là $h$.

Như chúng ta khẳng định từ trước, chính cơ chế thừa kế bội (một kiểu được thừa kế từ nhiều hơn một kiểu cơ sở) thường dẫn tới sự xung đột hay mâu thuẫn trong hệ thống. Như vậy, để kiểm tra xem một lược đồ có xung đột hay không thì chỉ cần kiểm tra trên $s$-đồ thị của nó có những đường đi bắt đầu từ những đỉnh mà kiểu tương ứng không ở dạng chuẩn, có các dãy nhãn (thuộc tính) bằng nhau sau khi loại bỏ các cung $h$ trên các đường đi, có dẫn tới những đỉnh tương ứng với các kiểu không tương thích hay không?

Để hình thức hóa được quá trình mô tả trên, chúng ta sử dụng một số khái niệm sau.

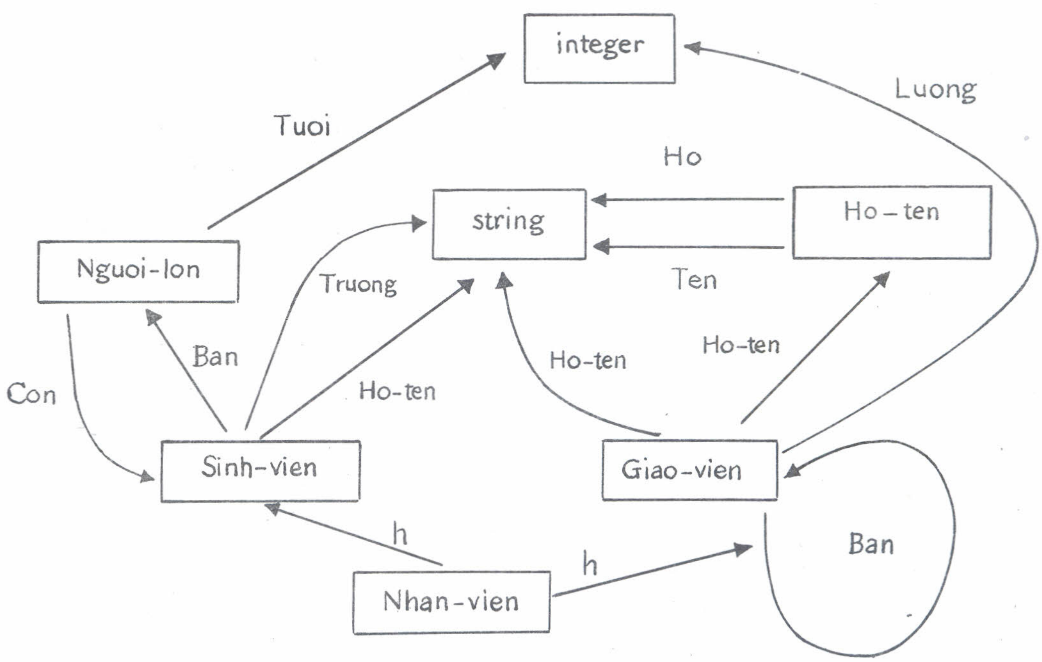
**Định nghĩa 9.** Cho trước $s$-đồ thị $G = (V, E)$, $p = (s, d, \langle a_1.a_2 \dots a_n\rangle)$ là đường đi của $G$ từ đỉnh $s$ đến $d$ nếu $\exists (\tau_i, a_i, \tau_{i+1}) \in E$, $i = 1, 2, \dots, n-1$ và $\tau_1 = s$, $\tau_n = d$, $n \geq 1$.

**Định nghĩa 10.** Hai đường đi trong $s$-đồ thị $G$, $p_1 = (s_1, d_1, \langle a_1.a_2 \dots a_n\rangle)$ và $p_2 = (s_2, d_2, \langle b_1.b_2 \dots b_n\rangle)$ được gọi là tựa tương đẳng, ký hiệu là $p_1 \cong p_2 \Leftrightarrow$ hai dãy nhãn $\langle a_1 a_2 \dots a_n\rangle$, $\langle b_1 b_2 \dots b_n\rangle$ cho cùng một kết quả sau khi loại bỏ đi nhãn có tên $h$ và $d_1 \neq d_2$.

Chúng ta dễ nhận thấy là các dãy sau khi loại bỏ nhãn $h$ của một đường đi chính là dãy các thuộc tính của kiểu tương ứng với đỉnh xuất phát, thu được từ quá trình phân giải thừa kế thông phép hội nhập kiểu $\downarrow$. Trong đồ thị chỉ những đường đi tựa tương đẳng mới có thể dẫn tới xung đột về kiểu thừa kế.

**Định nghĩa 11.** $s$-đồ thị có xung đột (có mâu thuẫn) khi và chỉ khi tồn tại ít nhất 2 đường đi tựa tương đẳng cùng bắt đầu từ một đỉnh (kiểu con không ở dạng chuẩn và thừa kế bội) và dẫn đến 2 đỉnh đích là 2 kiểu không tương thích (phép hội nhập kiểu $\downarrow$ của chúng không xác định). Lược đồ tương ứng với $s$-đồ thị không có mâu thuẫn được gọi là lược đồ phi mâu thuẫn.

