

Chương 4.

Tổng quan về Cơ Sở Dữ Liệu suy diễn

Chương này gồm các nội dung chính sau:

- 4.1. Nguyên lý của một hệ CSDL suy diễn
 - 4.1.1. Sự hình thành CSDL suy diễn
 - 4.1.2. Các thành phần của CSDL suy diễn
- 4.2. Ngôn ngữ Datalog
 - 4.2.1. Cú pháp trong Datalog
 - 4.2.2. Dạng câu trong Datalog
 - 4.2.3. Các thuật toán suy diễn trong Datalog
 - 4.2.4. Phương pháp chứng minh bằng hợp giải
- 4.3. Biểu diễn CSDL suy diễn trong Datalog
 - 4.3.1. Biểu diễn CSDL suy diễn trong Datalog
 - 4.3.2. Biểu diễn khung nhìn trong Datalog
- 4.4. Truy vấn trong CSDL suy diễn
 - 4.4.1. Câu hỏi trong CSDL suy diễn
 - 4.4.2. Các phương pháp định giá câu trong CSDL suy diễn

Bài tập chương 4

1.1 NGUYÊN LÝ CỦA MỘT HỆ CƠ SỞ DỮ LIỆU SUY DIỄN

4.1.1 Sự hình thành Cơ sở dữ liệu suy diễn

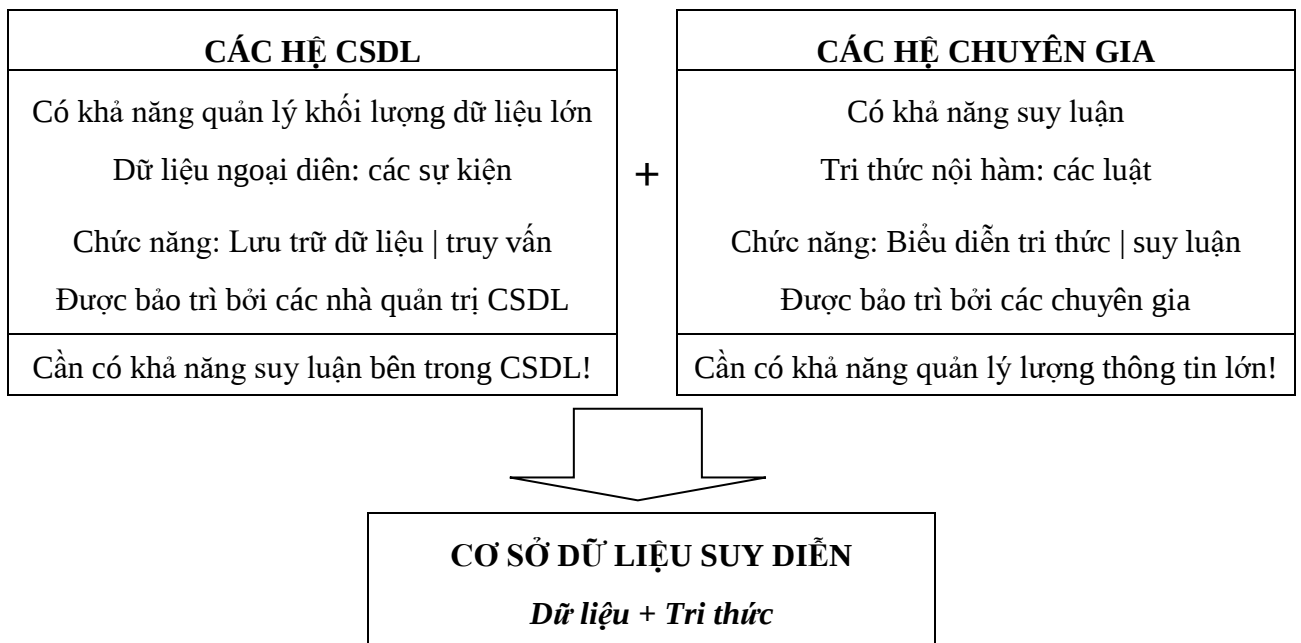
Các hệ quản trị CSDL đã bắt đầu xuất hiện từ những năm 1960 và đã phát triển nhanh chóng. Đặc biệt là từ khi mô hình CSDL quan hệ do Codd.E.F. đề xuất vào đầu những năm 1970, thì công nghệ CSDL đã phát triển rất nhanh và thu được rất nhiều thành tựu trong nhiều ứng dụng khác nhau của CNTT. Đặc điểm chung của các hệ thống CSDL truyền thống là có khả năng lưu trữ khối lượng dữ liệu rất lớn, có khả năng lưu trữ và xử lý phân tán...tuy nhiên chức năng chủ yếu vẫn là lưu trữ và xử lý dữ liệu, hầu như không có khả năng suy luận.

Bên cạnh đó, song song với sự phát triển của các hệ QTCSDL thì các hệ chuyên gia cũng được phát triển. Đặc điểm của các hệ chuyên gia là có thể cung cấp các khả năng suy luận nhằm hỗ trợ việc ra quyết định tuy nhiên lại không có khả năng lưu trữ và xử lý khối lượng dữ liệu lớn.

Một hệ thống mới được đề xuất dựa trên sự kết hợp của hai hệ thống trên, nhằm phát huy những ưu điểm nổi trội của mỗi hệ thống, với mong muốn hệ thống mới vừa có khả năng lưu trữ, xử lý khối lượng dữ liệu lớn, vừa có khả năng tư duy, suy luận với những thông tin đã được lưu trữ. Như vậy sự kết hợp của CSDL và hệ chuyên gia đã cho ra đời hệ thống CSDL suy diễn (Deductive Database).

Như vậy các hệ CSDL suy diễn được xem như sự tích hợp của dữ liệu (như trong một hệ CSDL) và tri thức (như trong một hệ chuyên gia).

Sơ đồ dưới đây mô tả sự kết hợp của các hệ CSDL và các hệ chuyên gia:



Hình 4.1. Sự hình thành CSDL suy diễn

4.1.2 Các thành phần chính của CSDL suy diễn

Để thực hiện được các chức năng của một hệ CSDL và một hệ chuyên gia như đã mô tả trên đây, một hệ CSDL suy diễn cần có các thành phần sau:

4.1.2.1 Một CSDL ngoại diên (Extension Database- EDB)

- EDB là một CSDL quan hệ tiêu chuẩn như mọi hệ CSDL truyền thống, được xây dựng trên một tập các lược đồ quan hệ, có khả năng lưu trữ một khối lượng lớn dữ liệu.
- Các dữ liệu trong EDB gọi là các sự kiện (*facts*), mỗi sự kiện là một bộ của một quan hệ, có thể cập nhật (thêm/sửa/xóa) như là các bộ trong CSDL quan hệ.
- Các sự kiện biểu diễn các thông tin cơ sở (được cho là đúng trong CSDL)

4.1.2.2 Một CSDL nội hàm (Intention Database- IDB)

- IDB là một CSDL chứa các thông tin nội hàm, lưu trữ một tập các luật, cho phép định nghĩa thông tin mới từ các thông tin được lưu trữ là các sự kiện. Có 2 loại luật được lưu trữ trong IDB:
- Các luật suy diễn (*deductive rules*): cho phép suy ra các sự kiện mới từ các sự kiện được lưu trữ trong EDB.
- Các ràng buộc toàn vẹn (*integrity constraints*): được viết dưới dạng các luật, phát biểu các điều kiện mà mỗi trạng thái của CSDL phải thỏa.

4.1.2.3 Ngôn ngữ thao tác CSDL suy diễn

- CSDL suy diễn có thể được hiểu là kết quả của việc áp dụng logic và trí tuệ nhân tạo vào lĩnh vực CSDL truyền thống. Ngôn ngữ được dùng để định nghĩa nội dung và cấu trúc thông tin trong CSDL suy diễn là ngôn ngữ Datalog (logic cho dữ liệu).
- Datalog là ngôn ngữ lập trình logic (tương tự như Prolog) được phát triển dựa trên Logic tân từ cấp một. Ngôn ngữ Datalog thao tác trên các *tân từ ngoại diên* (hay *tân từ cơ sở*) là tên của các quan hệ trong CSDL ngoại diên (EDB) và *tân từ nội hàm* (hay *tân từ dẫn suất*) là tên của các quan hệ trong CSDL nội hàm (IDB). Một CSDL suy diễn được biểu diễn bởi một chương trình Datalog

4.1.2.4 Hệ quản trị CSDL suy diễn:

- Hệ quản trị CSDL suy diễn phải có tất cả các chức năng của một hệ QTCSDDL thông thường, có thể cho phép khai thác và quản lý CSDL một cách tập trung hoặc phân tán, đảm bảo tính tin cậy và an toàn dữ liệu.
- Hệ quản trị CSDL suy diễn còn cho phép suy diễn ra các sự kiện mới (là các sự kiện dẫn suất của các tân từ nội hàm) từ các sự kiện đã có bằng việc sử dụng các quy tắc và các luật logic
- Hệ quản trị CSDL suy diễn cung cấp một thủ tục xử lý câu hỏi, có khả năng trả lời các câu hỏi được phát biểu theo các khung nhìn cũng như theo các tân từ cơ sở.

