Trí Tuệ Nhân Tạo

(Artificial Intelligence)

Lê Thanh Hương

Viện Công nghệ thông tin và Truyền thông
Trường Đại Học Bách Khoa Hà Nội

Nội dung môn học

Chương 1. Tổng quan

Chương 2. Tác tử thông minh

Chương 3. Giải quyết vấn đề

Chương 4. Tri thức và suy diễn

- Giới thiệu về logic
- Logic định đề
- Logic vị từ

Chương 5. Học máy

Giới thiệu về logic

- **Logic** là ngôn ngữ hình thức cho phép (giúp) biểu diễn thông tin dưới dạng các kết luận có thể được đưa ra
  - Logic = Syntax + Semantics
- **Cú pháp (syntax)**: để xác định các mệnh đề (sentences) trong một ngôn ngữ.
- **Ngữ nghĩa (semantics)**: để xác định “ý nghĩa" của các mệnh đề trong một ngôn ngữ
  - Tức là, xác định sự đúng đắn của một mệnh đề
- Ví dụ: Trong ngôn ngữ của toán học
  - $(x+2 \geq y)$ là một mệnh đề; $(x+y > \{\})$ không phải là một mệnh đề
  - $(x+2 \geq y)$ là đúng nếu và chỉ nếu giá trị $(x+2)$ không nhỏ hơn giá trị y
  - $(x+2 \geq y)$ là đúng khi $x = 7$, $y = 1$
  - $(x+2 \geq y)$ là sai khi $x = 0$, $y = 6$

Cú pháp (syntax)

- Cú pháp = Ngôn ngữ + Lý thuyết chứng minh
- **Ngôn ngữ (Language)**
  - Các ký hiệu (symbols), biểu thức (expressions), thuật ngữ (terms), công thức (formulas) hợp lệ
  - Ví dụ: *one plus one equal two*
- **Lý thuyết chứng minh (Proof theory)**
  - Tập hợp các luật suy diễn cho phép chứng minh (suy luận ra) các biểu thức
  - Ví dụ: Luật suy diễn *any plus zero* ⊢ *any*
- Một **định lý (theorem)** là một mệnh đề logic cần chứng minh
- Việc chứng minh một định lý không cần phải xác định ngữ nghĩa (interpretation) của các ký hiệu!

Ngữ nghĩa (semantics)

- Ngữ nghĩa = Ý nghĩa (diễn giải) của các ký hiệu
- Ví dụ
  - $\mathcal{I}$(*one*) nghĩa là **1** ($\in$ N)
  - $\mathcal{I}$(*two*) nghĩa là **2** ($\in$ N)
  - $\mathcal{I}$(*plus*) nghĩa là phép cộng **+** : N x N $\rightarrow$ N
  - $\mathcal{I}$(*equal*) nghĩa là phép so sánh bằng **=** : N x N $\rightarrow$ {*true*, *false*}
  - $\mathcal{I}$(*one plus one equal two*) nghĩa là *true*
- Nếu diễn giải của một biểu thức là đúng (true), chúng ta nói rằng phép diễn giải này là một **mô hình (model)** của biểu thức
- Một biểu thức đúng đối với bất kỳ phép diễn giải nào thì được gọi là một biểu thức **đúng đắn (valid)**
  - Ví dụ: A OR NOT A

