

Chương III. Các hệ CSDL suy diễn (Deductive Database Systems)

1. Logic và CSDL suy diễn
2. Ngôn ngữ luật Datalog (rule languages)
3. Ngữ nghĩa của chương trình Datalog
4. Đánh giá câu hỏi suy diễn

1

Mở đầu

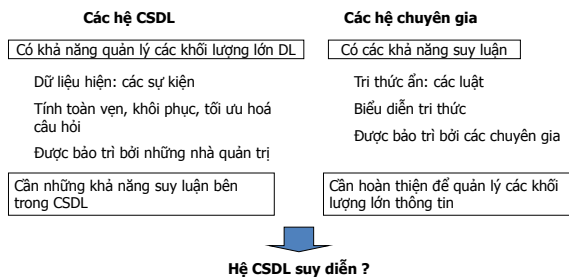
- Các hệ CSDL cổ điển không có khả năng suy diễn ra các sự kiện mới, các thông tin tiềm ẩn trong CSDL không được khai thác hết → lợi ích mang lại rất hạn chế
- 1970-1980: một trào lưu sôi nổi muốn đưa vào CSDL quan hệ các tri thức tổng quát và một cơ chế suy diễn với mục đích suy diễn ra nhiều thông tin và sự kiện mới.



Sự gặp gỡ của AI và CSDL

2

Mở đầu



3

Mở đầu

- Hệ CSDL suy diễn: - bổ sung các khả năng suy diễn cho các hệ CSDL trong đó CSDL suy diễn chứa các sự kiện (extensional relations-facts) và các luật để suy diễn ra các sự kiện mới (intensional relations)

CSDL là cơ sở tri thức

→ mở rộng việc truy vấn

- Datalog là một công cụ của DDB
 - tương tự với Prolog
 - có các sự kiện và các luật
 - cho phép định nghĩa các luật (views) – có khả năng đệ qui

4

Nội dung

- **Logic và CSDL suy diễn**
 - Logic vị từ cấp một
 - CSDL suy diễn
 - CSDL cài đặt và CSDL tiềm ẩn
 - Vấn đề thông tin ẩn
 - Hệ quản trị CSDL suy diễn
- **Ngôn ngữ luật Datalog đối với các CSDL suy diễn**
 - Cú pháp của Datalog
 - Chương trình Datalog
 - So sánh Datalog và ĐSQH
- **Ngữ nghĩa của Chương trình Datalog**
 - Thủ tục tính mô hình nhỏ nhất
 - Tính ngữ nghĩa của Chương trình Datalog sử dụng ĐSQH
- **Đánh giá câu hỏi suy diễn**
 - Đánh giá dưới – lên
 - Đánh giá trên – xuống

5

Logic vị từ cấp một

- Logic vị từ/tân từ cấp một (first-order logic - FOL) là một ngôn ngữ hình thức cho phép diễn tả các đối tượng và quan hệ giữa các đối tượng để suy diễn ra các quan hệ mới từ các quan hệ được xem là đúng
- Trong FOL, các công thức nguyên tố (atomic formulas) được diễn giải như các phát biểu về các **đối tượng** và **quan hệ giữa các đối tượng**.

6

Logic vị từ cấp một

• Vị từ và hằng:

Xét các phát biểu

« Mai là nữ

Bình là nam

Mai và Bình là vợ chồng”

Trong FOL, các phát biểu nguyên tố được biểu diễn bởi các vị từ, với các hằng như các đối số:

Nữ(mai)

Nam(bình)

Vợ_chồng(mai, bình)

Kí hiệu hằng: a, b, c,...

Kí hiệu vị từ: P, Q,...

7

Logic vị từ cấp một

• Biến và lượng từ:

Xét các phát biểu

« Mọi người là nữ hoặc là nam”

« Một người là nam thì sẽ không là nữ”

Trong FOL, các vị từ có thể xem các biến như các đối số, mà giá trị của nó được giới hạn bởi các lượng từ

Nam(x)

$\forall x \text{ Nữ}(x) \vee$

$\forall x \text{ Nữ}(x) \rightarrow \neg \text{Nam}(x)$

Kí hiệu biến : x, y, z,...

Kí hiệu lượng từ: \forall, \exists

8

Logic vị từ cấp một

• Hàm

Xét các phát biểu

« Bố của một người là nam »

Trong FOL, các đối tượng của miền có thể được biểu thị bởi các hàm áp dụng lên các đối tượng (đối tượng khác)

$\forall x \text{ Nam}(\text{Bố}(x))$

Kí hiệu hàm : $f(x_1, \dots, x_n)$, g , h, \dots

9

Logic vị từ cấp một

• Kí hiệu:

Bảng chữ:

- hằng: a, b, c, \dots
- biến: x, y, z, \dots
- hàm: $f(x_1, \dots, x_n)$, g, h, \dots
- các ký tự vị từ: P, Q, \dots
- các liên kết logic: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
- các lượng từ: \forall, \exists
- dấu ngoặc

10

Logic vị từ cấp một

• Hạng thức (term):

(i) Mọi biến và hằng

(ii) Nếu f là hàm n -ngôi và t_1, \dots, t_n là các hạng thức thì $f(t_1, \dots, t_n)$ cũng là một hạng thức

• Công thức nguyên tố

$P(t_1, \dots, t_n)$

11

Logic vị từ cấp một

• Công thức logic:

- Công thức được thiết lập đúng đắn (wff – well-formed formula):
 - (i) công thức nguyên tố
 - (ii) áp dụng các liên kết logic đối với các wff (iii) tác động lượng từ vào 1 wff
- literal là một công thức nguyên tố hay phủ định của công thức nguyên tố
- công thức dương (không có \neg)

12

Logic vị từ cấp một

- Biến tự do và biến giới hạn:
 $\forall x (R(y, z) \wedge \exists y (\neg P(y, x) \vee R(y, z)))$
 Các biến tô màu đỏ gọi là **biến tự do**, các biến còn lại là biến giới hạn
- Công thức đóng và mở
 Một công thức là đóng nếu nó không chứa biến tự do.
 Khi xây dựng các lý thuyết, người ta chỉ sử dụng các công thức đóng

13

Logic vị từ cấp một

- Thể hiện/điển giải (Interpretation)
 Cho D là một miền khác rỗng
 - gán mỗi hằng một phần tử thuộc D
 - gán cho mỗi hàm f n ngôi một ánh xạ $D^n \rightarrow D$
 - gán cho mỗi vị từ P n ngôi một ánh xạ $D^n \rightarrow \{0, 1\}$
- Mô hình
 Cho W là tập hợp wff. Một mô hình là một thể hiện sao cho mọi công thức thuộc W đều nhận giá trị đúng.