4.2 NGÔN NGỮ DATALOG

Datalog (logic dữ liệu) là ngôn ngữ được dùng để biểu diễn, xử lý dữ liệu trong một CSDL suy diễn, cũng giống như SQL được sử dụng trong hầu hết các CSDL quan hệ. Cơ sở của ngôn ngữ SQL là đại số quan hệ, còn ngôn ngữ Datalog thì được xây dựng trên nền tảng của logic tân từ cấp một.

Trong phần này, ta sẽ nghiên cứu nhiều khái niệm liên quan đến logic tân từ (hay logic vị từ) cấp một, được sử dụng trong Datalog.

4.2.1 Cú pháp trong Datalog

Cú pháp của Datalog dựa trên cú pháp của Logic tân từ cấp một, (còn gọi là các phép tính tân từ), đó là một ngôn ngữ hình thức được dùng để biểu diễn mối quan hệ giữa các đối tượng và suy ra các mối quan hệ mới từ các mối quan hệ được biết là đúng. Logic tân từ cấp một được xây dựng dựa trên tập ký hiệu và các quy ước về ký pháp như sau:

4.2.1.1 Các ký hiệu trong logic tân từ cấp một:

1. Các biến: thường được ký hiệu là $x, y, z, \dots$
2. Các hằng: thường được ký hiệu là $a, b, c, \dots$
3. Các tân từ: thường được ký hiệu là các chữ cái in hoa $P, Q, R, \dots$ (có thể có chỉ số). Một tân từ có thể có n đối là các biến hay các hằng, chẳng hạn: $P(x), Q(x, y), R(a, b, c) \dots$
4. Các phép toán logic: $\wedge$ (hội), $\vee$ (tuyển), $\neg$ (phủ định), $\rightarrow$ (kéo theo), ở đây thứ tự ưu tiên các phép toán là: $\neg, \wedge, \vee$ và cuối cùng là phép $\rightarrow$.
5. Các hàm: thường được ký hiệu là $f, g, h, \dots$
6. Các lượng từ phổ dụng: với mọi ($\forall$) và tồn tại ($\exists$)

4.2.1.2 Cú pháp trong logic tân từ cấp một

1. **Hạng thức (term):** được định nghĩa đệ quy là:
 - (i) Là một biến hay một hằng,
 - (ii) Hoặc là kết quả của một hàm tác động lên một hạng thức.

Thí dụ 4.1. Với x là biến, a là hằng, f là hàm một biến thì $x, a, b, f(x), f(a), f(f(x)) \dots$ là các hạng thức.

2. **Công thức (formula):** (còn gọi là công thức được thiết lập đúng đắn), là một mệnh đề được xây dựng đúng và được định nghĩa như sau:
 - (i) Nếu P là một tân từ n đối và $t_1, t_2, \dots, t_n$ là các hạng thức thì $P(t_1, t_2, \dots, t_n)$ là một công thức nguyên tố.
 - (ii) Nếu F_1 và F_2 là các công thức thì $F_1 \wedge F_2, F_1 \vee F_2, F_1 \rightarrow F_2, \neg F_1$ là các công thức.
 - (iii) Nếu F là một công thức và x là biến không được lượng từ hóa (còn gọi là biến tự do) trong F thì $\forall x(F)$ và $\exists x(F)$ là các công thức

Thí dụ 4.2. Giả sử P, Q là các tân từ 2 đối, f là hàm 1 biến, x, y là các biến, a, b là các hằng, khi đó ta có các công thức sau:

- $P(b, x)$
- $Q(a, b)$
- $P(b, x) \wedge Q(a, b)$
- $\forall x \exists y [Q(x, y) \vee P(a, x)]$
- $\forall x [P(x, a) \rightarrow Q(b, f(x))]$

4.2.1.3 Ngữ nghĩa của logic tân từ cấp một

Ngữ nghĩa của logic tân từ cấp một được giải thích qua ngữ nghĩa của các thành phần cơ bản của nó, đó là các công thức, các hàm, các tân từ, các hạng thức, các biến và các hằng...

- Một công thức được lý giải (giải thích) như một mệnh đề trên 1 tập đối tượng, tức là có thể cho nó 1 ý nghĩa trên tập đối tượng đó.
- *Miền thể hiện của một công thức* : là tập đối tượng lý giải cho công thức đó, kí hiệu là D .
- Hàm f với n đối có thể được hiểu là 1 ánh xạ: $f : D^n \rightarrow D$
- Một tân từ biểu diễn 1 quan hệ giữa các phần tử của D , có thể đúng, có thể sai.
- Các hằng được lý giải là những đối tượng cụ thể, còn các biến chỉ một đối tượng bất kỳ.

Thí dụ 4.3. Cho công thức:

$$\forall x \exists y [(Thủ_trưởng(x, y) \vee Thủ_trưởng(y, x)) \wedge Cùng_phòng(y, x)] \quad (1)$$

Công thức (1) có thể được lý giải trên tập người $D = \{Khang, Phong, Mai, Lan, Long\}$, D làm thành miền thể hiện của công thức. Các quan hệ hai ngôi *Thủ_trưởng* và *Cùng_phòng* có ý nghĩa rõ ràng trên tập D . Công thức (1) là đúng nếu năm người Khang, Phong, Mai, Lan, Long làm việc trong cùng phòng và có một người là trưởng phòng.

Thí dụ 4.4. Cho công thức: $\forall x (x > [x])$ (2)

Với $[x]$ là hàm “phần nguyên của x ”, làm ứng mọi số thực x với số nguyên lớn nhất không vượt quá x . Chẳng hạn $[3,14] = 3$; $[3] = 3$; $[-3,14] = -4$.

Công thức (2) có thể được lý giải trên tập các số thực R , tức là tập số thực R là một miền thể hiện của công thức (2), tuy nhiên công thức này sẽ không đúng trên miền thể hiện là R . Nếu chọn D là miền các số thực không phải là số nguyên thì công thức (2) sẽ đúng trên D .

- *Các công thức đóng* :

Qua thí dụ trên ta thấy: với 1 miền thể hiện cho trước, mỗi công thức sẽ có 1 giá trị chân lý cho trước, để tránh trường hợp mỗi công thức có thể có nhiều giá trị chân lý, ta chỉ xét các công thức trong đó tất cả các biến đều được lượng từ hóa, gọi là các *công thức đóng*. Chẳng hạn, nếu trong công thức (2), ta bỏ lượng từ $\forall x$ ở đầu công thức thì công thức còn lại không thể xác định được giá trị chân lý duy nhất.

Mọi công thức đóng F đều có một giá trị chân lý duy nhất với một miền thể hiện xác định, ký hiệu là $V(F)$. Với các công thức phức hợp, thì giá trị chân lý được xác định như sau:

- $V(F_1 \wedge F_2) = V(F_1) \wedge V(F_2)$
 - $V(F_1 \vee F_2) = V(F_1) \vee V(F_2)$
 - $V(\neg F_1) = \neg V(F_1)$
 - $V(\forall x F) = V(F_{x=a}) \wedge V(F_{x=b}) \wedge \dots$ trong đó $a, b, \dots$ là các hằng của miền thể hiện D .
 - $V(\exists x F) = V(F_{x=a}) \vee V(F_{x=b}) \vee \dots$ trong đó $a, b, \dots$ là các hằng của miền thể hiện D .
 - $V[P(a_1, a_2, \dots, a_n)]$ có giá trị đúng (True) nếu tân từ $P(a_1, a_2, \dots, a_n)$ là đúng, và ngược lại.
- *Mô hình của một tập công thức đóng:* là một miền thể hiện làm cho mọi công thức của tập này đều đúng.

Chẳng hạn, mô hình của tập gồm một công thức (2) là tập tất cả các số thực không phải là số nguyên. (miền D trong thí dụ 4)

4.2.2 Dạng câu trong Datalog

Với mọi công thức đóng F , ta có thể biến đổi để đưa nó về dạng chính tắc không chứa các lượng từ, được gọi là dạng câu (clause).

Dạng câu của công thức đóng: là một công thức đóng có dạng:

$$P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q_1 \vee Q_2 \vee \dots \vee Q_m \quad (3)$$

Trong đó các P_i, Q_j là các literal dương (là các công thức nguyên tố không có dấu phủ định). Trong câu (3), phần vế trái (trước dấu $\rightarrow$) gọi là phần điều kiện, còn phần vế phải là phần kết luận.

Khi $m=1$, câu (3) được gọi là câu Horn và có dạng:

$$P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q \quad (4)$$

Khi cần nhấn mạnh tới phần kết luận, ta thường viết các câu (3) dưới dạng:

$$Q_1 \vee Q_2 \vee \dots \vee Q_m \leftarrow P_1 \wedge P_2 \wedge \dots \wedge P_n \quad (5)$$

Dấu mũi tên $\leftarrow$ được đọc là “nếu”

Với câu Horn (4) thường viết là: $Q \leftarrow P_1 \wedge P_2 \wedge \dots \wedge P_n$, và được đọc là “ Q nếu $P_1, P_2, \dots, P_n$ ”. Q được gọi là đầu luật, còn $P_1 \wedge P_2 \wedge \dots \wedge P_n$ gọi là thân luật.

