

# Cơ sở dữ liệu nâng cao

---

## Cơ sở dữ liệu suy diễn

Phạm Nguyên Khang, Đỗ Thanh Nghị  
{pnkhang|dtngghi}@cit.ctu.edu.vn

Cần Thơ  
13-12-2010

# HQTCSDL suy diễn

---

## ■ Mục tiêu

- Giới thiệu các khái niệm CSDL suy diễn
- Giới thiệu sử dụng logic
- Tìm hiểu vấn đề hình thức hóa và đánh giá các câu truy vấn đệ quy
- Tìm hiểu các tiếp cận cài đặt

# Tài liệu tham khảo

---

**[Ramakrishnan, 1997]**

Ramakrishnan, R. (1997)

Database Management Systems

Mc Graw Hill

chapitre 20

**[Bidoit, 1992]**

Bidoit, N. (1992)

Bases de données déductives: présentation de Datalog

Armand Colin

# Động lực

---

- **Tích hợp các chức năng của HQTCSDL và hệ chuyên gia**
  - Lưu trữ
  - Tìm kiếm
  - Suy diễn
- **Cung cấp một hệ thống hoàn chỉnh hỗ trợ việc phát triển ứng dụng và cho phép điều khiển dữ liệu**

# Các chức năng

---

- **Cho phép biểu diễn tri thức**
  - Ngôn ngữ biểu diễn tri thức
  - Tri thức cơ bản (đối tượng hoặc sự kiện)
  - Mối quan hệ giữa các đối tượng
- **Suy diễn thông tin mới từ:**
  - dữ liệu trong CSDL
  - các luật mô tả tri thức

# Các chức năng

---

- **Đảm bảo thực thi hiệu quả quá trình suy diễn**
  - Lưu trữ luật
  - Tối ưu hóa tập luật
  - Điều khiển thực thi

# Ví dụ

---

- **CSDL mô tả mối quan hệ huyết thống**
  - Quan hệ PARENT (cha, con)
  - Quan hệ ANCETRE (tổ tiên, hậu duệ)
- **Câu hỏi của người dùng có dạng:**
  - Tìm các hậu duệ của Jean
  - Tìm các tổ tiên của Paul
- **Cần định nghĩa**
  - làm thế nào để có được hậu duệ và tổ tiên
  - và làm thế nào chúng ta có thể suy diễn được từ dữ liệu của CSDL

**NẾU PARENT(X,Y) VÀ PARENT(Y,Z)**  
**THÌ ANCETRE(X,Z)**

# Vấn đề cần giải quyết

---

## ■ Định nghĩa tri thức

- Ngôn ngữ luật (rules)
- Ngôn ngữ khung (frames)
- Ngôn ngữ kịch bản (scripts)
- Vấn đề biểu diễn tri thức trong trí tuệ nhân tạo

## ■ Chiến lược suy diễn

- Suy diễn tiến
- Suy diễn lùi
- Suy diễn kết hợp tiến và lùi

## ■ Điều khiển thực thi

- Phân tầng chương trình
- Tương hợp các luật



# Tri thức

---

## ■ Định nghĩa tri thức

- Thông tin tổng quát

*Ví dụ :* Các loại bằng cấp khác nhau

- Tri thức riêng của một lĩnh vực

*Ví dụ:* Ở Bắc Mỹ, bachelor là một bằng cấp ở bậc đại học

- Chiến lược suy diễn

*Ví dụ:* Nếu một sinh viên học ở Québec và anh ta có bằng bachelor thì anh ta có bằng đại học

# Biểu diễn tri thức

---

## ■ Biểu diễn tri thức

- Sử dụng trừu tượng hóa: các khái niệm
- Công cụ biểu diễn tri thức
- Tổ chức tri thức