Tính bao hàm

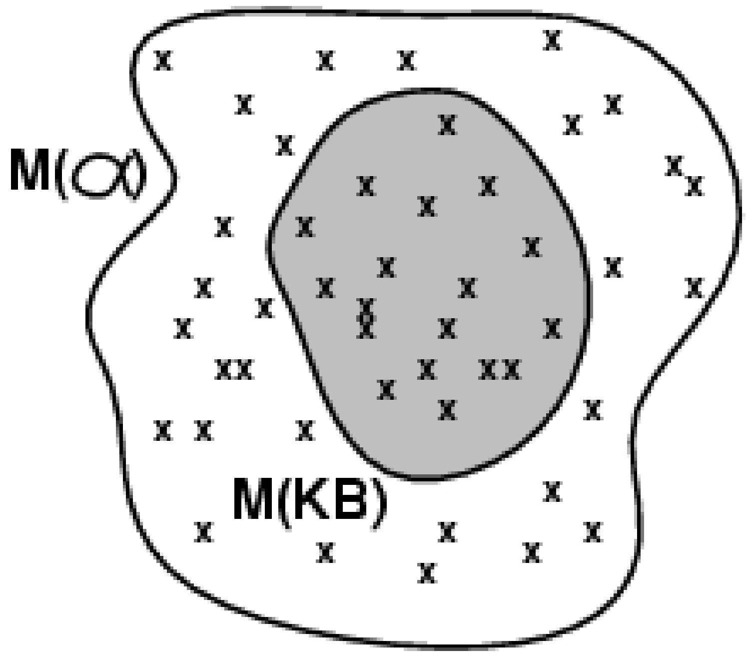
- Tính bao hàm có nghĩa là một cái gì đó tuân theo (bị hàm chứa ý nghĩa bởi, được suy ra từ) một cái gì khác:

$$KB \models \alpha$$

- Một cơ sở tri thức *KB* **bao hàm** (hàm chứa) mệnh đề α nếu và chỉ nếu α là đúng **trong mọi mô hình** (thế giới) mà trong đó *KB* là đúng. Tức là: nếu *KB* đúng, thì α cũng phải đúng
  - Ví dụ: Nếu một cơ sở tri thức *KB* chứa các mệnh đề "Đội bóng A đã thắng" và "Đội bóng B đã thắng", thì *KB* bao hàm mệnh đề "Đội bóng A hoặc đội bóng B đã thắng"
  - Ví dụ: Mệnh đề (x+y = 4) bao hàm mệnh đề (4 = x+y)

Các mô hình

- Các nhà logic học thường hay xem xét các sự việc theo các mô hình (models)
- Các mô hình là các không gian (thế giới) có cấu trúc, mà trong các không gian đó tính đúng đắn (của các sự việc) có thể đánh giá được
- **Định nghĩa:** *m* là một mô hình của mệnh đề α nếu α là đúng trong *m*
- *M(α)* là tập hợp tất cả các mô hình của α
- KB ╞ α nếu và chỉ nếu *M(KB)* ⊆ *M*(α)
  - Ví dụ: *KB* = "Đội bóng A đã thắng và đội bóng B đã thắng", α = "Đội bóng A đã thắng"



Suy diễn logic (1)

- *KB* $\vdash_i \alpha$
  - Mệnh đề *α* **được suy ra** từ *KB* bằng cách áp dụng thủ tục (suy diễn) *i*
  - (Nói cách khác) Thủ tục *i* **suy ra** mệnh đề *α* từ *KB*
- **Tính đúng đắn (soundness)**
  - Một thủ tục suy diễn *i* được gọi là **đúng đắn (sound)**, nếu thủ tục *i* suy ra **chỉ** các mệnh đề được bao hàm (entailed sentences)
  - Thủ tục *i* là đúng đắn, nếu bất cứ khi nào *KB* $\vdash_i \alpha$, thì cũng đúng đối với *KB* $\models \alpha$
  - Nếu thủ tục *i* suy ra mệnh đề *α*, mà *α* không được bao hàm trong *KB*, thì thủ tục *i* là không đúng đắn (unsound)

Suy diễn logic (2)

- **Tính hoàn chỉnh (completeness)**
  - Một thủ tục suy diễn *i* được gọi là **hoàn chỉnh (complete)**, nếu thủ tục *i* có thể suy ra **mọi** mệnh đề được bao hàm (entailed sentences)
  - Thủ tục *i* là hoàn chỉnh, nếu bất cứ khi nào $KB \models \alpha$, thì cũng đúng đối với $KB \vdash_i \alpha$
- (Trong phần tiếp theo của bài giảng) chúng ta sẽ xét đến logic vị từ bậc 1 (first-order logic)
  - Có khả năng biểu diễn (diễn đạt) hầu hết các phát biểu logic
  - Với logic vị từ bậc 1, tồn tại một thủ tục suy diễn *đúng đắn* và *hoàn chỉnh*

Suy diễn logic (3)