Áp dụng một số công thức biến đổi logic, công thức (3) tương đương với công thức:

$$(\neg P_1) \vee (\neg P_2) \vee \dots \vee (\neg P_n) \vee Q_1 \vee Q_2 \vee \dots \vee Q_m \quad (6)$$

Như vậy, từ (6), ta có định nghĩa khác cho câu: *một câu là một tuyển của các literal*.

Đưa các công thức đóng về dạng câu: Với công thức đóng F bất kỳ, trước hết ta cần loại hết các lượng từ theo các bước sau: (để đưa F về dạng chính tắc không chứa các lượng từ)

Bước 1. Loại bỏ các lượng từ phổ dụng $\forall$: Với giả thiết rằng tất cả các biến đều được lượng từ hoá phổ dụng, ta có thể loại bỏ các lượng từ này.

Bước 2. Loại bỏ các lượng từ tồn tại $\exists$ (nếu có): Công thức F với biến x có lượng từ tồn tại, ta có thể loại bỏ lượng từ này nhờ áp dụng công thức:

$$\exists x(F) = (F_{x=a}) \vee (F_{x=b}) \vee \dots \text{ trong đó } a, b, \dots \text{ là các hằng của miền thể hiện } D.$$

Sau đó sử dụng các luật tương đương sau đây để đưa về dạng câu (3) hoặc về dạng tuyển của các literal (6).

▪ Cho E và F là các công thức trong logic cấp một, khi đó ta có các luật tương đương:

$$(i) \quad E \rightarrow F \equiv (\neg E) \vee F$$

$$(ii) \quad E \leftrightarrow F \equiv (E \rightarrow F) \wedge (F \rightarrow E)$$

$$(iii) \quad \neg(\neg E) \equiv E$$

$$(iv) \quad \neg(E \vee F) \equiv (\neg E) \wedge (\neg F)$$

$$(v) \quad \neg(E \wedge F) \equiv (\neg E) \vee (\neg F)$$

$$(vi) \quad \neg(\exists x(F)) \equiv \forall x(\neg F)$$

$$(vii) \quad \neg(\forall x(F)) \equiv \exists x(\neg F)$$

Thí dụ 4.5. Hãy chuyển các công thức đóng dưới đây thành công thức dạng câu:

$$1/. \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z)).$$

Giải: Tất cả các biến đã được lượng từ hóa phổ dụng, ta chỉ cần loại bỏ tất cả các lượng từ phổ dụng để nhận được công thức ở dạng câu (3):

$$(P(x, y) \wedge P(y, z)) \rightarrow P(x, z)$$

Có thể đưa về dạng tuyển của các literal như dạng câu (6) nhờ áp dụng luật tương đương (iii):

$$\neg P(x, y) \vee \neg P(y, z) \vee P(x, z)$$

$$2/. \forall x \forall y (P(x, y) \rightarrow \exists z Q(x, z)), \text{ với } z_1, z_2, \dots, z_n \text{ là giá trị của biến } z \text{ trong miền thể hiện } D.$$

Giải: Trước hết loại bỏ các lượng từ phổ dụng với các biến x và y, ta nhận được công thức :

$$P(x, y) \rightarrow \exists z Q(x, z)$$

Áp dụng công thức (ii) để loại lượng từ tồn tại, ta nhận được công thức dạng câu (3):

$$P(x, y) \rightarrow (Q(x, z_1) \vee Q(x, z_2) \vee \dots \vee Q(x, z_n))$$

Áp dụng công thức (iii) nhận được dạng câu là tuyển các literal:

$$\neg P(x, y) \vee Q(x, z_1) \vee Q(x, z_2) \vee \dots \vee Q(x, z_n) \text{ (câu dạng (6))}$$

4.2.3 Các thuật toán suy diễn trong Datalog

Thuật toán suy diễn là một thủ tục để chứng minh một công thức T từ tập các công thức $\{A_1, A_2, \dots, A_n\}$ đã được biết là đúng. T còn được gọi là định lý cần chứng minh, còn $A_1, A_2, \dots, A_n$ gọi là các tiên đề. Nếu tồn tại một chứng minh của T từ $\{A_1, A_2, \dots, A_n\}$ thì ta ký hiệu:

$$\{A_1, A_2, \dots, A_n\} \vdash T.$$

Để chứng minh một định lý, thuật toán suy diễn sẽ dùng các quy tắc suy diễn để từ các tiên đề suy dẫn ra một dãy các công thức, mà định lý cần chứng minh là công thức cuối cùng được suy dẫn ra.

Một quy tắc suy diễn là một quy tắc cho phép sinh ra một công thức từ hai hoặc nhiều công thức. Có hai quy tắc suy diễn quen biết trong logic tân từ cấp một:

4.2.3.1 Quy tắc suy diễn *Modus ponens*.

Modus ponens trong tiếng Latin có nghĩa là *phương pháp khẳng định*, thường được sử dụng trong thuyết “Tam đoạn luận”. Quy tắc này cho phép suy ra Q từ $P \rightarrow Q$ và P , có nghĩa là nếu $P \rightarrow Q$ và P đã được chứng minh thì Q cũng được chứng minh.

Có thể hiểu quy tắc khẳng định này như sau: Cho một luật ‘ $P \rightarrow Q$ ’ (được thừa nhận là đúng), nếu quan sát được sự kiện ‘ P ’ (là đúng) thì suy diễn ra sự kiện ‘ Q ’ (là đúng).

$$\begin{array}{l} \text{Ký hiệu quy tắc } \textit{Modus ponens}: \quad \frac{P \rightarrow Q, P}{Q} \end{array} \quad (7)$$

Quy tắc “đảo” của *Modus ponens* là quy tắc *Modus tollens*, còn gọi là “*phương pháp phủ định*”

$$\begin{array}{l} \text{Ký hiệu quy tắc } \textit{Modus tollens}: \quad \frac{P \rightarrow Q, \neg Q}{\neg P} \end{array} \quad (8)$$

Thí dụ 4.6.

1/. Suy diễn theo quy tắc *Modus ponens*:

Luật: Nếu hôm nay là thứ bảy thì tôi sẽ đến thư viện. ($P \rightarrow Q$)

Quan sát: Hôm nay là thứ bảy. (P)

Suy diễn: Vậy tôi sẽ đến thư viện. (Q).

2/. Suy diễn theo quy tắc *Modus tollens*:

Luật: Nếu hôm nay là thứ bảy thì tôi sẽ đến thư viện. ($P \rightarrow Q$)

Quan sát: Hôm nay tôi không đến thư viện. ($\neg Q$).

Suy diễn: Vậy hôm nay không phải là thứ bảy. ($\neg P$)

4.2.3.2 Quy tắc riêng biệt hóa

Quy tắc riêng biệt hóa cho phép suy ra $F(a)$ từ công thức: $\forall x F(x)$.

Một cách trực quan, có nghĩa là nếu $F(x)$ đã được chứng minh (là đúng) với mọi giá trị của x , thì $F(a)$ cũng được chứng minh là đúng..

$$\begin{array}{l} \text{Ký hiệu quy tắc } \textit{riêng biệt hóa}: \quad \frac{\forall x F(x)}{F(a)} \end{array} \quad (9)$$

4.2.4 Phương pháp chứng minh bằng hợp giải

4.2.4.1 Thuật toán hợp nhất

Thuật toán hợp nhất cho phép so sánh hai công thức nguyên tố (là hai tân từ với n đối), quyết định xem hai công thức nguyên tố có thể được làm đồng nhất nhờ các phép thế hay không. Thủ tục này là một phần quan trọng trong phần lớn các phương pháp suy diễn trong logic tân từ.

Thuật toán hợp nhất của hai công thức nguyên tố được thực hiện như sau:

- Cho hai công thức là hai tân từ cùng tên, cùng số đối:

$$L_1 = P(t_1, t_2, \dots, t_n); L_2 = P(u_1, u_2, \dots, u_n).$$

- Tìm cách thay thế các biến trong mỗi công thức bằng các hạng thức phù hợp (là các hằng các biến, các hàm...), với mục đích làm cho hai công thức trở thành giống hệt nhau. Tức là sử dụng các phép gán: $S = \{t_1 := a_1; t_2 := a_2; \dots; u_1 = b_1; u_2 = b_2; \dots\}$, trong đó các a_i, b_i là các hằng hay biến phù hợp làm cho $L_1 = L_2$, thì hai công thức L_1 và L_2 được gọi là hợp nhất được.
- Tập các phép gán S gọi là phép thế S , ký hiệu là $S = [t_1/a_1; t_2/a_2; \dots; s_1/b_1; u_2/b_2; \dots]$.
- Nếu L_1 và L_2 là hợp nhất được bởi phép thay thế S thì ta kí hiệu $L_1[S] = L_2[S]$.

Chú ý rằng hai công thức $L_1 = P(t_1, t_2, \dots, t_n)$ và $L_2 = P(u_1, u_2, \dots, u_n)$ không phải bao giờ cũng hợp nhất được.

Thí dụ 4.7.

- Hai công thức $P(x, y)$ và $P(a, b)$ là hợp nhất được với phép thế $S = [x/a; y/b]$.
- Hai công thức $P(f(x), a)$ và $P(f(f(a)), y)$ là hợp nhất được với phép thế $S = [x/f(a); y/a]$.
- Hai công thức $P(f(x), x)$ và $P(f(f(x)), x)$ là không thể hợp nhất được vì x không thể đồng thời là $f(x)$ và x , cho nên không thể có phép thế nào để hai công thức trên trở thành giống nhau.