14

Ngữ nghĩa của tập luật

Hai cách diễn giải tập luật:

- Diễn giải theo lý thuyết chứng minh
- Diễn giải theo lý thuyết mô hình

15

Diễn giải theo lý thuyết chứng minh

Lý thuyết chứng minh :

Tất cả các sự kiện (facts) suy diễn được thông qua các luật là suy được bằng modus ponens
 $\text{person}(0111, \text{'Albert'}, 44, \text{salary})$
 $\text{person}(x, y, z, 45) \text{ :- } \text{person}(x, y, z, w) \ \& \ z \geq 35$
 $\text{person}(0111, \text{'Albert'}, 44, 45)$

Ngữ nghĩa của tập luật theo lý thuyết chứng minh là tập các sự kiện được suy từ các sự kiện cho trước hay có trong CSDL, dùng các luật theo hướng “suy diễn tiến”, nghĩa là chiều suy diễn từ các điều kiện/dịch con của luật dẫn đến kết luận/dịch của luật.

16

Diễn giải luật theo lý thuyết mô hình

- Các luật được định nghĩa theo thể giới khả dĩ hay mô hình
- Diễn giải: gán giá trị chân lý đúng hay sai cho các thể hiện khả dĩ của vị từ
- Mô hình của tập luật là một diễn giải làm cho các luật đúng từ phép gán các trị trong miền trị cho các biến trong từng luật

17

Ví dụ:

Cho p, q và r là các vị từ sau:

- $p(x) :- q(x)$ (1)
- $q(x) :- r(x)$ (2)

Miền trị: số nguyên

Các diễn giải:

- $M1 = \{r(1), q(1), p(1), q(2), p(2), p(3)\}$
- $M2 = \{r(1), q(1), p(1)\}$
- $M3 = \{r(1), q(2), p(2)\}$

Lưu ý : $p \rightarrow q = \sim p \vee q$, chỉ sai khi p đúng và q sai

18

Mô hình nhỏ nhất

- Gọi M_1, \dots, M_n là các mô hình của tập các công thức wff.
- Mô hình nhỏ nhất của S là mô hình M_k sao cho:
 - $M_k \subseteq M_j, j \in \{1, \dots, k-1, k+1, \dots, n\}$
 - $\nexists M, M$ là mô hình của S và $M \subseteq M_k$

19

Các khái niệm tương tự trong FOL và DBS

| DBS | FOL |
|-----------------------|------------------------------------|
| Quan hệ | Vị từ |
| Thuộc tính | Đối của vị từ |
| Bộ | Mệnh đề nền |
| View | Luật |
| Truy vấn | Mục tiêu (goal) |
| Ràng buộc(constraint) | Mục tiêu(sai nếu không nhất quán) |

20

CSDL suy diễn

Định nghĩa : CSDLSD bao gồm:

- (1) Một tập các sự kiện sơ cấp và các luật suy diễn
- (2) Một tập các ràng buộc toàn vẹn IC, là những công thức đóng bất kỳ

21

Dạng luật suy diễn

- Luật suy diễn có dạng tổng quát:

$$P_1 \wedge \dots \wedge P_k \rightarrow R_1 \vee \dots \vee R_m$$

- $R_1 \vee \dots \vee R_m$ là phần đầu hay kết luận của luật.
- Các R_i là đích của luật.
- $P_1 \wedge \dots \wedge P_k$ là phần thân của luật.
- Các P_j là đích con hay tiền đề của luật.

22

Vị từ được suy diễn và vị từ nền

- **Vị từ được suy diễn/tiềm ẩn IDB** (intensional predicate): là vị từ xuất hiện trong phần kết luận của luật. Vị từ được suy diễn cũng có thể xuất hiện trong phần thân của luật.
- **Vị từ nền EDB** (extensional predicate): ứng với một quan hệ được lưu trữ trong CSDL.
- Một vị từ có thể nhận giá trị đúng hay sai. Nếu p là vị từ nền và P là quan hệ nền thì $p(a, b, c)$ với a, b, c là đối số sẽ có giá trị đúng nếu bộ (a, b, c) sẽ tạo được trong tiến trình suy diễn.

23

Các dạng luật/câu

- $\rightarrow R(c_1, \dots, c_m)$: biểu diễn **sự kiện nền** (khẳng định) trong CSDL ứng với vị từ nền R
- $P(c_1, \dots, c_m) \rightarrow$: biểu diễn **sự kiện âm**
- $P_1 \wedge \dots \wedge P_k \rightarrow$: biểu diễn các ràng buộc toàn vẹn
Ví dụ: $\text{Cha}(x, y) \wedge \text{Mẹ}(x, y) \rightarrow$: *Không ai có thể vừa là cha vừa là mẹ của một người khác*
- $P_1 \wedge \dots \wedge P_k \rightarrow R_1 \vee \dots \vee R_m$: biểu diễn một ràng buộc toàn vẹn hoặc định nghĩa một vị từ được suy diễn/tiềm ẩn mới
Ví dụ: $\text{Chăm}(x, y) \rightarrow \text{Cha}(x, y) \vee \text{Mẹ}(x, y)$: *Mỗi người có nhiều nhất là Bố&Mẹ*
(Một câu được gọi là xác định nếu về phải chứa đúng một công thức nguyên tố)

- Trong CSDL truyền thống, các câu được xử lí như các **ràng buộc toàn vẹn**
- Trong CSDL suy diễn một số câu trong số chúng được xử lí như các **luật suy diễn** để định nghĩa các vị từ mới

24

CSDL cài đặt và CSDL tiềm ẩn

- Trong CSDL suy diễn, chúng ta xác định 2 kiểu quan hệ:
 - (1) Các quan hệ cơ sở tương ứng với tập các vị từ nền/cơ sở hay tập các sự kiện tạo nên một cơ sở dữ liệu cài đặt (EDB) - được lưu trữ vật lý trong CSDL
 - (2) Các quan hệ tiềm ẩn tương ứng với tập các vị từ được suy diễn/tiềm ẩn tạo nên một cơ sở dữ liệu tiềm ẩn (IDB) – các quan hệ tạm thời này thường lưu giữ các kết quả trung gian
- Một CSDL suy diễn bao gồm một CSDL cài đặt (EDB) và một CSDL tiềm ẩn (IDB)

25

Các dạng truy vấn

- Biểu diễn truy vấn: $? p(x_1, \dots, x_m)$
 - x_i là hằng với $i = 1, \dots, m \rightarrow$ Truy vấn Y|N
 - một số đối số là hằng \rightarrow Tìm mối quan hệ đặc biệt giữa các đối số còn lại
 - x_i là biến với $i = 1, \dots, m \rightarrow$ Tìm mối quan hệ tổng quát giữa tất cả các đối số trong p