*Ví dụ 5.* Xét $s$-đồ thị của lược đồ cho trước trong ví dụ 2.




Xét 2 đường đi tựa tương đẳng trong đồ thị:

$$(\text{Nhan\_vien, Hovaten, } \langle \text{h.Ho\_ten}\rangle) \cong (\text{Nhan\_vien, string, } \langle \text{h.Ho\_ten}\rangle).$$

Hai đường đi này dẫn đến 2 kiểu không tương thích: string (kiểu nguyên thủy) và kiểu Hovaten (được định nghĩa) và string $\downarrow$ Hovaten $= \perp$. Vậy một lần nữa dựa vào $s$-đồ thị chúng ta khẳng định lược đồ cho trước trong ví dụ 2 là có xung đột.

**Định nghĩa 12.** Lược đồ $\Sigma = \langle T,\ L,\ P\rangle$ được gọi là đúng đắn khi và chỉ khi nó là phi mâu thuẫn và với mọi kiểu $\tau \in T$ (không dạng chuẩn $\tau \in I$) quá trình thừa kế là kết thúc.

Từ tính chất của các đường đi tựa tương đẳng chúng ta thu được điều kiện cần và đủ để khẳng định tính đúng đắn của hệ thống như sau.

**Định lý 4.** *Lược độ* $\Sigma = \langle T,\ L,\ P\rangle$ *là đúng đắn* $\Leftrightarrow$ *s-đồ thị tương ứng thỏa mãn điều kiện sau:* $\forall\tau \in I$ *(không ở dạng chuẩn) và với mọi cặp đường đi* $p_1 = (\tau,\ d_1,\ \langle v_1\rangle) \cong p_2 = (\tau,\ d_2,\ \langle v_2\rangle)$, $d_1,\ d_2 \in T \cup L$ *thì*

(i) $d_1,\ d_2 \neq \tau$ *(kết thúc thừa kế) và cả 2 đều không phải là nguyên thủy (không dẫn tới xung đột),*

(ii) *Ngoài ta nếu tồn tại 2 đường đi tựa tương đẳng dạng* $p'_1 = (d'_1,\ d'_1,\ \langle u_1\rangle)$, $p'_2 = (d'_2,\ d'_2\ \langle u_2\rangle)$ *thì* $d_1 \neq d'_1$ *hoặc* $d_2 \neq d'_2$ *(kết thúc thừa kế).*

*Chứng minh. Điều kiện cần:* Chứng minh theo phản ứng. Giả thiết rằng $\Sigma$ là đúng đắn và $\exists\tau \in I$ cùng hai đường đi $p_1 = (\tau,\ d_1,\ \langle v_1\rangle) \cong p_2(\tau,\ d_2,\ \langle v_2\rangle)$, $d_1,\ d_2 \in T \cup L$ mà một trong hai điều kiện (i) hoặc (ii) trong định lý là sai.

a. Nếu điều kiện (i) không đúng, nghĩa là $d_1 = \tau$ (hoặc $d_2 = \tau$) hoặc $d_1 \in L$ (và/hoặc $d_2 \in L$).

Trường hợp đầu tiên, $d_1 = \tau$ thì $p_1 = (\tau,\ \tau\,\langle v_1\rangle) \cong p_2 = (\tau,\ d_2\,\langle v_2\rangle)$. Do vậy khi tính các dãy $\langle v_1\rangle$, $\langle v_2\rangle$ phải tính $d_1 \downarrow d_2$, tức là phải chuyển $\tau$ về dạng chuẩn, mà quá trình này không kết thúc. Điều này mâu thuẫn với giả thiết.

Trường hợp thứ 2, $p_1 = (\tau,\ d_1,\ \langle v_1\rangle) \cong p_2 = (\tau,\ d_2,\ \langle v_2\rangle)$ vậy $d_1 \neq d_2$, mà hoặc $d_1 \in L$ hoặc $d_2 \in L$, suy ra $d_1 \downarrow d_2 = \perp$. Điều này cũng mâu thuẫn với tính đúng đắn của lược đồ.

b. Khi điều kiện (ii) sai, nghĩa là có $p'_1 = (d_1,\ d_1,\ \langle u_1\rangle) \cong p'_2 = (d_2,\ d_2,\ \langle u_2\rangle)$. Khi đó chúng ta có hai đường đi tương ứng $p_3 = (\tau,\ d_1,\ \langle v_1.u_1\rangle) \cong p_4 = (\tau,\ d_2,\ \langle v_2.u_2\rangle)$ nên cần áp dụng quá trình thừa kế đối với $\tau$ để xác định những thuộc tính tiếp theo, mà quá trình này lại không kết thúc vì luôn có 2 chu trình (đường vòng tròn) $p'_1 = (d_1,\ d_1,\ \langle u_1\rangle) \cong p'_2 = (d_2,\ d_2,\ \langle u_2\rangle)$. Do vậy mâu thuẫn với tính đúng đắn của lược đồ.