4.2.4.2 Quy tắc suy diễn Robinson

Quy tắc suy diễn Robinson là một quy tắc suy diễn tổng quát cho các công thức của logic tân từ cấp một ở dạng câu. Quy tắc suy diễn Robinson được phát biểu như sau:

Quy tắc Robinson

Cho C_1 và C_2 là hai công thức có dạng câu:

$$C_1 = F_1 \vee L_1 \text{ và } C_2 = F_2 \vee \neg L_2 \text{ (hoặc } L_2 \rightarrow F_2),$$

trong đó tồn tại một phép thế S để $L_1[S] = L_2[S]$. Khi đó quy tắc Robinson cho phép suy diễn ra công thức: $F_1[S] \vee F_2[S]$.

- Ký hiệu quy tắc suy diễn Robinson :
$$\frac{F_1 \vee L_1, F_2 \vee \neg L_2, L_1[S] = L_2[S]}{F_1[S] \vee F_2[S]} \quad (10)$$
- Công thức $F_1[S] \vee F_2[S]$ được gọi là giải thức của C_1 và C_2 .

4.2.4.3 Phương pháp chứng minh bằng hợp giải.

Quy tắc Robinson có tính đầy đủ, tức là mọi công thức có thể chứng minh từ một tập các tiên đề thì sẽ được suy ra bằng việc áp dụng liên tiếp quy tắc Robinson cho các tiên đề và giải thức thu được.

Phương pháp hợp giải cho phép áp dụng nhiều lần quy tắc Robinson để chứng minh một định lý T từ tập các tiên đề $\{A_1, \dots, A_2, \dots, A_n\}$ bằng phương pháp phản chứng, theo các bước sau:

1. Biểu diễn các tiên đề và phủ định của định lý ($\neg T$) dưới dạng tập các câu C.
2. Áp dụng quy tắc Robinson cho mỗi cặp công thức có thể hợp nhất được trong tập câu C, để nhận được một công thức mới là một giải thức, như vậy số câu trong C sẽ giảm dần cho tới khi thu được tập câu nhỏ nhất, không thể hợp giải được nữa, tập câu cuối cùng này có thể là tập câu rỗng (tập $C = \emptyset$).

Khi kết quả là câu rỗng (tất cả các câu trong C đều bị triệt tiêu), khi đó miền thể hiện của tập câu là rỗng, tức là nó không thể được thỏa mãn. Nhưng tập các tiên đề được giả thiết là đúng, vậy phủ định của định lý bị bác bỏ, do đó định lý T được chứng minh.

Trường hợp ngược lại, không thể thu được kết quả là câu rỗng, thì không thể chứng minh được định lý T từ tập tiên đề đã cho.

Chú ý: Trong quy tắc Robinson thì F_1, F_2 có thể là các công thức rỗng, khi đó ta có các trường hợp riêng của quy tắc Robinson:

$$1/. \frac{L_1, F_2 \vee \neg L_2, L_1[S] = L_2[S]}{F_2[S]}, \text{ (khi } F_1 = \emptyset \text{)} \quad (11)$$

$$2/. \frac{F_1 \vee L_1, \neg L_2, L_1[S] = L_2[S]}{F_1[S]}, \text{ (khi } F_2 = \emptyset \text{)} \quad (12)$$

$$3/. \frac{L_1, \neg L_2, L_1[S] = L_2[S]}{\Phi}, \text{ (khi cả } F_1 = \emptyset, F_2 = \emptyset \text{)} \quad (13)$$

Trong đó trường hợp 3 được áp dụng khá phổ biến : nếu trong tập câu C, có cặp công thức là các tân từ cùng tên, là phủ định của nhau và hợp nhất được thì loại bỏ cặp công thức đó.

Thí dụ 4.8. Áp dụng phương pháp hợp giải, chứng minh định lý T: $P(a, c)$ từ tập các tiên đề sau:

(A1): $P(a, b)$,

(A2): $P(b, c)$,

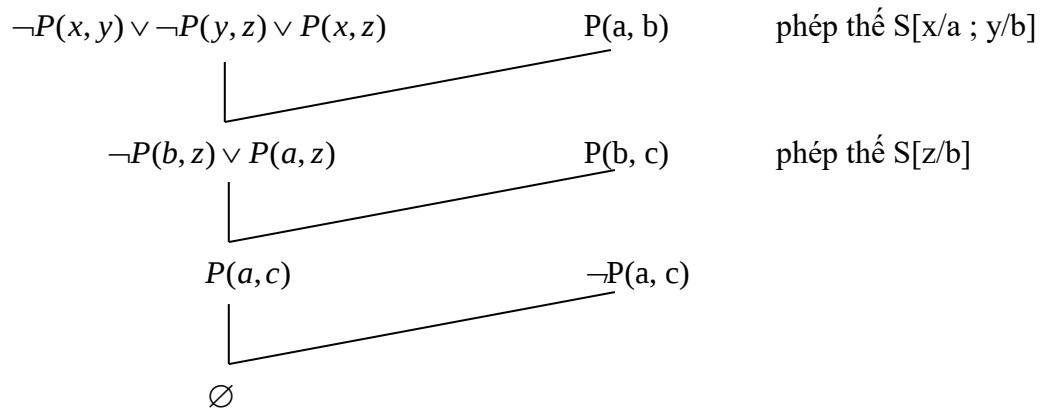
(A3): $\forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z))$.

Ta thực hiện các bước:

1. Biểu diễn các tiên đề và phủ định của định lý ($\neg T$) dưới dạng tập các câu:

$$C = \{ P(a, b), P(b, c), \neg P(x, y) \vee \neg P(y, z) \vee P(x, z), \neg P(a, c) \}$$

2. Áp dụng quy tắc Robinson nhiều lần theo sơ đồ sau:



Hình 4.2. Sơ đồ chứng minh bằng hợp giải cho định lý T.

Từ sơ đồ trên, ta thấy kết quả tập $C = \emptyset$. Vậy định lý T: $P(a, c)$ đã được chứng minh.

4.3 BIỂU DIỄN CƠ SỞ DỮ LIỆU SUY DIỄN TRONG DATALOG

4.3.1 Biểu diễn CSDL suy diễn trong Datalog

Cơ sở dữ liệu suy diễn có thể được biểu diễn như một chương trình Datalog, gồm các thành phần chính sau đây:

- Một tập các sự kiện (facts)
- Một tập các luật suy diễn (deductive rules)
- Một tập các ràng buộc toàn vẹn (integrity constraints)

Các thành phần trên đều đều có thể được biểu diễn bởi các câu Horn tổng quát có dạng

$$A_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_n \quad (\text{với } n \geq 0) \quad (14)$$

trong đó A_0 là một nguyên tử (gọi tắt của công thức nguyên tử, là một tân từ n đối) biểu diễn phần kết luận, còn mỗi L_i là một literal, biểu diễn một điều kiện trong phần điều kiện. Mỗi câu Horn dạng (14) được coi là một luật: “nếu $L_1 \wedge L_2 \wedge \dots \wedge L_n$ thì A_0 ”, hay để nhấn mạnh phần kết luận, luật được đọc là: “ A_0 nếu $L_1 \wedge L_2 \wedge \dots \wedge L_n$ ” với A_0 được gọi là *đầu luật*, còn $L_1 \wedge L_2 \wedge \dots \wedge L_n$ gọi là *thân luật*. Các biến trong phần kết luận (đầu luật) và trong các điều kiện (thân luật) được giả thiết là được lượng từ hóa phổ dụng trên toàn bộ công thức, vì vậy, các lượng từ phổ dụng ($\forall$) luôn được bỏ qua.

Trong nhiều trường hợp, để cho ngắn gọn, câu Horn (14) có thể viết dưới dạng:

$$A_0 \leftarrow L_1, L_2, \dots, L_n \quad (15)$$

Chúng ta sẽ không phân biệt hai cách viết (14) và (15).

Ta sẽ xem xét cách biểu diễn ba thành phần trên đây của CSDL suy diễn trong Datalog.

4.3.1.1 Biểu diễn tập sự kiện EDB

Các sự kiện được biết là đúng trong CSDL suy diễn, vì vậy nó có thể biểu diễn bởi các luật có phần thân là rỗng (do sự kiện luôn đúng, không cần điều kiện gì). Do đó mỗi sự kiện được biểu diễn bởi đầu luật là một nguyên tố (tân từ n đối, với các đối được gán giá trị là hằng), không cần có thân luật và mũi tên ngược ($\leftarrow$) cũng được bỏ đi cho gọn.

Chẳng hạn các sự kiện phát biểu rằng Mai là mẹ của Bách và Dương là bố của Tân được biểu diễn bởi:

- Mẹ (Mai, Bách)
- Bố (Dương, Tân)

Một CSDL suy diễn chỉ chứa các sự kiện cơ sở, tức là các sự kiện trong EDB, đó là các công thức nguyên tố, trong đó các hạng thức t_i đều là các hằng.