## ■ Ngôn ngữ hình thức

- Định nghĩa và sử dụng ngôn ngữ hình thức để biểu diễn tất cả các loại tri thức

logic

Ngôn ngữ luật

Ngôn ngữ đối tượng

# Sử dụng tri thức

---

## ■ Sử dụng tri thức

- **Tìm kiếm**

Tri thức được lưu trữ đâu đó, ta phải tìm kiếm được các kết quả phù hợp

- **Đạt được**

Cần phải đạt được tri thức mới

- **Suy diễn**

Suy diễn thông tin mới từ thông tin đang có



**Logic**

# Sử dụng logic

---

## ■ Logic bậc 1

- **Ngôn ngữ hình thức cho phép biểu diễn**  
đối tượng  
quan hệ giữa các đối tượng
- **được định nghĩa bởi**  
tập từ vựng  
văn phạm
- **Cho phép chúng ta**  
xây dựng công thức  
diễn dịch công thức

# Logic bậc 1

---

## ■ Ký hiệu

- Biến:  $x, y, z$
- Hằng:  $a, b, c$
- Vị từ:  $P, Q, R$ , theo sau là các đối số được đặt trong cặp dấu ngoặc  $()$
- Phép toán logic:  $\neg, \Rightarrow, \wedge, \vee$
- Hàm :  $f, g, h$  theo sau là các đối số được đặt trong cặp dấu ngoặc  $()$
- Lượng từ:  $\forall, \exists$

# Logic bậc 1

---

## ■ Văn phạm cho phép xây dựng công thức

- Mục (term): được định nghĩa đệ quy

Hằng: "Paul"

Biến:  $x$

Áp dụng hàm:  $f(x); f(g(y))$

- Luật xây dựng mục mới

Nếu  $t_1, t_2, t_n$  là các mục (vd: "Paul";  $x$ )

Thì  $P(t_1, t_2, t_n)$  cũng là một mục (vd:  $P(\text{"Paul"}, x)$ )

# Logic bậc 1

---

- **Biểu thức**

- Mục là biểu thức

- Nếu  $F1, F2$  là biểu thức

- thì  $F1 \wedge F2, F1 \vee F2, F1 \Rightarrow F2$  và  $\neg F1$  là biểu thức  
và  $\forall x F1, \exists x F2$  cũng là biểu thức



# Diễn dịch công thức

---

## ■ Diễn dịch công thức của logic bậc 1

- kết hợp một giá trị luận lý (đúng, sai) vào một công thức

## ■ Cần định nghĩa

- Lĩnh vực đang nói đến  $D$   
Biến và hằng
- Quan hệ giữa các đối tượng trong lĩnh vực  $D$   
Vị từ
- Hàm  $D^n \rightarrow D$   
Kí hiệu hàm

# Diễn dịch công thức

---

## ■ Diễn dịch công thức :

- |                       |                |
|-----------------------|----------------|
| • $F1 \wedge F2$      | $F1 \vee F2$   |
| • $F1 \Rightarrow F2$ | $\neg F1$      |
| • $\forall x F1$      | $\exists x F2$ |

# Ví dụ

---

- **Những người chơi tennis đều là người thích thể thao**

$\forall x \text{ Person}(x) \wedge \text{Pratic}(\text{tennis}, x) \Rightarrow \text{Sportive}(x)$

- **Paul là người**

$\text{Person}(\text{Paul})$

- **Nếu người x đặt hàng món hàng y thì y là một sản phẩm**

$\forall x (\forall y (\text{Command}(x, y) \Rightarrow (\text{Product}(y)))$

- **Tổ tiên của Paul là ai ?**

$\text{Ancetres}(x, \text{Paul}) ?$

A thick red line forming an L-shape, starting with a vertical segment on the left and a horizontal segment extending to the right.

**CSDL logic**

# CSDL và logic

---

## ■ Các việc cần phải làm với CSDL suy diễn

- Hiểu CSDL thông qua logic
- Sử dụng logic để định nghĩa (hoặc định nghĩa lại) các phép toán đại số quan hệ
- Sử dụng logic để định nghĩa cơ chế suy diễn
- Sử dụng logic trong CSDL hướng đối tượng

## ■ Sử dụng logic trong CSDL

- Sử dụng logic vị từ
- Định nghĩa phép tính quan hệ giữa các mẫu tin (tuples)
- Giới thiệu câu truy vấn đệ quy

# CSDL suy diễn

---

## ■ CSDL ngoại diễn

- tương ứng với tập sự kiện đang có
- được xây dựng từ nội dung của CSDL
- trong CSDL quan hệ: tập các quan hệ