26

Vấn đề thông tin âm

« Ai là cha của Dũng, Hùng có là cha của Dũng không ? »

Giả sử có:

CHA(Hùng, Mạnh)

CHA(Sơn, Thuỳ)

không có khả năng trả lời khi tra cứu CSDL, nói cách khác

$\neg \text{CHA}(\text{Hùng}, \text{Dũng})$ không có trong CSDL

27

Vấn đề thông tin âm

Giả thiết thế giới đóng (Closed World Assumption - CWA),
hay qui ước với thông tin âm:

Sự kiện không xuất hiện trong CSDL là sai

Qui tắc « phủ định bởi thất bại »:

Thông tin âm mà không chứng minh được người ta coi nó là đúng

28

Hệ QTCSDL suy diễn

- là một hệ QTCSDL: có ngôn ngữ DDL, DML
- cung cấp một giao diện có ngôn ngữ luật: cho phép từ các quan hệ cơ sở được lưu trữ trong CSDL suy diễn ra các quan hệ mới thông qua các luật suy diễn

29

Hệ QTCSDL suy diễn

- Có khả năng thực hiện các phép toán quan hệ cổ điển
- Định qui: cho phép định nghĩa một quan hệ bởi chính nó
- Có khả năng chấp nhận phủ định (xử lý thông tin âm) để dẫn trở đến các sự kiện không tồn tại
- Cập nhật các sự kiện thông qua các luật

30

Hệ QTCSDL suy diễn

- Có nhiều cách tiếp cận để xây dựng:
 - mở rộng, phát triển: xuất phát từ một hệ QTCSDL quan hệ
 - mở rộng, phát triển: xuất phát từ một ngôn ngữ luật và một mô hình suy diễn
- Cần phải đưa vào đó một cơ chế suy diễn đòi hỏi:
 - Ngôn ngữ diễn đạt các luật
 - Mô hình bên trong để lưu trữ các sự kiện và các luật
 - Tính nhất quán

31

Nội dung

- **Logic và CSDL suy diễn**
 - Logic vị từ cấp một
 - CSDL suy diễn
 - CSDL cài đặt và CSDL tiềm ẩn
 - Vấn đề thông tin âm
 - Hệ quản trị CSDL suy diễn
- **Ngôn ngữ luật Datalog đối với các CSDL suy diễn**
 - cú pháp của Datalog
 - Chương trình Datalog
 - So sánh Datalog và ĐSQH
- **Ngữ nghĩa của Chương trình Datalog**
 - Thủ tục tính mô hình nhỏ nhất
 - Tính ngữ nghĩa của Chương trình Datalog sử dụng ĐSQH
- **Đánh giá câu hỏi suy diễn**
 - Đánh giá dưới – lên
 - Đánh giá trên – xuống

32

Ngôn ngữ luật Datalog đối với các CSDL suy diễn

Một ngôn ngữ luật phải là

- tập hợp và phi thủ tục
- dễ giao diện với một hệ QTCSDL quan hệ (dễ dàng được dịch thành một chương trình của đại số quan hệ mở rộng)

33

Datalog

- Là một ngôn ngữ luật cho các CSDL, cho phép **xác định** các quan hệ suy diễn/tiềm ẩn nhờ phép **kéo theo** đơn giản **không có kí hiệu hàm**
- Có thể xem đó là một biến dạng của **Prolog** với một ngữ nghĩa tập hợp (kết quả của một chương trình không phụ thuộc vào thứ tự các câu/luật)

34

Datalog

Các mở rộng:

- Đưa thêm kí hiệu hàm vào trong đối số của các vị từ $\text{DATALOG}^{\text{func}}$
- Thêm phép phủ định $\text{DATALOG}^{\text{neg}}$
- Cập nhật tường minh trong một luật $\text{DATALOG}^{\text{maj}}$
- Đưa vào những điều kiện tổng quát chứa các phép hội, tuyển, phủ định (phi Horn) $\text{DATALOG}^{\text{non}}$

35

Cú pháp của Datalog

Bảng chữ được dùng gồm các kí hiệu:

- Hằng: a, b, c, \dots
- Biến: x, y, z, \dots
- các vị từ quan hệ: P, Q, R, \dots
- các vị từ so sánh: $=, <, >, \dots$
- các liên kết logic: $;(\vee), ;(\wedge), \neg, \rightarrow, \Leftrightarrow$

Hạng thức: hằng hay biến

Công thức nguyên tố: Literal dương $P(t_1, t_2, \dots, t_n)$

Công thức nguyên tố **cá biệt** (được làm cá biệt): công thức nguyên tố không chứa biến

36

Cú pháp của Datalog

Luật: biểu thức có dạng $Q \leftarrow P_1, P_2, \dots, P_n$

- đầu luật Q là công thức nguyên tố (kết luận-đích)
- thân luật P_1, P_2, \dots, P_n là các vị từ (tiền đề hay điều kiện)
- mỗi P_i gọi là một đích con
- Luật được gọi là đệ qui nếu vị từ của đầu luật cũng xuất hiện trong thân luật

37

Ví dụ về các luật Datalog

- hằng: Hùng, Dũng, Mai, Thanh, ..
- các literal cá biệt còn gọi là các sự kiện:
 $BO(\text{Hùng}, \text{Dũng}) \leftarrow$
 $ME(\text{Mai}, \text{Dũng}) \leftarrow$

- Các luật khác:
 $CHAME(x,y) \leftarrow BO(x,y)$
 $CHAME(x,y) \leftarrow ME(x,y)$
 $ONG(x,z) \leftarrow BO(x,y), CHAME(y,z)$
 $TOTIEN(x,y) \leftarrow CHAME(x,y)$
 $TOTIEN(x,z) \leftarrow TOTIEN(x,y), CHAME(y,z)$
 $ANHEMHO(x,y) \leftarrow TOTIEN(z,x), TOTIEN(z,y)$

38

Chương trình Datalog

Một chương trình Datalog là một tập các luật (thứ tự các luật không quan trọng)

Datalog cho phép người dùng phản ánh các luật và các sự kiện

CSDL Datalog = CSDL cài đặt + CSDL tiềm ẩn
(viết bằng Datalog)

39

So sánh DATALOG và ĐSQH

- DATALOG có sức mạnh của ĐSQH với sự cho phép **đệ qui** (ĐSQH không cho phép đệ qui)
- Phép hợp: 1 số luật cùng đầu
- Phép chiếu: một luật có một số biến ở phần thân bị lấy đi khỏi phần đầu của luật
- Phép chọn: một luật có ít nhất một vị từ so sánh trong phần thân
- Phép kết nối: luật gồm một số vị từ quan hệ ở phần thân có đối chung