Tân từ ứng với sự kiện cơ sở gọi là tân từ cơ sở, là tân từ cùng tên và các đối là các biến. Chẳng hạn với hai sự kiện trên ta sẽ có các tân từ cơ sở tương ứng là

- Mẹ (x, y)
- Bố (x, y)

4.3.1.2 Biểu diễn tập các luật suy diễn (deductive rules)

Các luật suy diễn định nghĩa các thông tin nội hàm, là thông tin không được lưu trữ hiện trong CSDL suy diễn. Thông tin nội hàm thu thập từ các sự kiện về các tân từ dẫn xuất.

Tân từ dẫn xuất P là tập tất cả các luật suy diễn có cùng đầu luật là P.

Chẳng hạn, hai luật:

- “nếu x là Bố của y thì x là Cha_mẹ của y”
- “nếu x là Mẹ của y thì x là Cha_mẹ của y”,

sẽ định nghĩa tân từ dẫn xuất mới “Cha_mẹ(x, y)” được biểu diễn là:

- Cha_mẹ (x, y) $\leftarrow$ Bố (x, y) (đọc: x là cha mẹ của y nếu x là bố của y)
- Cha_mẹ (x, y) $\leftarrow$ Mẹ (x, y)

Các sự kiện ứng với các tân từ dẫn xuất gọi là các sự kiện dẫn xuất, cũng được coi là đúng. Các sự kiện này được coi là các thông tin nội hàm, và không được lưu trữ hiện trong CSDL suy diễn. (không lưu trữ rõ ràng như các sự kiện trong EDB)

4.3.1.3 Biểu diễn tập các ràng buộc toàn vẹn (integrity constraints)

Các ràng buộc toàn vẹn thường được biểu diễn bằng các luật (câu) có đầu luật rỗng, còn gọi là một phủ định hay một phản bác, có dạng:

$$\leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_n \text{ (với } n \geq 1 \text{)} \quad (16)$$

Ràng buộc toàn vẹn (16) khẳng định rằng $L_1 \wedge L_2 \wedge \dots \wedge L_n$ không bao giờ đúng trong CSDL chứa ràng buộc toàn vẹn này.

Để cho thống nhất với cách viết các luật, đầu mỗi ràng buộc toàn vẹn (IC) cho phép viết một tân từ không nhất quán IC1, IC2, ... nhằm đặt tên cho ràng buộc đó.

Chẳng hạn, một ràng buộc toàn vẹn khẳng định rằng không ai có thể vừa là Bố, vừa là Mẹ, có thể được biểu diễn là:

$$IC2 \leftarrow \text{Bố}(x, y) \wedge \text{Mẹ}(x, z)$$

Nếu một sự kiện IC_i đúng trong một trạng thái CSDL nào đó, thì ràng buộc tương ứng bị vi phạm trong trạng thái đó của CSDL.

Kết luận: một CSDL suy diễn D là một bộ ba $D = \{F, DR, IC\}$, trong đó F là tập hữu hạn các sự kiện cơ sở (hay sự kiện), DR là tập hữu hạn các luật suy diễn và IC là tập hữu hạn các ràng buộc toàn vẹn. Tập F là CSDL ngoại diễn (EDB), còn DR và IC làm thành CSDL nội hàm (IDB).

Các tân từ trong CSDL suy diễn được chia thành hai loại:

- *Tân từ cơ sở:* là các tân từ ứng với các sự kiện, còn gọi là tân từ ngoại diễn. Tân từ cơ sở xuất hiện trong EDB và có thể xuất hiện trong thân các luật trong IDB (các luật suy diễn và các ràng buộc).
- *Tân từ dẫn xuất:* là các tân từ được định nghĩa nhờ vào một hay một số luật suy diễn nào đó, còn gọi là tân từ nội hàm hay khung nhìn. Tân từ dẫn xuất chỉ xuất hiện trong IDB, các sự kiện ứng với các tân từ dẫn xuất gọi là sự kiện dẫn xuất (là một tân từ dẫn xuất với các đối là các hằng). Sự kiện dẫn xuất không được lưu trữ hiển trong CSDL suy diễn, chỉ được xuất hiện trong các khung nhìn được định nghĩa bởi một tân từ dẫn xuất.

4.3.1.4 Thí dụ về biểu diễn một CSDL suy diễn.

Thí dụ 4.9. Cho một CSDL suy diễn mô tả các mối quan hệ trong một gia tộc.

EDB : Các sự kiện cơ sở

- Mẹ (Mai, Bách)
- Bố (Dương, Tân)
- Bố (Phát, Mai)

IDB : Các luật suy diễn

- Cha_mẹ $(x, y) \leftarrow \text{Bố}(x, y)$
- Cha_mẹ $(x, y) \leftarrow \text{Mẹ}(x, y)$
- Bà $(x, y) \leftarrow \text{Mẹ}(x, z) \wedge \text{Cha_mẹ}(z, y)$
- Tổ_tiên $(x, y) \leftarrow \text{Cha_mẹ}(x, y)$
- Tổ_tiên $(x, y) \leftarrow \text{Cha_mẹ}(x, z) \wedge \text{Tổ_tiên}(z, y)$
- Tổ_tiên_không_trực_tiếp $(x, y) \leftarrow \text{Tổ_tiên}(x, y) \wedge \neg \text{Cha_mẹ}(x, y)$

ICs : Các ràng buộc toàn vẹn:

- $IC1(x) \leftarrow \text{Cha_mẹ}(x, x)$
- $IC2(x) \leftarrow \text{Bố}(x, y) \wedge \text{Mẹ}(x, z)$

Chú ý rằng tân từ không nhất quán cũng có thể chứa biến, nhằm xác định cá thể vi phạm ràng buộc toàn vẹn.

4.3.2 Biểu diễn khung nhìn trong CSDL suy diễn

Khái niệm khung nhìn (View) được dùng trong CSDL để phân ranh giới nội dung CSDL với một nhóm người dùng. Khung nhìn là một cấu trúc dữ liệu ảo, được dẫn xuất từ các sự kiện cơ sở hay các khung nhìn khác nhờ một hàm định nghĩa.

Trong CSDL suy diễn, các khung nhìn tương ứng với các tân từ dẫn xuất, và được định nghĩa nhờ vào các luật suy diễn. Chẳng hạn ta có khung nhìn “Bà” được định nghĩa bởi một luật suy diễn có đầu luật là tân từ Bà (x, y):

$$\circ \text{ Bà (x, y)} \leftarrow \text{Mẹ (x, z)} \wedge \text{Cha_mẹ (z, y)}$$

Khung nhìn được gọi là “đệ quy”, nếu nó được định nghĩa bởi các luật đệ quy, là các luật mà có tân từ ở đầu luật cũng xuất hiện trong thân luật. Khung nhìn “Tổ_tiên” dưới đây là một khung nhìn đệ quy:

$$\circ \text{ Tổ_tiên (x, y)} \leftarrow \text{Cha_mẹ (x, y)}$$

$$\circ \text{ Tổ_tiên (x, y)} \leftarrow \text{Cha_mẹ (x, z)} \wedge \text{Tổ_tiên (z, y)}$$

Cũng giống như trong các CSDL quan hệ, khung nhìn trong CSDL suy diễn có các ưu việt sau:

- Khung nhìn làm đơn giản giao diện người dùng, vì có thể bỏ qua những dữ liệu không liên quan đến người dùng. Chẳng hạn với luật:

$$\circ \text{ Bà(x, y)} \leftarrow \text{Mẹ(x, z)} \wedge \text{Cha_mẹ(z, y)},$$

thì khung nhìn Bà(x, y) chỉ cung cấp thông tin về người bà x và người cháu y, còn thông tin về cha mẹ (tức z) được che dấu bởi định nghĩa của khung nhìn.

- Khung nhìn hỗ trợ tính độc lập logic của dữ liệu, vì nó cho phép thay đổi cấu trúc logic của dữ liệu trong CSDL suy diễn, mà không cần phải tiến hành các thay đổi tương ứng cho các luật khác. Chẳng hạn, giả sử tân từ cơ sở Bó (x, y) phải được thay bằng hai tân từ Bó1(x, y) và Bó2(x, y), mỗi tân từ chứa một tập con các xuất hiện của Bó(x, y), khi đó ta xem Bó(x, y) là một tân từ khung nhìn được định nghĩa bởi:

$$\circ \text{ Bó (x, y)} \leftarrow \text{Bó1 (x, y)}$$

$$\circ \text{ Bó (x, y)} \leftarrow \text{Bó2 (x, y)},$$

thì ta không cần phải thay đổi các luật tham chiếu tới tân từ gốc Bó (x, y)

- Khung nhìn cung cấp một biện pháp bảo vệ vì chúng ngăn ngừa người dùng truy cập tới dữ liệu bên ngoài khung nhìn của họ.

4.4 TRUY VẤN CÂU HỎI TRONG CƠ SỞ DỮ LIỆU SUY DIỄN

4.4.1 Câu hỏi trong CSDL suy diễn

Các hệ QTCSDL suy diễn phải cung cấp một cơ chế xử lý câu hỏi, có khả năng trả lời các câu hỏi được phát biểu theo các khung nhìn cũng như theo các tân từ cơ sở của một CSDL suy diễn. Một thủ tục định giá câu hỏi sẽ tìm câu trả lời phù hợp với ngữ nghĩa của CSDL.