## ■ CSDL nội hàm

- tương ứng với các sự kiện có thể được suy diễn ra
- sự kiện không có sẵn trong CSDL
- tập luật chính là phương tiện để sinh ra các sự kiện mới

# Datalog

---

## ■ Ngôn ngữ logic

- tương tự như prolog
- dành cho CSDL

## ■ Ngôn ngữ CSDL

- DATA và LOGic
- thao tác dữ liệu dựa trên logic
- Khả năng biểu diễn tốt hơn SQL

## ■ Ngôn ngữ cho CSDL suy diễn

- dựa trên kiểu mẫu (prototypes)
- cho phép so sánh khả năng biểu của của các ngôn ngữ khác

# Datalog

---

## ■ Tiên đề của 1 CSDL suy diễn

- Tiên đề của CSDL ngoại diên  
Parent (Jacques, Olivier)
- Tiên đề của CSDL nội hàm: các biểu thức logic  
 $\text{Parent}(x, y) \Rightarrow \text{Ancetre}(x, y)$

## ■ Ngôn ngữ luật

- là ngôn ngữ mô tả
- cho phép thực hiện các thao tác cơ bản trong CSDL:  
chọn, chiếu, kết nối, ...
- cách tiếp cận tương tự như Prolog



# Datalog

---

## ■ Cú pháp của Datalog

- Ngôn ngữ luật chi CSDL
- Mô tả quan hệ suy diễn dựa trên chuẩn Horn

## ■ Các mệnh đề theo chuẩn Horn :

$$Q \leftarrow P1 \wedge P2 \wedge \dots \wedge Pn$$

- biểu thức không chứa lượng tử
- có dạng chuẩn VÀ
- Chỉ có 1 biến ở vế trái

## ■ Biến đổi

- Tất cả các biểu thức logic đều có thể được chuyển về chuẩn Horn

# Datalog

---

## ■ Ví dụ chương trình Datalog

```
{  
  parent (jacques, olivier) ←  
  ancetre(x, y) ← parent (x, y)  
  parent (olivier, adrien) ←  
  ancetre (x, z) ← ancetre (x, y) ∧ parent (y, z)  
  parent (suzanne, jacques) ←  
  parent (olivier, juliette) ←  
}
```

# Chú ý

---

- Thứ tự luật trong chương trình không quan trọng
- Về trái: kết luận
- Về phải: các tiên đề



# **Đại số quan hệ và DataLog**

# Biểu diễn các phép toán

---

## ■ Cho các quan hệ sau:

- Person (NP, Nom, Prénom, Ville)
- Student (NE, NomEtud, PreEtud, Ville, Age)
- Inscription (NE, NC, Session, Date)

## ■ Hợp (union): trích tên và họ của người (person) và của sinh viên (Student)

- $R(y,z) \leq \text{Person}(-, y, z, -)$
- $R(y,z) \leq \text{Students}(-, y, z, -)$

# Đại số

---

- **Hiệu: tìm người không phải sinh viên**
  - $R(y,z) \Leftarrow \text{Person}(-, y, z, -) \text{ and NOT Student}(-, y, z, -)$
- **Giao: tìm người là sinh viên**
  - $R(y,z) \Leftarrow \text{Person}(-, y, z, -) \text{ and Student}(-, y, z, -)$
- **Chiếu: tìm tên và họ của sinh viên**
  - $R(y,z) \Leftarrow \text{Students}(-, y, z, -)$

# Đại số quan hệ và Datalog

---

- **Chọn: tìm mã số (NP) của những người sống ở Montréal**
  - $R(x) \Leftarrow \text{Person}(x, -, -, \text{"Montréal"})$
  - hoặc:
  - $R(x) \Leftarrow \text{Person}(x, -, -, w) \text{ AND } w = \text{"Montréal"}$  »
- **Kết nối (join): tìm họ và tên của các sinh viên có đăng ký học**
  - $R(y, z) \Leftarrow \text{Student}(x, -, -, -) \text{ AND } \text{Inscription}(x, -, -, -)$

# **Chiến lược thực thi**



# Vấn đề

---

## ■ Thực thi một chương trình luật

- Sử dụng Datalog để truy vấn CSDL suy diễn
- Thực thi một chương trình luật như thế nào ?
- Một số chương trình rất phức tạp
- Sử dụng chiến lược nào ?