*Điều kiện đủ:* Giả thiết lược độ $\Sigma$ và s-đồ thị tương ứng $G$ thỏa mãn cả hai tính chất (i) và (ii). Chúng ta xét quan hệ $\prec \subseteq T \times T$, được định nghĩa như sau:

$$\begin{aligned}(\alpha_1,\ \alpha_2) \prec (\delta_1,\ \delta_2) \Leftrightarrow\ & \alpha_1,\ \alpha_2 \in T,\ \delta_1,\ \delta_2 \in T \cup L \text{ và trên } G \text{ có 2 đường đi } g_1 = (\alpha_1,\ \delta_1,\ \langle\cdots\rangle)\\ & \cong g_2 = (\alpha_2,\ \delta_2,\ \langle\cdots\rangle) \text{ và } \exists\tau \in I \text{ cùng 2 đường đi xuất phát từ } \tau\text{:}\\ & e_1 = (\tau,\ \alpha_1,\ \langle\cdots\rangle) \cong e_2 = (\tau\,\alpha_2,\ \langle\cdots\rangle).\end{aligned}$$

Chúng ta dễ dàng chứng minh được rằng $\prec$ là quan hệ thứ tự bộ phận. Mặt khác vì số lượng các kiểu trong hệ thống là hữu hạn nên với mọi dãy các cặp kiểu bất kỳ luôn tồn tại một cặp kiểu cực đại theo thứ tự $\prec$. Vì thế, khi $\tau \in I$ (không dạng chuẩn và thừa kế bội), trong số các cặp đường đi tựa tương đẳng xuất phát từ $\tau$, chúng ta xét một cặp đường đi tới 2 đỉnh cực đại theo quan hệ trên, ví dụ $\omega_1$, $\omega_2$. Tất nhiên $\omega_1$, $\omega_2$ đều khác $\tau$, vì ngược lại thì mâu thuẫn với (i). Do vậy $\omega_1$, $\omega_2$ sẽ là những kiểu mà bất kỳ cặp đường đi nào xuất phát tương ứng từ $\omega_1$, $\omega_2$ đều là tựa tương đẳng, nghĩa là có dãy thuộc tính như nhau sau khi loại bỏ nhãn $h$. Từ điều kiện (ii) và tính chất cực đại của cặp $\omega_1$, $\omega_2$ suy ra những đường đi như thế phải dẫn đến cùng một kiểu (mọi kiểu thừa kế bội đều tương thích), nghĩa là lược đồ tương ứng là đúng đắn. □

## 5. KẾT LUẬN

Tính nhất quán dữ liệu là vấn đề rất quan trọng trong thiết kế và cài đặt các hệ cơ sở dữ liệu. Trong các hệ hướng đối tượng thì vấn đề này lại càng cần phải quan tâm nhiều hơn vì khi sử dụng những công cụ rất mạnh như thừa kế bội từ nhiều lớp đối tượng cơ sở (sử dụng lại nhiều tính chất từ các lớp cha) có thể dẫn tới những xung đột về kiểu và hơn thế nữa để khẳng định được hệ thống là phi mâu thuẫn thì phải kiểm tra được tính dừng của quá trình thừa kế. Trong bài báo này, sau khi giới thiệu một số tính chất quan trọng của quan hệ thừa kế và quan hệ kiểu con, chúng tôi đã nêu điều kiện cần và đủ để kiểm tra tính đúng đắn của một lược đồ cơ sở dữ liệu hướng đối tượng. Những vấn đề tiếp theo còn mở là nghiên cứu để xây dựng, cài đặt những thuật toán hữu hiệu (với độ phức tạp là đa thức) để kiểm tra tính dừng của quá trình thừa kế kiểu và tổng quát hơn là kiểm tra tính đúng đắn của hệ thống.

## TÀI LIỆU THAM KHẢO

[1] Amadio R. M. and Cardelli R., Subtyping recusive Types, *ACM Trans. Program. Lang. Systems* **15** (4) (1993) 575-631.

[2] Donal K., Practical Application of Object-Oriened Techniques to Relational Databases, John Wiley and Sons, 1994.

[3] Đoàn Văn Ban, Một số tính chất của quá trình thừa kế kiểu trong mô hình dữ liệu hướng đối tượng, Tạp chí Tin học và Điều khiển học **15** (3) (1999) 1-10.

[4] Formica A. et al., Object-oriented databbase schema analysis and inheritance processing: a graph-theoretic approach, *Data Knowle. Eng. Journal* **24** (2) (1997) 157-181.

[5] Kim W., Object-oriented databases: definition and research directions, *IEEE Trans. Knowle. Eng.* **2** (3) (1990).

[6] Missikoff M. and Tolati M., MOSAICO - a system for conceptual modeling and rapid prototyping of object-oriented database applications, *Proceeding of the ACM SIGMOD Confe. Minneapolits, MN*, May 24-27, 1995, 508-519.

*Nhận bài ngày 24 - 7 - 1999*

*Viện Công nghệ thông tin*

*Nhận lại sau khi sửa ngày 12 - 10 - 1999*