Theo cú pháp của Datalog, một câu hỏi Q trên CSDL suy diễn có dạng:

Q: ? W(x)

trong đó:

- x là một vector biến và / hoặc các hằng.
- W(x) là một hội của các literal

Câu trả lời cho câu hỏi là một tập các giá trị của x sao cho W(x) là đúng và phù hợp với EBD và IDB trong CSDL.

Thí dụ 4.10. Cho một CSDL suy diễn trong Datalog:

CSDL_Gia tộc

EDB : Các sự kiện cơ sở

- Bó (An, Dương)
- Bó (An, Mai)
- Bó (Giáp, An)
- Bó (Giáp, Hồng)
- Mẹ (Xuân, An)
- Mẹ (Xuân, Hồng)
- Mẹ (Hồng, Giang)
- Mẹ (Giang, Minh)

IDB : Các luật suy diễn:

- Cha_mẹ (x, y) $\leftarrow$ Bó (x, y) (Luật R1)
- Cha_mẹ (x, y) $\leftarrow$ Mẹ (x, y) (Luật R2)
- Bà (x, y) $\leftarrow$ Mẹ (x, z) $\wedge$ Cha_mẹ (z, y) (Luật R3)
- Tổ_tiên (x, y) $\leftarrow$ Cha_mẹ (x, y) (Luật R4)
- Tổ_tiên (x, y) $\leftarrow$ Cha_mẹ (x, z) $\wedge$ Tổ_tiên (z, y) (Luật R5)

Với “CSDL_Gia tộc” trên đây, có thể đặt một số câu hỏi như sau:

1/. ? Tổ_tiên (Giáp, Mai).

- trả về giá trị đúng (True) nếu Giáp là tổ tiên của Mai. Câu trả lời là sai (False) trong trường hợp trái lại.

2/. ? Tổ_tiên (Giáp, x).

- trả về tập tất cả những người x nhận Giáp là tổ tiên (tập các con cháu của Giáp).

3/. ? Tổ_tiên (x, Mai).

- trả về tập tất cả những người x là tổ tiên của Mai (tập cha mẹ, ông bà, cụ kỵ...của Mai).

4/. ? Tổ_tiên (y, Mai) $\wedge$ Tổ_tiên (y, Dương)

- trả về tập tất cả những người y là tổ tiên chung của Mai và Dương.

4.4.2 Các phương pháp định giá câu trong CSDL suy diễn

Có ba cách tiếp cận cơ bản để tìm câu trả lời cho một câu hỏi Q trong CSDL suy diễn:

(a) *Tiếp cận dưới- lên* (xâu chuỗi tiến: forward chaining):

Theo cách tiếp cận này, thủ tục định giá câu hỏi bắt đầu từ những sự kiện cơ sở và áp dụng tất cả các luật suy diễn để tìm các sự kiện mới (các sự kiện dẫn xuất), cho đến khi không thể suy ra thêm sự kiện mới nào. Sau đó câu hỏi sẽ được định giá bằng cách so khớp với toàn bộ tập sự kiện cơ sở và sự kiện dẫn xuất.

(b) *Tiếp cận trên-xuống* (xâu chuỗi lùi: backward chaining):

Theo cách tiếp cận này, thủ tục định giá câu hỏi bắt đầu từ câu hỏi Q, và áp dụng các luật để tìm những câu hỏi con tương đương với Q gần với các sự kiện cơ sở hơn, cho đến khi tìm được những câu hỏi con có thể so khớp được với các sự kiện trong EDB, từ đó có câu trả lời cho câu hỏi Q.

(c) *Tiếp cận bằng ma tập* (magic sets):

Đây là cách tiếp cận tổ hợp cả hai cách tiếp cận trên, nhằm đạt được các ưu điểm của mỗi cách tiếp cận.

Chúng ta sẽ trình bày chi tiết các phương pháp định giá câu hỏi trong CSDL suy diễn, dựa trên các cách tiếp cận trên. Các phương pháp này sẽ được minh họa trên CSDL **Gia tộc**:

4.4.2.1 Định giá câu hỏi theo tiếp cận dưới – lên (forward chaining)

Đây là một thủ tục ngây thơ (the naive procedure) cho việc định giá câu hỏi. Theo cách tiếp cận này, thủ tục định giá câu hỏi bắt đầu từ những sự kiện cơ sở và áp dụng tất cả các luật suy diễn để tìm các sự kiện mới cho đến khi không thể suy ra thêm sự kiện mới nào. Sau đó câu hỏi sẽ được định giá bằng cách so khớp với toàn bộ tập sự kiện cơ sở và sự kiện dẫn xuất

Thủ tục này gồm hai bước:

- *Bước 1*: Tính mọi sự kiện là hệ quả logic của các quy tắc suy diễn để mở rộng tập sự kiện. Điều này được thực hiện nhờ việc áp dụng các luật suy diễn trên tập EDB cho đến khi không còn sự kiện mới nào được sinh ra. Gọi tập sự kiện mở rộng là EDB’.
- *Bước 2*: Tìm câu trả lời từ việc so khớp câu hỏi Q với tập sự kiện mở rộng EDB’ vì tập này chứa tất cả các thông tin được suy ra từ CSDL.

Thí dụ 4.11. Định giá câu hỏi Q trên CSDL “Gia tộc” theo cách tiếp cận dưới- lên:

Q: ? Bà (x, Mai)

Bước 1: Mọi thông tin được suy ra từ CSDL “Gia tộc” được tính bởi các phép lặp sau:

Lặp 0: Mọi sự kiện cơ sở đưa vào EDB’, có EDB’ = EDB. Chú ý rằng tập EDB’ luôn thay đổi sau mỗi lần lặp dưới đây.

Lặp 1: Áp dụng luật R1 cho EDB’, thu được các sự kiện mới, và thêm vào EDB’:

- Cha_mẹ (An, Dương)
- Cha_mẹ (An, Mai)

- Cha_mẹ (Giáp, An)
- Cha_mẹ (Giáp, Hồng)

Lặp 2: Áp dụng luật R2 cho EDB' (mới), thu được các sự kiện mới, và thêm vào EDB':

- Cha_mẹ (Xuân, An)
- Cha_mẹ (Xuân, Hồng)
- Cha_mẹ (Hồng, Giang)
- Cha_mẹ (Giang, Minh)

Lặp 3: Áp dụng luật R3 cho EDB' (mới), thu được các sự kiện mới, và thêm vào EDB':

- Bà (Xuân, Dương)
- Bà (Xuân, Giang)
- Bà (Xuân, Mai)
- Bà (Hồng, Minh)

Lặp 4: Áp dụng lại các luật R1, R2, R3 cho EDB' (mới), không có sự kiện mới nào được suy ra. Bước 1 kết thúc.

Như vậy sau bước 1, ta được EDB' gồm 20 sự kiện: 8 sự kiện cơ sở trong EDB và 12 sự kiện dẫn suất trong các lần lặp (b), (c), (d).

Bước 2: So khớp câu hỏi Q trên tập EDB' gồm tất cả các sự kiện cơ sở và dẫn xuất, đó là các thông tin có được từ CSDL. Vì Bà (Xuân, Mai) là một sự kiện trong tập này cho nên câu trả lời là x = 'Xuân'. Có nghĩa: Xuân là bà duy nhất của Mai được biết bởi CSDL "gia tộc".

- Nếu với câu hỏi : Q: ? Bà (Xuân, y) thì bước 1 cũng thực hiện như trên, ở bước 2, so khớp câu hỏi với EDB', ta sẽ có 3 sự kiện thỏa câu hỏi này:
 - Bà (Xuân, Dương)
 - Bà (Xuân, Giang)
 - Bà (Xuân, Mai)

Vì vậy tập các cháu y nhận Xuân là bà sẽ gồm: {Dương, Giang, Mai}

Nhận xét: Thủ tục định giá câu hỏi theo tiếp cận dưới-lên có ưu điểm chính là đơn giản, dễ thực hiện. Tuy nhiên cũng có nhiều nhược điểm:

- Nó suy ra nhiều hệ quả không liên quan đến câu hỏi. Chẳng hạn trong thí dụ trên, thủ tục suy dẫn ra nhiều sự kiện không cần thiết: Cha_mẹ (Giang, Minh), Cha_mẹ (Hồng, Giang), Cha_mẹ (Giáp, An), Bà (Xuân, Giang)...mà lẽ ra chỉ cần những sự kiện liên quan đến cha mẹ của Mai là đủ.
- Thứ tự thực hiện các luật có thể ảnh hưởng đến hiệu quả định giá câu hỏi, tức là chi phí tính toán phụ thuộc vào thứ tự áp dụng các luật. Chẳng hạn, trong thí dụ trên, nếu ta áp dụng luật R3 trước tiên (ở lần lặp 1) thì sẽ không suy ra được kết quả nào, sau đó sau khi thực hiện R1, lại phải áp dụng R3 một lần nữa. Rõ ràng là chi phí tính toán sẽ cao hơn thứ tự đã thực hiện trên đây.

4.4.2.2 Định giá câu hỏi theo tiếp cận trên – xuống (backward chaining)

Thủ tục này được mô tả như sau:

Cho trước câu hỏi Q, hãy xác định tập câu hỏi con Q_i sao cho câu trả lời cho Q chính là hợp các câu trả lời cho mỗi câu hỏi con Q_i . Để có được các câu hỏi con Q_i , mỗi tân từ dẫn xuất P trong Q phải được thay thế bởi thân các luật định nghĩa P. Theo cách đó, các câu hỏi con thu được sẽ “đơn giản” hơn câu hỏi ban đầu vì chúng được định nghĩa bởi các tân từ “gần hơn” với các tân từ cơ sở.