## ■ Cách tiếp cận:

- Suy diễn tiến
- Suy diễn lùi
- Cơ chế điều khiển

# Suy diễn tiến

---

## ■ Nguyên lý:

- Bắt đầu từ dữ liệu để thiết lập câu trả lời
- Tất cả các sự kiện (fact) phải suy diễn đều được suy diễn
- Lọc các sự kiện phù hợp với câu truy vấn

# Suy diễn tiến

---

## ■ Ví dụ :

- parent (x, adrien) ?

## ■ Bước 1 :

- Sinh ra tất cả các tổ tiên bằng cách áp dụng luật lên tất cả các sự kiện ban đầu (được khởi tạo trước)
- Bước này dừng khi không thể áp dụng được luật nào nữa

## ■ Bước 2 :

- Lọc lại để tìm kết quả

# Suy diễn tiến

---

**Luật :**  $\text{parent}(x, y) \leftarrow \text{father}(x, y)$   
 $\text{parent}(x, y) \leftarrow \text{mother}(x, y)$

**Câu truy vấn:**  $\text{parent}(x, \text{adrien})$

**Bước 1 : sự kiện**

$\text{father}(\text{jacques}, \text{olivier})$   
 $\text{father}(\text{olivier}, \text{adrien})$   
 $\text{mother}(\text{suzanne}, \text{jacques})$   
 $\text{mother}(\text{brigitte}, \text{adrien})$   
 $\text{mother}(\text{colette}, \text{olivier})$

**kết quả (sự kiện mới)**

$\text{parent}(\text{jacques}, \text{olivier})$   
 $\text{parent}(\text{olivier}, \text{adrien})$   
 $\text{parent}(\text{suzanne}, \text{jacques})$   
 $\text{parent}(\text{brigitte}, \text{adrien})$   
 $\text{parent}(\text{colette}, \text{olivier})$

**Etape 2 : lọc**

$\text{parent}(\text{olivier}, \text{adrien})$   
 $\text{parent}(\text{brigitte}, \text{adrien})$

# Suy diễn lùi

---

## ■ Nguyên lý:

- bắt đầu từ câu truy vấn của người dùng
- quay lên các giá trị đã biết của các vị từ thông qua luật khi suy diễn lùi
- việc quay lên dừng lại khi ta nhận được các sự kiện đã được lưu trữ trong CSDL
- nếu các sự kiện đều được tìm thấy trong CSDL, câu trả lời cho câu truy vấn là đúng.

## ■ Ưu điểm:

- Ta chỉ tìm các sự kiện phù hợp với câu truy vấn

# Suy diễn lùi

---

**Câu truy vấn: ancêtre (x, adrien) ?**

luật 1:             $\text{parent}(x, y) \leftarrow \text{father}(x, y)$

luật 2:             $\text{parent}(x, y) \leftarrow \text{mother}(x, y)$

**Sự kiện phù hợp :**

luật 1 :             $\text{father}(x, \text{adrien}) ?$

kết quả :         $\text{father}(\text{olivier}, \text{adrien})$

luật 2 :             $\text{mother}(x, \text{adrien}) ?$

kết quả :         $\text{mother}(\text{brigitte}, \text{adrien})$

# Đánh giá các luật đệ quy

---

## ■ Luật đệ quy:

- Trong định nghĩa luật có sử dụng lại khái niệm cần định nghĩa
- Ví dụ: định nghĩa khái niệm tổ tiên

$\text{ancetre}(x, y) \leftarrow \text{parent}(x, y)$

$\text{ancetre}(x, z) \leftarrow \text{ancetre}(x, y) \wedge \text{parent}(y, z)$

## ■ Cần

- Giảm thời gian thực thi
- Giảm số lượng bộ (tuples) sinh ra
- Đảm bảo việc thực thi phải kết thúc
- Giảm tương tác với hệ thống lưu trữ

# Chiến lược

---

## ■ Phương pháp ngây thơ (naïve)

- Sinh ra sự kiện mới bằng cách áp dụng tất cả các luật lên tất cả các sự kiện đang có cho đến khi không thể áp dụng được nữa