- Logic là một cách để biểu diễn hình thức và suy diễn tự động
- Việc suy diễn (reasoning) có thể được thực hiện ở mức cú pháp (bằng các chứng minh): **suy diễn diễn dịch (deductive reasoning)**
- Việc suy diễn có thể được thực hiện ở mức ngữ nghĩa (bằng các mô hình): **suy diễn dựa trên mô hình (model-based reasoning)**

Suy diễn logic (4)

- Suy diễn ngữ nghĩa ở mức của một phép diễn giải (mô hình):
  - Với một biểu thức, có tồn tại một mô hình không? **có thể thỏa mãn được (satisfiability)?**
  - Với một biểu thức và một phép diễn giải, kiểm tra xem phép diễn giải có phải là một mô hình của biểu thức không?: **kiểm tra mô hình (model checking)**
- Suy diễn ngữ nghĩa ở mức của tất cả các phép diễn giải có thể: **kiểm tra tính đúng đắn (validity checking)**

Logic định đề: Cú pháp (1)

- Logic định đề (propositional logic) là loại logic đơn giản nhất

- **Biểu thức định đề (propositional formula)**
  - Một ký hiệu định đề (S_1, S_2, …) là một biểu thức (định đề)
  - Các giá trị hằng logic **đúng (true)** và **sai (false)** là các biểu thức
  - Nếu S_1 là một biểu thức, thì ($\neg S_1$) cũng là một biểu thức (Phép **phủ định**)

Logic định đề: Cú pháp (2)

- **Biểu thức định đề (propositional formula)…**
  - Nếu S_1 và S_2 là các biểu thức, thì $(S_1 \wedge S_2)$ cũng là một biểu thức (Phép **kết hợp / và**)
  - Nếu S_1 và S_2 là các biểu thức, thì $(S_1 \vee S_2)$ cũng là một biểu thức (Phép **tuyển / hoặc**)
  - Nếu S_1 và S_2 là các biểu thức, thì $(S_1 \Rightarrow S_2)$ cũng là một biểu thức (Phép **suy ra / kéo theo**)
  - Nếu S_1 và S_2 là các biểu thức, thì $(S_1 \Leftrightarrow S_2)$ cũng là một biểu thức (Phép **tương đương**)
  - Không gì khác (các dạng trên) là một biểu thức

Cú pháp của logic định đề: Ví dụ

- p
- q
- r
- true
- false
- $\neg p$
- $(\neg p) \wedge \text{true}$
- $\neg((\neg p) \vee \text{false})$
- $(\neg p) \Rightarrow (\neg((\neg p) \vee \text{false}))$
- $(p \wedge (q \vee r)) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

Thứ tự ưu tiên của các toán tử logic

- Thứ tự ưu tiên của các toán tử logic (từ cao xuống thấp)
  - $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- Sử dụng cặp ký tự "()" để xác định mức độ ưu tiên
- Các ví dụ
  - $p \wedge q \vee r$ tương đương $(p \wedge q) \vee r$
    chứ không phải $p \wedge (q \vee r)$
  - $\neg p \wedge q$ tương đương $(\neg p) \wedge q$
    chứ không phải $\neg(p \wedge q)$
  - $p \wedge \neg q \Rightarrow r$ tương đương $(p \wedge (\neg q)) \Rightarrow r$
    chứ không phải $p \wedge (\neg(q \Rightarrow r))$ hoặc $p \wedge ((\neg q) \Rightarrow r)$

Logic định đề: Ngữ nghĩa (1)

- Với một mô hình (model) cụ thể, nó sẽ xác định giá trị đúng/sai cho mỗi ký hiệu định đề
  - Ví dụ: Với 3 ký hiệu S_1, S_2 và S_3, thì có thể lấy ví dụ một mô hình m_1 xác định như sau:

    $m_1 \equiv (S_1=\text{sai}, S_2=\text{đúng}, S_3=\text{sai})$

- Với 3 ký hiệu định đề như ví dụ trên, có thể chỉ ra 8 mô hình có thể

Logic định đề: Ngữ nghĩa (2)

- Ngữ nghĩa của một mô hình *m* = Các quy tắc để đánh giá giá trị chân lý (đúng/sai) của các mệnh đề trong mô hình *m* đó