40

Mở rộng DATALOG với các hàm

Nhờ các hàm có thể tính toán, xử lý trên những đối tượng phức tạp (hình vẽ, kiểu dữ liệu trừu tượng)

Do vậy đưa vào DATALOG các hàm của logic vị từ cấp một:

- $+$, $-$, \times , $/$
- LOG, EXP,...
- hàm định nghĩa bởi người dùng

Đưa vào các kí hiệu hàm f , g ,... có một số cố định đối số

Các đối của vị từ có thể là hằng hoặc biến hoặc là **hàm** tác động lên các hạng thức $f(t_1, t_2, \dots, t_n)$

41

Mở rộng DATALOG với các hàm (tiếp)

Ví dụ:

Có bản đồ đường đi nối các thành phố (đồ thị có hướng)

$\{\text{Đường_đi}(x, y, d) \leftarrow \text{Cung}(x, y, d);$
 $\text{Đường_đi}(x, y, e+d) \leftarrow \text{Đường_đi}(x, z, e), \text{Cung}(z, y, d)\}$

? Đường_đi (Hà Nội, Sài Gòn, t)

42

Mở rộng DATALOG với các hàm (tiếp)

Ví dụ:

- $\{\text{lương}(100) \leftarrow ; \text{cao_hơn}(y, x) \leftarrow \text{lương}(x), x < y\}$

? $\text{cao_hơn}(y, x)$ cho vô số đáp số

- $\{\text{nguyên}(0) \leftarrow ; \text{nguyên}(x+1) \leftarrow \text{Nguyên}(x)\}$

chương trình này phát sinh tất cả các số nguyên dương

Một luật gọi là có trường hạn chế nếu tất cả các biến trong đầu luật đều xuất hiện trong một vị từ quan hệ ở thân luật.

Một chương trình gọi là an toàn nếu không phát sinh vô hạn đáp số

43

Mở rộng DATALOG với phép phủ định

Ví dụ : tính độ dài các đường đi trên đồ thị có hướng, có sử dụng quan hệ lưu trữ các cung bị cấm

$\{\text{Đường_đi}(x, y, d) \leftarrow \text{Cung}(x, y, d), \neg \text{Cấm}(x, y);$

$\text{Đường_đi}(x, y, e+d) \leftarrow \text{Đường_đi}(x, z, e), \text{Cung}(z, y, d), \neg \text{Cấm}(z, y)\}$

Giao của hai mô hình nói chung không là mô hình

44

Mở rộng DATALOG với phép phủ định (tiếp)

Ví dụ:

```
{chim (ngựa_có_cánh) ← ;
chim_cánh_cụt (x) ← chim (x), ¬bay(x);
bay(x) ← chim (x), ¬ chim_cánh_cụt (x)}
```

Chương trình này có 2 mô hình:

Mô hình 1: {chim (ngựa_có_cánh), chim_cánh_cụt (ngựa_có_cánh)}

Mô hình 2: {chim (ngựa_có_cánh), bay (ngựa_có_cánh)}

Giao của hai mô hình là {chim (ngựa_có_cánh)} không là mô hình

45

Nội dung

- **Logic và CSDL suy diễn**
 - Logic vị từ cấp một
 - CSDL suy diễn
 - CSDL cài đặt và CSDL tiềm ẩn
 - Vấn đề thông tin ẩn
 - Hệ quản trị CSDL suy diễn
- **Ngôn ngữ luật Datalog đối với các CSDL suy diễn**
 - Cú pháp của Datalog
 - Chương trình Datalog
 - So sánh Datalog và ĐSQH
- **Ngữ nghĩa của Chương trình Datalog**
 - Thủ tục tính mô hình nhỏ nhất
 - Tính ngữ nghĩa của Chương trình Datalog sử dụng ĐSQH
- **Đánh giá câu hỏi suy diễn**
 - Đánh giá dưới – lên
 - Đánh giá trên – xuống

46

Ngữ nghĩa của chương trình Datalog

- Ngữ nghĩa của một chương trình Datalog là tập các sự kiện chương trình đó tính được:

- Phương pháp thủ tục (từng bộ một)/ dựa trên phương pháp chứng minh bằng phương pháp giải và "phủ định bởi thất bại"
- Phương pháp khai báo/dựa trên việc tính mô hình của một chương trình logic

47

Mô hình của một chương trình Datalog

Một mô hình của một chương trình Datalog là một diễn giải/thể hiện thỏa mãn các tính chất sau:

- a) Với mọi bộ (a_1, \dots, a_n) của một quan hệ cơ sở B, $B(a_1, \dots, a_n)$ đúng trong thể hiện
- b) Với mọi luật $Q(t_1, t_2, \dots, t_n) \leftarrow P_1, P_2, \dots, P_n$ và với mọi phép gán θ trong thể hiện:
Nếu $\theta(P_1, P_2, \dots, P_n)$ đúng trong thể hiện
thì $\theta(Q(t_1, t_2, \dots, t_n))$ cũng đúng

48

Mô hình của một chương trình Datalog

Một mô hình của một chương trình Datalog là một tập các vị từ cá biệt:

- chứa tất cả các sự kiện của CSDL và
- chứa tất cả các sự kiện có thể được suy diễn bằng cách áp dụng các luật

49

Mô hình nhỏ nhất của một chương trình Datalog (Ngữ nghĩa *chính tắc* của chương trình Datalog)

- Giao của hai mô hình cũng là mô hình
- Giao của tất cả các mô hình: mô hình nhỏ nhất được gọi là ngữ nghĩa chính tắc
- Dùng Toán tử T_p (Van Edem 1976) để tính ngữ nghĩa chính tắc

50

Thủ tục tính mô hình nhỏ nhất

- Cho I là tập sự kiện
- Tính $T_p(I) = \{ Q / \exists P_1, P_2, \dots, P_n \in I$
sao cho $Q \leftarrow P_1, P_2, \dots, P_n$ là một luật cá biệt của $P \}$
- Bắt đầu với $I = \text{CSDL cài đặt}$, khi đó $T_p(\emptyset) = \text{CSDL cài đặt}$ -tập các sự kiện của chương trình

51

Thủ tục tính mô hình nhỏ nhất

Ví dụ

Cho chương trình DATALOG sau:

```
P = { BO(Hùng, Dũng) ← ;
      ME(Mai, Dũng) ← ;
      CHAME(x,y) ← BO(x,y);
      CHAME(x,y) ← ME(x,y);
      TOTIEN(x,y) ← CHAME(x,y) }
```

Tính:

bắt đầu

$T_p(\emptyset) = \{ \text{BO(Hùng, Dũng)} \leftarrow ; \text{ME(Mai, Dũng)} \leftarrow \}$

52

Thuật toán tính mô hình nhỏ nhất

Ví dụ

$T_p(\emptyset) = \{ BO(Hùng, Dũng) \leftarrow ; ME(Mai, Dũng) \leftarrow \}$

....