Việc thay các câu hỏi bằng các câu hỏi con được lặp lại nhiều lần, cho đến khi nhận được các câu hỏi chỉ chứa các tân từ cơ sở, khi đó chúng được so khớp với EDB và dễ dàng tìm được câu trả lời. Các hằng trong câu hỏi Q ban đầu được sử dụng trong quá trình tạo các câu hỏi con, vì chúng sẽ trở tới các tân từ cơ sở có liên quan đến câu hỏi ban đầu.

Thí dụ 4.12. Định giá câu hỏi Q trên CSDL_Gia tộc theo cách tiếp cận trên - xuống:

Q: ? Bà (x, Mai)

Thủ tục định giá câu hỏi theo cách tiếp cận trên -xuống sẽ thực hiện như sau:

Bước 1: Dùng luật R3: $Bà(x, y) \leftarrow Mẹ(x, z) \wedge Cha_mẹ(z, y)$, thay $y = 'Mai'$, câu hỏi Q được đưa về câu hỏi con:

Q1 : ? $Mẹ(x, z) \wedge Cha_mẹ(z, Mai)$

Bước 2: Thay tân từ $Cha_mẹ$ trong Q1 bởi các tân từ gần với các tân từ cơ sở hơn bằng cách dùng

các luật R1 và R2, ta được hai câu hỏi tương đương với Q1:

Q2a: ? $Mẹ(x, z) \wedge Bố(z, Mai)$

Q2b: ? $Mẹ(x, z) \wedge Mẹ(z, Mai)$

Bước 3: Xét các câu hỏi Q2a và Q2b.

- Với câu hỏi Q2a, so khớp với EDB, ta có sự kiện $Bố(An, Mai)$ là đúng, thay $z = 'An'$, câu hỏi này tương đương với câu hỏi:

Q3: ? $Mẹ(x, An)$

- Câu hỏi Q2b không có câu trả lời vì trong EDB không có sự kiện nào có thể so khớp được với $Mẹ(z, Mai)$.

Kết luận: Như vậy, câu hỏi Q ban đầu là tương đương với câu hỏi Q3. Câu hỏi này chỉ chứa tân từ cơ sở, được định giá trong EDB và cho câu trả lời là $x = \text{Xuân}$ (do có 1 sự kiện $Mẹ(Xuân, An)$).

Nhận xét: Thủ tục định giá câu hỏi theo tiếp cận trên-xuống có ưu điểm hơn cách tiếp cận dưới-lên vì quá trình định giá câu hỏi có tính tới các hằng có trong câu hỏi. Cách tiếp cận này không tính tất cả các hệ quả có thể có của CSDL, mà chỉ tính những hệ quả cần thiết cho quá trình định giá. Tuy nhiên thủ tục này cũng có những nhược điểm:

- Nó xét đến từng sự kiện cơ sở mỗi khi chúng xuất hiện trong các câu hỏi con nào đó: cách làm này kém hiệu quả vì phải quét CSDL nhiều lần.

- Thủ tục theo cách tiếp cận trên xuống có thể không kết thúc mà đi vào vòng lặp vô hạn, nếu ta lựa chọn các câu hỏi con không phù hợp. Chẳng hạn nếu có luật suy diễn:
 - $Tổ_tiên(x, y) \leftarrow Cha_mẹ(x, y)$
 - $Tổ_tiên(x, y) \leftarrow Cha_mẹ(x, z) \wedge Tổ_tiên(z, y)$

Giả sử việc tính toán bắt đầu với luật đệ quy: chuyển một câu hỏi về Tổ_tiên (đầu luật) lại về câu hỏi về Tổ_tiên trong thân luật ...đi vào vòng lặp không bao giờ kết thúc.

4.4.2.3 Định giá câu hỏi theo tiếp cận Ma tập (Magic sets)

Tiếp cận ma tập là tổ hợp của hai cách tiếp cận trên, nhằm khai thác những ưu việt của mỗi phương pháp.

Ý tưởng của phương pháp ma tập: cho một CSDL suy diễn D và câu hỏi Q trên một tân từ dẫn xuất P, Ta sẽ viết lại các luật của D, trong đó các luật viết lại có tính đến thông tin trong câu hỏi Q, ta nhận được CSDL tương đương D', sau đó áp dụng cách tiếp cận dưới-lên trong CSDL D'. Vì D' chứa các luật viết lại, cho nên các sự kiện mới tính được chỉ liên quan đến câu hỏi Q. Mục đích của sự viết lại các luật là đưa sự mô phỏng của tiếp cận trên-xuống vào các luật, làm cho việc định giá câu hỏi theo phương pháp dưới-lên đỡ tốn công sức tính các sự kiện suy dẫn.

Như vậy, thủ tục định giá câu hỏi bằng ma tập sẽ gồm 2 bước:

Bước 1: Xây dựng CSDL D' bằng cách viết lại các luật trong D, bằng cách đưa thông tin của Q vào thân các luật viết lại. Thông tin trong câu hỏi Q được biểu thị bằng một tân từ “ma ảo”, được coi như một tân từ ngoại diên và được gắn vào thân các luật viết lại.

Bước 2: Áp dụng thủ tục dưới-lên cho CSDL D' để tìm câu trả lời. Chú ý rằng do CSDL D' chứa các luật viết lại, cho nên chỉ suy ra các sự kiện liên quan đến câu hỏi Q. Do D là tương đương D', kết quả việc định giá câu hỏi Q trên D' là tương đương với định giá Q trên CSDL ban đầu.

Thí dụ 4.13.

Hãy định giá câu hỏi Q trên CSDL “Gia tộc” theo cách tiếp cận ma tập:

Q: ? Tổ_tiên (Hồng, x)

Bước 1: Với CSDL D (CSDL_Gia tộc), xây dựng CSDL D' bằng cách sửa các luật:

Thêm vào tân từ ma ảo: Magic_Toti (x) với sự kiện Magic_Toti (Hồng). Tân từ này có thể hiểu đơn giản: x là dòng họ của Hồng, (đương nhiên Hồng là dòng họ của Hồng!).

Các luật về “Tổ_tiên” R4 à R5 được viết lại như sau:

- $Tổ_tiên(x, y) \leftarrow Magic_Toti(x) \wedge Cha_mẹ(x, y)$ (Luật R6)
- $Magic_Toti(z) \leftarrow Magic_Toti(x) \wedge Cha_mẹ(x, z)$ (Luật R7)
- $Tổ_tiên(x, y) \leftarrow Magic_Toti(x) \wedge Cha_mẹ(x, z) \wedge Tổ_tiên(z, y)$ (Luật R8)

Ta nhận được CSDL D' như sau :

CSDL_D'

EDB : Các sự kiện cơ sở

- Bó (An, Dương)
- Bó (An, Mai)
- Bó (Giáp, An)
- Bó (Giáp, Hồng)
- Mẹ (Xuân, An)
- Mẹ (Xuân, Hồng)
- Mẹ (Hồng, Giang)
- Mẹ (Giang, Minh)

IDB : Các luật suy diễn:

- Cha_mẹ (x, y) $\leftarrow$ Bó (x, y) (Luật R1)
- Cha_mẹ (x, y) $\leftarrow$ Mẹ (x, y) (Luật R2)
- Bà (x, y) $\leftarrow$ Mẹ (x, z) $\wedge$ Cha_mẹ (z, y) (Luật R3)
- Tổ_tiên (x, y) $\leftarrow$ Cha_mẹ (x, y) (Xóa Luật R4)
- Tổ_tiên (x, y) $\leftarrow$ Cha_mẹ (x, z) $\wedge$ Tổ_tiên (z, y) (Xóa Luật R5)
- Tổ_tiên (x, y) $\leftarrow$ Magic_Toti (x) $\wedge$ Cha_mẹ (x, y) (Luật R6)
- Magic_Toti (z) $\leftarrow$ Magic_Toti (x) $\wedge$ Cha_mẹ (x, z) (Luật R7)
- Tổ_tiên (x, y) $\leftarrow$ Magic_Toti (x) $\wedge$ Cha_mẹ (x, z) $\wedge$ Tổ_tiên (z, y) (Luật R8)

Bước 2: Định giá câu hỏi Q theo tiếp cận dưới lên trong CSDL D'

Giả sử các sự kiện về cha mẹ đã được tính trong CSDL “Gia tộc” nhờ các luật R1, R2, trong đó nói riêng, ta đã có:

- Cha_mẹ (Hồng, Giang)
- Cha_mẹ (Giang, Minh)

Các sự kiện mới của D' được suy ra nhờ thực hiện các lần lặp :

Lặp 1: áp dụng luật R6, suy dẫn ra : Tổ_tiên (Hồng, Giang).

Lặp 2: áp dụng luật R7, suy dẫn ra : Magic_Toti (Giang).

Lặp 3: áp dụng luật R8: Không có sự kiện mới nào được suy ra.

Lặp 4: áp dụng luật R6, suy dẫn ra : Tổ_tiên (Giang, Minh).

Lặp 5: áp dụng luật R7, suy dẫn ra : Magic_Toti (Minh).