## ■ Phương pháp nửa ngây thơ (semi-naïve)

- Suy diễn tiến bằng cách chỉ áp dụng các luật lên các sự kiện mới được sinh ra, ta sẽ giảm được các số lượng các sự kiện

## ■ Phương pháp tập hợp ma thuật

- Trước khi áp dụng suy diễn tiến đánh dấu các quan hệ hữu ích lên các vị từ đệ quy bằng các vị từ ma thuật (magical predicates)



# Điều khiển thực thi

---

## ■ Vấn đề liên quan đến thực thi

- Chọn luật để kích hoạt
- Tối ưu hóa việc truy cập CSDL
- Điều khiển kết thúc
- Sắp thứ tự luật

## ■ Giải pháp

- Phân tầng chương trình
- Giải thuật tối ưu

# Stratification: ví dụ

---

## ■ Cho các quan hệ sau:

- LIBRARY (Book) chứa tất cả các quyển sách trong thư viện
- LECTURE (Lecteur, Book) mô tả ai đọc quyển sách nào

# Ví dụ

---

## ■ Câu truy vấn SQL:

- Tìm các độc giả đọc tất cả các quyển sách

```
SELECT DISTINCT Lecteur
FROM Lecture L1
WHERE NOT EXISTS (SELECT *
                  FROM LIBRARY B1
                  WHERE NOT EXISTS (SELECT *
                                    FROM Lecture
                                    WHERE lecteur=L1.lecteur
                                    AND Book=B1.Book))
```

# Ví dụ

---

## ■ Biểu diễn trong Datalog:

Time (x, y) <== Lecture (x, -) AND Library (y)

-----

Bad (x) <== Time (x, y) AND NOT Lecture (x, y)

-----

Solution (x) <== Lecture (x, -) AND NOT Bad (x)

# Điều khiển thực thi

---

## ■ Phân tần

- Nếu có phép toán hiệu, cần phải sinh ra tất các mẫu tin cho một luật trước khi thực hiện luật kế tiếp
- Ta không thể làm phép toán hiệu (giữa kết quả của luật 1 và của luật 2) khi việc thực thi luật 1 chưa kết thúc
- Trong ví dụ của chúng ta: cần phải có 2 tầng

# SQL3 và câu truy vấn đệ quy

---

## ■ Định nghĩa quan hệ nội hàm

- Sử dụng vị từ **WITH**

WITH Rel AS <định nghĩa Rel> <truy vấn suy ra Rel>

- Khả năng sử dụng từ khóa **RECURSIVE**
- Sử dụng toán tử hợp (union) để định nghĩa quan hệ nội hàm
- Định nghĩa quan hệ nội hàm chỉ có giá trị trong ngữ cảnh của câu truy vấn **WITH**
- Kết quả của định nghĩa cho quan hệ này là tạm thời

# SQL3 và câu truy vấn đệ quy

---

$\text{ancetre}(\text{Anc}, \text{Desc}) \leftarrow \text{parent}(\text{Par}, \text{Enf})$   
 $\text{ancetre}(\text{Anc}, \text{Enf}) \leftarrow \text{ancetre}(\text{Anc}, x) \wedge \text{parent}(x, \text{Enf})$

**WITH RECURSIVE** Ancetres (Anc, Desc) **AS**

**(SELECT** Par, Enf **FROM** Parents)

**UNION**

**(SELECT** A.Anc, P.Enf

**FROM** Ancêtres A, Parents P

**WHERE** A.Desc=P.Par)

# SQL3

---

**SELECT \* FROM Ancetres;**

- Câu truy vấn này cho phép sinh ra tất cả các tổ tiên ancetres
- Mệnh đề **SELECT** có thể là câu truy vấn sinh ra kết quả của phép chọn:

**SELECT \* from Ancetres where Anc="Paul";**



# Kết luận

---

## ■ CSDL suy diễn mang lại

- Khả năng biểu diễn
- Xử lý câu truy vấn đệ quy

## ■ Khó khăn

- Thực thi hiệu quả các câu truy vấn đệ quy
- Tối ưu hóa câu truy vấn



Cám ơn !