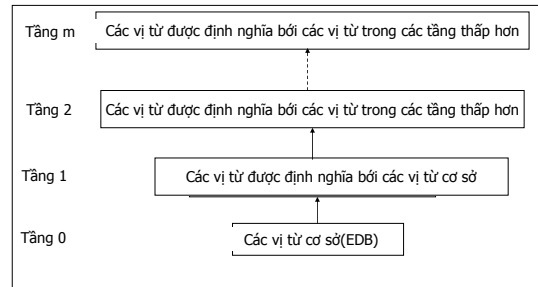
$T_p(I) = I'$ I' là mô hình nhỏ nhất của chương trình P

Với những chương trình lớn cần có những thuật toán tối ưu hơn.

53

Ngữ nghĩa của các chương trình Datalog không đệ quy

- Phân tầng các vị từ sử dụng sự phụ thuộc của các vị từ:



Tính quan hệ tiềm ẩn

- Đối với một luật r có p_i ở đầu luật, để tính quan hệ IDB tương ứng với p_i chỉ sử dụng luật r :
 - tính các quan hệ tương ứng với các vị từ trong phần thân của luật
 - thực hiện phép kết tự nhiên các quan hệ tương ứng với các đích con khác nhau
 - thực hiện phép chọn với điều kiện chọn tương ứng với các vị từ so sánh
 - thực hiện phép chiếu với tập thuộc tính chiếu tương ứng với các đối số của p_i

55

Tính quan hệ tiềm ẩn

- Tính quan hệ tiềm ẩn ứng với vị từ p nằm ở phần đầu của một số luật trong chương trình Datalog:
 - Xét các luật có p trong phần đầu, giả sử $Rules = \{r_1, r_2, \dots, r_r\}$
 - Tính quan hệ tiềm ẩn ứng với vị từ p sử dụng 1 luật r_i , $1 \leq i \leq r$
 - Hợp các quan hệ kết quả (các quan hệ cần khả hợp)
- Tính chính xác vị từ p trong Rules thành một vị từ $p(X_1, \dots, X_k)$ - tập đối số của các vị từ p là giống nhau, sao cho :
 - Sử dụng các biến mới X_i cho các vị từ trong phần đầu của luật
 - Các biến X_i là khác nhau

56

Tinh chỉnh luật

- Luật r có vị từ trong phần đầu là $p(Y_1, \dots, Y_k)$ tạo vị từ trong phần đầu với $p(X_1, \dots, X_k)$ với:
 - Các biến X_i là khác nhau
 - Thay thế các đối Y_i bởi X_i với các vị từ xuất hiện trong thân của luật
 - Thêm vị từ so sánh $X_i = Y_i$ vào thân luật với Y_i là hằng
 - Thêm vị từ so sánh $X_i = X_j$ vào thân luật với $Y_i = Y_j$
- Ví dụ:** Xét các luật
 - $p(a, X, Y) :- r(X, Y)$
 - $p(X, Y, X) :- r(Y, X)$
- Các luật được tinh chỉnh:
 - $p(U, V, W) :- r(U, W), U = a$
 - $p(U, V, W) = r(V, U), W = U$

57

Ví dụ:

- Luật: $\text{cousin}(X, Y) :- \text{parent}(X, X_p), \text{parent}(Y, Y_p), \text{sibling}(X_p, Y_p)$
 - Tính quan hệ C tương ứng với vị từ cousin:
 - Các quan hệ S, P tương ứng với các vị từ sibling, parent đã được tính
 - $C(X, Y) = P(X, X_p) * P(Y, Y_p) * S(X_p, Y_p)$
 - Bộ của C có dạng (a, b, c, d)
 - Tồn tại b, d với (a, b) thuộc P
 - (c, d) thuộc P và
 - (b, d) thuộc S

58

Ví dụ:

- Với luật: $\text{sibling}(X, Y) :- \text{parent}(X, Z), \text{parent}(Y, Z), X \neq Y$
 - $S(X, Y) = \Pi_{X, Y} \sigma_{X \neq Y} (P(X, Z) * P(Y, Z))$
 - Với luật: $p(X, Y) :- q(a, X), r(X, Z), s(Y, Z)$
 - Quan hệ đ/v $q(a, X)$: $\sigma_{S_1=a}(Q)$ với $Q(V, X)$
 - Quan hệ đ/v $r(X, Z)$: R
 - Quan hệ đ/v $s(Y, Z)$: S
 - $P(X, Y) = \Pi_{X, Y} (\sigma_{S_1=a}(Q(V, X)) * R(X, Z) * S(Y, Z))$
- (Với S_k là thuộc tính thứ k trong quan hệ)

59

Thuật toán 1

- Vào:** Một luật Datalog r với phần thân có chứa các đích con S_1, \dots, S_n và các biến X_1, \dots, X_n . Với mỗi $S_i = p_i(A_{i1}, \dots, A_{ik_i})$ là vị từ quan hệ sẽ có một quan hệ đã được tính R_i trong đó các đối A_{ij} là biến hoặc hằng.
- Ra:** Biểu thức đại số quan hệ, ký hiệu là $\text{EVAL_RULE}(r, R_1, \dots, R_n)$ cho phép tính một quan hệ $R(X_1, \dots, X_m)$ từ các quan hệ R_1, \dots, R_n có chứa các bộ (a_1, \dots, a_m) sao cho khi thay a_j vào $X_j, 1 \leq j \leq m$ tất cả đích con S_1, \dots, S_n đều đúng.

60

Phương pháp

Biểu thức được xây dựng qua các bước sau:

- 1. Đối với mỗi đích con S_i là vị từ quan hệ p_i , gọi Q_i là biểu thức $\Pi_{V_i}(\sigma_{F_i}(R_i))$. Với V_i là tập hợp các biến chỉ xuất hiện một lần trong các đối của S_i . Biểu thức F_i là phép \wedge của các điều kiện sau:
 - Nếu ở vị trí k của S_i có hằng a thì F_i có điều kiện $S_k = a$
 - Nếu ở vị trí k và l của S_i chứa các đối giống nhau thì F_i có điều kiện $S_k = S_l$
 - Đặc biệt nếu F_i không có một điều kiện nào, chẳng hạn khi $S_i = p_i(X, Y)$ thì xem F_i là hằng đúng, như thế $Q_i = R_i$.