Lặp 6: áp dụng luật R8 suy dẫn ra : Tổ_tiên (Hồng, Minh).

Lặp 7: Không có sự kiện mới nào được suy ra nhờ áp dụng các luật R6, R7, R8.

Kết quả thu được là: {Tổ_tiên (Hồng, Giang), Tổ_tiên (Hồng, Minh)} là hai sự kiện so khớp được với câu hỏi. Vậy Hồng là tổ tiên của Giang và Minh.

Vậy bằng cách tính toán trên các luật viết lại, rõ ràng các sự kiện được suy dẫn nhờ các luật trong D' không nhiều, chỉ gồm những sự kiện liên quan đến câu hỏi Q . Điều này được thực hiện nhờ tân từ $\text{Magic_Toti}(x)$ được đưa vào thân các luật và bởi sự kiện $\text{Magic_Toti}(\text{Hong})$, cho phép ta chỉ tính đến các con cháu trong “dòng họ của Hồng”.

BÀI TẬP CHƯƠNG 4

1. Hãy trình bày những thành phần chính của một hệ Cơ sở dữ liệu suy diễn. Mục đích và yêu cầu của một hệ CSDL suy diễn?

2. Thế nào là một công thức đóng?, thế nào là một công thức dạng câu? Hãy chuyển các công thức đóng dưới đây thành công thức dạng câu:

$$1. \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z))$$

$$2. \forall x \forall y (P(x, y) \rightarrow \exists z Q(x, z)), \text{ với } z_1, z_2, \dots, z_n \text{ là giá trị của biến } z \text{ trong miền thể hiện } D$$

$$3. \forall x \forall z \exists y (P(x, y) \rightarrow Q(x, z)), \text{ với } y_1, y_2, \dots, y_m \text{ là giá trị của biến } y \text{ trong miền thể hiện } D$$

3. Phân biệt tân từ cơ sở và tân từ dẫn xuất, sự kiện cơ sở và sự kiện dẫn xuất trong CSDL suy diễn. Với các dữ kiện sau đây của một CSDL suy diễn, hãy xác định các tân từ cơ sở, tân từ dẫn xuất, sự kiện cơ sở và sự kiện dẫn xuất:

$$P(a, b), P(b, c), P(b, d), P(x, y), Q(x, y),$$

$$Q(x, z) \leftarrow P(x, y) \wedge P(y, z)$$

với x, y, z là các biến, a, b, c, d là các hằng.

4. Hãy xây dựng các tân từ cơ sở và xác định các sự kiện cơ sở ứng với các quan hệ sau đây :

R1	A	B	C
	1	3	6
	1	3	5
	2	4	6

R2	C	D
	5	7
	6	8

5. Chứng minh rằng quy tắc Modus Ponens là một trường hợp riêng của qui tắc Robinson.

6. Cho các tiên đề:
- A1: $P(a, b)$,
 - A2: $P(b, c)$, (với a, b, c : các hằng)
 - A3: $Q(x, y) \leftarrow P(x, y)$,
 - A4: $Q(x, y) \leftarrow P(x, z) \wedge Q(z, y)$, (với x, y, z : các biến)

Dùng quy tắc Robinson, hãy chứng minh định lý $Q(a, c)$ (theo phép chứng minh bằng hợp giải).

7. Trình bày các phương pháp định giá câu hỏi theo các cách: “Dưới lên”, “Trên xuống” và “Ma tập”. Hãy so sánh ưu nhược điểm của các phương pháp này.

8. Cho các sự kiện (tập EDB) :

- bo (an, duong) (được hiểu là sự kiện : an là bố của duong), tương tự với các sự kiện:
- bo (an, mai).
- bo (giap, an).
- bo (giap, hong).
- me (xuan, an). (được hiểu là sự kiện : xuan là mẹ của an), tương tự với các sự kiện :
- me (xuan, hong).
- me (hong, giang).
- me (giang, minh).

a/. Hãy viết các tân từ cơ sở của các sự kiện trên.

b/. Hãy viết các luật định nghĩa tân từ chame(X,Y) (ngữ nghĩa: X là cha mẹ của Y), ông_ngoại(X, Y) (ngữ nghĩa: X là ông ngoại của Y), các luật này làm thành tập IDB của CSDL ở bài tập này.

c/. Với CSDL đã bổ xung các luật ở phần b/ , hãy định giá câu hỏi Q: ?ông_ngoại(X, giang) bằng phương pháp dưới-lên; trên-xuống và ma tập.

9. Bài tập trắc nghiệm:

Trong các câu hỏi dưới đây, hãy chọn chỉ một phương án trả lời đúng nhất (tô đầy hình tròn ứng với mỗi phương án đã chọn):

9.1 Trong Datalog, phát biểu nào dưới đây là đúng nhất:

- ☐ (A) Một tần từ cơ sở là một công thức trong logic tần từ cấp 1.
- ☐ (B) Một sự kiện cơ sở là một công thức trong logic tần từ cấp 1.
- ☐ (C) Tất cả các câu trả lời A và B đều sai.
- ☐ (D) Tất cả các câu trả lời A và B đều đúng.

9.2 Trong Datalog, phát biểu nào dưới đây là đúng nhất:

- ☐ (A) $\forall x \forall y P(x, y)$ là một công thức đóng.
- ☐ (B) $P(a, b)$ là một công thức đóng.
- ☐ (C) Tất cả các câu trả lời A và B đều đúng.
- ☐ (D) Tất cả các câu trả lời A và B đều sai.

9.3 Giả sử có một luật (đúng) : 'Nếu trời mưa thì đường ướt'. Sử dụng quy tắc suy diễn Modus ponens

$\frac{P \rightarrow Q, P}{Q}$ hoặc Modus tollens $\frac{P \rightarrow Q, \neg Q}{\neg P}$, các kết luận nào trong các trường hợp dưới đây là đúng:

- ☐ (A) Sự kiện quan sát được: 'đường không ướt'. Kết luận: 'trời không mưa'.
- ☐ (B) Sự kiện quan sát được: 'đường ướt'. Kết luận: 'trời mưa'.
- ☐ (C) Tất cả các câu trả lời A và B đều sai.
- ☐ (D) Tất cả các câu trả lời A và B đều đúng.

9.4 Giả sử có một luật (đúng) : 'Nếu xe của tôi hết xăng thì xe của tôi không chạy'. Sử dụng quy tắc suy

diễn Modus ponens $\frac{P \rightarrow Q, P}{Q}$ hoặc Modus tollens $\frac{P \rightarrow Q, \neg Q}{\neg P}$, các kết luận nào trong các trường

hợp dưới đây là đúng:

- ☐ (A) Sự kiện quan sát được: 'Xe của tôi không chạy'. Kết luận: 'xe của tôi hết xăng'.
- ☐ (B) Sự kiện quan sát được: 'Xe của tôi hết xăng'. Kết luận: 'xe của tôi không chạy'.
- ☐ (C) Tất cả các câu trả lời A và B đều sai.
- ☐ (D) Tất cả các câu trả lời A và B đều đúng.

9.5 Giả sử có một luật (đúng) : 'Nếu xe của tôi hết xăng thì xe của tôi không chạy'. Sử dụng quy tắc suy

diễn Modus ponens $\frac{P \rightarrow Q, P}{Q}$ hoặc Modus tollens $\frac{P \rightarrow Q, \neg Q}{\neg P}$, các kết luận nào trong các trường

hợp dưới đây là đúng:

- ☐ (A) Sự kiện quan sát được: 'Xe của tôi không chạy'. Kết luận: 'xe của tôi hết xăng'.
- ☐ (B) Sự kiện quan sát được: 'Xe của tôi không hết xăng'. Kết luận: 'xe của tôi đang chạy'.
- ☐ (C) Tất cả các câu trả lời A và B đều sai.
- ☐ (D) Tất cả các câu trả lời A và B đều đúng.

9.6 Việc định giá câu hỏi trong CSDL suy diễn bằng phương pháp 'Trên - Xuống' (Back chaining) cần phải :

- ☐ (A) Tính tất cả các sự kiện dẫn xuất.
- ☐ (B) Thay đổi các luật suy diễn.
- ☐ (C) Thay câu hỏi ban đầu bằng câu hỏi tương đương có thể so sánh với các sự kiện cơ sở.
- ☐ (D) Tất cả các câu trả lời trên đều sai.

9.7 Việc định giá câu hỏi trong CSDL suy diễn bằng phương pháp Ma tập (Magic set) cần phải :

- ☐ (A) Tính tất cả các sự kiện dẫn xuất.
- ☐ (B) Thay đổi các luật suy diễn.
- ☐ (C) Thay đổi câu hỏi ban đầu bằng câu hỏi tương đương có thể so sánh với các sự kiện cơ sở.
- ☐ (D) Cả 3 phương án đều sai.

9.8 Cho tần từ cơ sở $P(x, y)$ và 2 sự kiện $P(a, b)$, $P(b, c)$, khi đó :

- ☐ (A) Suy dẫn được ra sự kiện $P(a, c)$.
- ☐ (B) Không suy dẫn được ra sự kiện $P(a, c)$.
- ☐ (C) Tất cả các câu trả lời A và B đều sai.
- ☐ (D) Tất cả các câu trả lời A và B đều đúng.