61

Phương pháp

- 2. Đối với mỗi biến X không xuất hiện trong các đích con là vị từ quan hệ, tính biểu thức Dx nhằm tạo ra quan hệ một ngôi chứa tất cả các giá trị mà X có thể nhận trong phép gán làm thỏa tất cả đích con của luật r . Do luật r là an toàn nên X phải được gán bằng với biến Y có giới hạn nào đó qua một chuỗi các phép gán bằng "=" và Y được giới hạn nhờ một hằng a nào đó trong một đích con, hoặc Y là một đối của một đích con vị từ quan hệ.
 - Nếu $Y = a$ là một đích con, thì đặt Dx là biểu thức hằng $\{a\}$
 - Nếu Y xuất hiện như là đối thứ j của một đích con vị từ quan hệ S_j , thì đặt Dx là $\Pi_{S_j}(R_j)$

62

Phương pháp

- 3. Gọi E là nối tự nhiên của tất cả các Q_i được định nghĩa trong (1) và các Dx được định nghĩa trong (2). Trong phép nối này, ta xem Q_i là quan hệ với các thuộc tính là các biến trong S_i , và xem Dx là quan hệ có thuộc tính X
- 4. Gọi **EVAL_RULE**(r, R_1, \dots, R_n) là $\sigma_F(E)$ trong đó F là hội các biểu thức $X\theta Y$ tương ứng với các vị từ so sánh với X, Y xuất hiện trong số các đích con p_1, \dots, p_n và E là biểu thức được xây dựng trong bước (3). Nếu không có vị từ so sánh nào thì biểu thức cuối cùng chính là E .

63

Định lý 1

- Thuật toán 1 là đúng theo nghĩa quan hệ R được tạo ra có tất cả và chỉ những bộ (a_1, \dots, a_m) sao cho khi thay thế mỗi X_j bằng a_j , mỗi đích con S_i đều được làm đúng.
(Xem chứng minh trong JD Ullman, Vol1, chương 3)

64

Ví dụ:

Luật: $P(X,Z) :- q(a,X), r(X,Z,X), s(Y,Z)$

- Đích con: S1 là $q(a,X)$
 $Q1 = \Pi_{S2}(\sigma_{S1=a}(Q))$
- Đích con: S2 là $r(X,Z,X)$
 $Q2 = \Pi_{S1,S2}(\sigma_{S1=S2}(R)) = U(X,Z)$
- Đích con: S3 là $s(X,Z)$
 $Q3 = S(X,Z)$
- Không có vị từ so sánh (bước 4 trong thuật toán 1):
 $Q1 * Q2 * Q3$

65

Thuật toán 2

- **Vào:** Một chương trình Datalog không đệ qui và một quan hệ cơ sở Pi với mỗi vị từ EDB pi xuất hiện trong chương trình.
- **Ra:** Một biểu thức đại số quan hệ trên các quan hệ tương ứng với các vị từ EDB biểu diễn một quan hệ tiềm ẩn tương ứng với mỗi vị từ IDB

66

Phương pháp

1. Tinh chỉnh tập tất cả các luật của chương trình Datalog.
2. Phân tầng các vị từ trong chương trình Datalog không đệ qui sử dụng sự phụ thuộc của các vị từ
3. Tinh quan hệ tiềm ẩn ứng với các vị từ IDB từ tầng thấp đến tầng cao như sau: với $j=1, 2, \dots, m$, tạo biểu thức đại số cho quan hệ Pi ứng với vị từ IDB pi tại tầng j như sau:
 - Xét các luật có pi trong phần đầu, giả sử $Rules = \{r_1, r_2, \dots, r_r\}$
 - Đối với mỗi luật r_k có pi là phần đầu, sử dụng thuật toán 1 để tìm biểu thức E_{rk} tính quan hệ R_{rk} ứng với vị từ các quan hệ ứng với các vị từ tại các tầng thấp hơn xuất hiện trong thân của r_k
 - Thay các biểu thức thích hợp cho mỗi xuất hiện của một quan hệ IDB trong biểu thức E_{rk} để có được một biểu thức mới F_{rk} .
 - Đặt $Pi = \cup_k F_{rk}$.

67

Định lý 2

- Thuật toán 2 là đúng và cho phép tính chính xác quan hệ tiềm ẩn cho mỗi vị từ được suy diễn theo nghĩa biểu thức được xây dựng cho mỗi vị từ IDB sẽ tạo ra:
 - Tập các sự kiện (facts) cho vị từ đó mà có thể chứng minh từ CSDL
 - Tập tất cả các sự kiện tính được là mô hình nhỏ nhất duy nhất của chương trình Datalog

68

Ví dụ:

Cho các quan hệ EDB và IDB như sau:

- $p(a,Y) :- s(X,Y)$
- $p(X,Y) :- s(X,Z), r(Z,Y)$
- $q(X,X) :- p(X,b)$
- $q(X,Y) :- p(X,Z), s(Z,Y)$

Tính chỉnh luật:

- $p(X,Y) :- s(X,Y), X = a$
- $p(X,Y) :- s(X,Z), r(Z,Y)$
- $q(X,Y) :- p(X,Y), X=Y, Y = b$
- $q(X,Y) :- p(X,Z), s(Z,Y)$

69

Ví dụ:

Phân tầng các vị từ:

- Tầng 0: 2 vị từ EDB r và s
- Tầng 1: vị từ p
- Tầng 2: vị từ q

Tính quan hệ tiềm ẩn đ/v vị từ p tại tầng 1:

- Rules (p) = $\{ p(X,Y) :- s(X,Y), X = a; p(X,Y) :- s(X,Z), r(Z,Y) \}$
- $P(X,Y) :- \sigma_{X=a}(S(X,Y)) \cup \Pi_{X,Y}(S(X,Z) * R(Z,Y))$

Tính quan hệ tiềm ẩn đ/v vị từ q tại tầng 2:

- Rules (q) = $\{ q(X,Y) :- p(X,Y), X=Y, Y = b; q(X,Y) :- p(X,Z), s(Z,Y) \}$
- $Q(X,Y) = \sigma_{X=Y, Y=b}(P(X,Y)) \cup \Pi_{X,Y}(P(X,Z) * S(Z,Y))$

70

Ngữ nghĩa của các chương trình Datalog đệ qui

- Ngữ nghĩa của các chương trình đệ qui là không rõ ràng.

Ví dụ:

1. $T(a);$
2. $R(X) :- T(X), \text{not } S(X);$
3. $S(X) :- T(X), \text{not } R(X).$

→ $R(a)? S(a)?$

R phụ thuộc vào T và $\text{not } S$, S lại phụ thuộc vào T và $\text{not } R$

- Chương trình phải được phân tầng: Nếu T phụ thuộc vào $\text{not } S$, thì S không thể phụ thuộc vào T (hay $\text{not } T$).

71

Ngữ nghĩa của các chương trình Datalog đệ qui

- Các quan hệ trong chương trình có thể được phân thành các tầng:

– Tầng 0: Tất cả các quan hệ trong CSDL cài đặt.

– Tầng 1: Các quan hệ được định nghĩa theo các quan hệ trong tầng 1 hoặc tầng thấp hơn.

– Nếu T phụ thuộc vào $\text{not } S$, thì S phải thuộc tầng thấp hơn tầng của T .

72

Ngữ nghĩa của các chương trình Datalog đệ qui

- Ngữ nghĩa của chương trình phân tầng được xác định bởi điểm bất động của tập tất cả các luật trong chương trình.
- Bắt đầu với tập sự kiện I, thuật toán tính điểm bất động của tập luật R trong 1 tầng luôn kết thúc do tính đơn điệu của thuật toán và điểm bất động của các quan hệ là hữu hạn.
- Tính ngữ nghĩa như sau: Tính điểm bất động của các quan hệ trong từng tầng từ thấp đến cao bắt đầu với tập sự kiện trong các quan hệ EDB ở tầng 0.

73

Thuật toán 3

• Ý tưởng:

Áp dụng thuật toán 2 và gán cho các quan hệ IDB cần tính ban đầu là \emptyset , sau đó thêm dần các bộ mới vào các quan hệ IDB này sử dụng một tiến trình lặp cho đến khi đi đến một bước mà kết quả không thay đổi.

• Thuật toán 3:

- Vào: Một chương trình Datalog đệ qui với các vị từ EDB r_1, \dots, r_k và các vị từ IDB p_1, \dots, p_m và một danh sách các quan hệ R_1, \dots, R_k tương ứng với các vị từ EDB.
- Ra: Điểm bất động nhỏ nhất của chương trình Datalog.

74

Phương pháp

1. Tính chính tập tất cả các luật của chương trình Datalog.
2. Phân tầng các vị từ trong chương trình Datalog không đệ qui sử dụng quan hệ phụ thuộc vị từ. Giả sử, chương trình Datalog được phân thành m tầng và tập quan hệ IDB thuộc tầng j là T_j với $j=1, 2, \dots, m$.
3. Tính quan hệ tiềm ẩn ứng với các vị từ IDB từ tầng thấp đến tầng cao như sau: với $j=1, 2, \dots, m$, các quan hệ IDB P_i tại tầng j được tính như sau:
 - Xây dựng phương trình cho các P_i là $P_i = \text{EVAL}(P_i, R_1, \dots, R_k, T_1, \dots, T_j)$. Sau đó khởi gán giá trị rỗng cho từng P_i rồi lặp đi lặp lại việc tính EVAL để thêm dần các bộ mới cho các P_i .
 - Khi không còn bổ sung thêm được nữa, P_i thu được là điểm bất động.

75

Chi tiết thuật toán

For j:= 1 to m do

begin

For i:= 1 to T_j do $P_i := \emptyset$;

Repeat

For i:= 1 to T_j do $Q_i := P_i$; // Lưu giá trị cũ

For i:= 1 to T_j do $P_i := \text{EVAL}(P_i, R_1, \dots, R_k, T_1, \dots, T_j)$;

Until $P_i = Q_i$ với mọi $i=1, \dots, T_j$

end

76

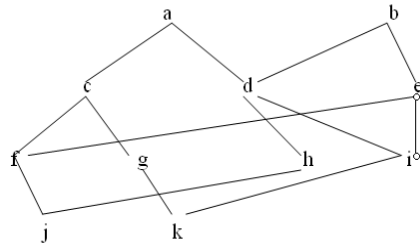
Ví dụ:

- Cho các luật đã được tinh chỉnh :
 1. $\text{sibling}(X,Y) :- \text{parent}(X,Z), \text{parent}(Y,Z), X \neq Y$
 2. $\text{cousin}(X,Y) :- \text{parent}(X,Xp), \text{parent}(Y,Yp), \text{sibling}(Xp,Yp)$
 3. $\text{cousin}(X,Y) :- \text{parent}(X,Xp), \text{parent}(Y,Yp), \text{cousin}(Xp,Yp)$
 4. $\text{related}(X,Y) :- \text{sibling}(X,Y)$
 5. $\text{related}(X,Y) :- \text{related}(X,Z), \text{parent}(Y,Z)$
 6. $\text{related}(X,Y) :- \text{related}(Z,Y), \text{parent}(X,Z)$

77

Ví dụ:

Các bộ của quan hệ Parent(X,Y) viết tắt P(X,Y) được định nghĩa:



78

Ví dụ:

Phân tầng các vị từ:

- Tầng 0: vị từ EDB parent, quan hệ tương ứng $P(X,Y)$
- Tầng 1: vị từ sibling, quan hệ tương ứng $S(X,Y)$
- Tầng 2: vị từ cousin, related, quan hệ tương ứng $C(X,Y)$ và $R(X,Y)$

Tính quan hệ tiềm ẩn đ/v các vị từ tại tầng 1:

- $S = \text{EVAL}(P) \rightarrow S(X,Y) = \Pi_{X,Y} (\sigma_{X \neq Y}(P(X,Z) * P(Y,Z)))$

Tính quan hệ tiềm ẩn đ/v các vị từ tại tầng 2:

- $C = \text{EVAL}(C,P,S) \rightarrow C(X,Y) = \Pi_{X,Y} (P(X,Xp) * P(Y,Yp) * S(Xp,Yp) \cup \Pi_{X,Y} (P(X,Xp) * P(Y,Yp) * C(Xp,Yp))$
- $R = \text{EVAL}(R,P,S) \rightarrow R(X,Y) = S(X,Y) \cup \Pi_{X,Y} (R(X,Z) * P(Y,Z)) \cup \Pi_{X,Y} (R(Z,Y) * P(X,Z))$

79

Ứng dụng thuật toán ta có các bước và giá trị sau

| Bước | S | C | R |
|------|----------------|----------------------|--|
| 1 | cd de fg hi fi | | |
| 2 | | fh fi ii gh gi hi jk | cd de fg hi fi |
| 3 | | jj kk | df dg ch di ci eh ei gj fk hk ij |
| 4 | | | fh dj gh jk gi dk cj ii ck ej ek |
| 5 | | | fi hj gk ik |
| 6 | | | jj kk |

80

Nội dung

- **Logic và CSDL suy diễn**
 - Logic vị từ cấp một
 - CSDL suy diễn
 - CSDL cài đặt và CSDL tiềm ẩn
 - Vấn đề thông tin ẩn
 - Hệ quản trị CSDL suy diễn
- **Ngôn ngữ luật Datalog đối với các CSDL suy diễn**
 - Cú pháp của Datalog
 - Chương trình Datalog
 - So sánh Datalog và ĐSQH
- **Ngữ nghĩa của Chương trình Datalog**
 - Thủ tục tính mô hình nhỏ nhất
 - Tính ngữ nghĩa của Chương trình Datalog sử dụng ĐSQH
- **Đánh giá câu hỏi suy diễn**
 - Đánh giá dưới – lên
 - Đánh giá trên – xuống

81

Đánh giá dưới-lên

- Phương pháp đánh giá dưới-lên:
 - Xuất phát từ CSDL cài đặt, áp dụng các luật suy diễn theo chiều tiến (suy diễn tiến) để tính ngữ nghĩa của chương trình Datalog (tập luật Datalog), lọc kết quả để sinh câu trả lời
- Kỹ thuật đánh giá dưới-lên:
 - Tính ngữ nghĩa của chương trình Datalog
 - Lọc kết quả để sinh câu trả lời cho câu hỏi

82

Đánh giá trên-xuống

- Phương pháp đánh giá trên-xuống:
 - Xuất phát từ câu truy vấn, áp dụng các luật suy diễn theo chiều lùi (suy diễn lùi) để tìm các sự kiện thích đáng, lọc kết quả để sinh câu trả lời
- Kỹ thuật đánh giá trên-xuống:
 - So sánh các thành phần để tìm vị từ/quan hệ và các sự kiện trong tiến trình suy diễn.
 - So khớp và đồng nhất biến:
 - Sau khi so khớp để xác định vị từ, thực hiện đồng nhất biến theo nghĩa thay thế biến bằng một giá trị cụ thể.
 - Thực hiện tiến trình suy diễn lùi để tạo sinh dữ liệu

83

Đánh giá trên-xuống

Ví dụ: Xét các luật sau:

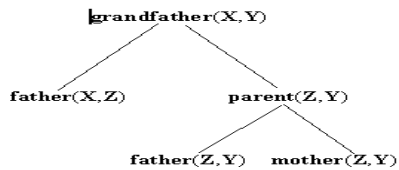
- r1: grandfather(X,Y): - father(X,Z) , parent(Z,Y)
 - r2: parent(X,Y): - father(X,Y)
 - r3: parent(X,Y): - mother(X,Y)
- 2 vị từ EDB father(X,Y), mother(X,Y) tương ứng với 2 quan hệ cơ sở Father(A,B), Mother(A,B)
 - Từ các vị từ EDB, nhờ các luật suy diễn chúng ta có thể tạo các quan hệ cho các IDB là Parent(A,B) hay Grandfather(A,B).

84

Đánh giá trên-xuống

- Đồ thị suy diễn

➤ Có thể mô tả các luật suy diễn bằng đồ thị suy diễn. Ví dụ với 3 luật trên ta có thể tạo đồ thị dạng cây suy diễn và-hoặc ở hình sau:



85

Đánh giá trên-xuống

- Trong tiến trình suy diễn để tạo quan hệ cho vị từ được suy diễn **grandfather(X,Y)**

➤ Tạo quan hệ cho vị từ **father(X,Z)**

➤ Tạo quan hệ cho vị từ **parent(Z,Y)**.

86

Đánh giá trên-xuống

- Do **father(X,Z)** là quan hệ cơ sở/nền nên chỉ cần đánh giá **parent(Z,Y)** bằng cách so khớp và đồng nhất biến để có dạng sau: parent(Z,Y): - father(Z,Y) và parent(Z,Y): - mother(Z,Y)
 - Với vị từ nền **father(Z,Y)** chúng ta sử dụng so khớp và đồng nhất biến theo thứ tự xuất hiện của đối tượng trong vị từ và thứ tự thuộc tính xuất hiện trong quan hệ cơ sở/nền.
 - Tương tự với vị từ nền **mother(Z,Y)**

87

Đánh giá trên-xuống

- Từ quan hệ Father(A,B), chúng ta có các đồng nhất biến sau cho **father(X,Z)** và **father(Z,Y)**.

| Father(A, B) | father(X Z) | father(Z Y) |
|--------------|---------------|---------------|
| an son | an son | an son |
| loc vinh | loc vinh | loc vinh |

88

Đánh giá trên-xuống

Hai bước chính là:

- Tạo đồ thị/cây suy diễn theo các luật.
 - Duyệt cây để đánh giá và tạo sinh dữ liệu cho vị từ được suy diễn.
- Ví dụ: Với các luật:
 - r1: `sibling(X,Y):- parent(Z,X), parent(Z,Y), (X<>Y)`
 - r2: `parent(X,Y):- father(X,Y)`
 - r3: `parent(X,Y):- mother(X,Y)`

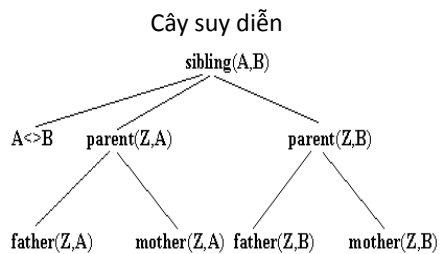
89

Đánh giá trên-xuống

- Đích **sibling(A,B)** tương ứng với vị từ được suy diễn **sibling(X,Y)** sẽ được đánh giá như sau:
 - Tìm luật so khớp được với vị từ được suy diễn và thực hiện đồng nhất biến.
 - Tạo cây con có gốc chính là đích của bài toán. Xử lý đệ qui với các đích con là các lá của cây con vừa mới tạo được. Nếu vị từ trong lá là vị từ nền thì không thể mở rộng được cây con.

90

Đánh giá trên-xuống



91

Đánh giá trên-xuống

- Sau khi đã tạo xong đồ thị suy diễn, duyệt cây để tạo sinh dữ liệu cho các vị từ được suy diễn.

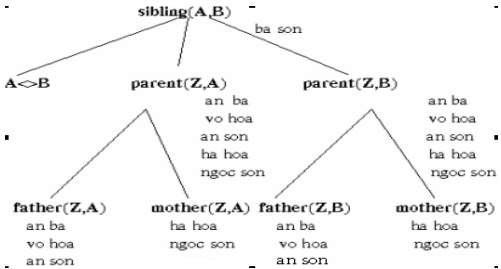
Ý tưởng của thuật toán như sau:

Với 1 luật có phần đầu là 1 vị từ được suy diễn .

- Tạo dữ liệu cho các vị từ trong phần thân của luật.
- Thực hiện các phép toán logic để tạo dữ liệu cho vị từ được suy diễn .

92

Đánh giá trên-xuống



93

94