  $\neg S_1$ là đúng, khi và chỉ khi S_1 là sai

  $S_1 \wedge S_2$ là đúng, khi và chỉ khi S_1 là đúng <u>và</u> S_2 là đúng

  $S_1 \vee S_2$ là đúng, khi và chỉ khi S_1 là đúng hoặc S_2 là đúng

  $S_1 \Rightarrow S_2$ là đúng, khi và chỉ khi S_1 là sai hoặc S_2 là đúng;
  là sai, khi và chỉ khi S_1 là đúng và S_2 là sai

  $S_1 \Leftrightarrow S_2$ là đúng, khi và chỉ khi $S_1 \Rightarrow S_2$ là đúng và $S_2 \Rightarrow S_1$ là đúng

- Ví dụ: Với mô hình m_1 như trong ví dụ trước, thì giá trị của biểu thức logic định đề sau sẽ là:

  $\neg S_1 \wedge (S_2 \vee S_3) = \textit{đúng} \wedge (\textit{đúng} \vee \textit{sai}) = \textit{đúng} \wedge \textit{đúng} = \textit{đúng}$

Ngữ nghĩa của logic định đề: Ví dụ (1)

- Xét mô hình *$m_1 \equiv$ (p=đúng, q=sai)*, ta có ngữ nghĩa (giá trị logic) của các biểu thức sau
  - $\neg p$ là *sai*
  - $\neg q$ là *đúng*
  - $p \wedge q$ là sai
  - $p \vee q$ là đúng
  - $p \Rightarrow q$ là *sai*
  - $q \Rightarrow p$ là *đúng*
  - $p \Leftrightarrow q$ là sai
  - $\neg p \Leftrightarrow q$ là đúng

Ngữ nghĩa của logic định đề: Ví dụ (2)

- Xét mô hình $m_2 \equiv (p=sai,\ q=đúng)$, ta có ngữ nghĩa (giá trị logic) của các biểu thức sau
  - $\neg p$ là *đúng*
  - $\neg q$ là *sai*
  - $p \wedge q$ là sai
  - $p \vee q$ là đúng
  - $p \Rightarrow q$ là *đúng*
  - $q \Rightarrow p$ là *sai*
  - $p \Leftrightarrow q$ là sai
  - $\neg p \Leftrightarrow q$ là đúng

Bảng chân lý đối với các toán tử logic

S_1	S_2	$\neg S_1$	$S_1 \wedge S_2$	$S_1 \vee S_2$	$S_1 \Rightarrow S_2$	$S_1 \Leftrightarrow S_2$
sai	sai	đúng	sai	sai	đúng	đúng
sai	đúng	đúng	sai	đúng	đúng	sai
đúng	sai	sai	sai	đúng	sai	sai
đúng	đúng	sai	đúng	đúng	đúng	đúng

Các phép biến đổi tương đương

Hai câu có ý nghĩa tương đương nếu cùng giá trị đúng:

$$\left.\begin{aligned}(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha)\\ (\alpha \vee \beta) &\equiv (\beta \vee \alpha)\end{aligned}\right\}\ \text{giao hoán}$$

$$\left.\begin{aligned}((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma))\\ ((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma))\end{aligned}\right\}\ \text{kết hợp}$$

$$\neg(\neg\alpha) \equiv \alpha \quad \text{phủ định kép}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{tương phản}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$$

$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$$

$$\left.\begin{aligned}\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta)\\ \neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta)\end{aligned}\right\}\ \text{de Morgan}$$

$$\left.\begin{aligned}(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))\\ (\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))\end{aligned}\right\}\ \text{phân phối}$$

Biểu diễn bằng logic định đề: Ví dụ

- Giả sử chúng ta có các định đề sau
  - $p \equiv$ "Chiều nay trời nắng"
  - $q \equiv$ "Thời tiết lạnh hơn hôm qua"
  - $r \equiv$ "Tôi sẽ đi bơi"
  - $s \equiv$ "Tôi sẽ đi đá bóng"
  - $t \equiv$ "Tôi sẽ về đến nhà vào buổi tối"
- Biểu diễn các phát biểu trong ngôn ngữ tự nhiên
  - "Chiều nay trời *không* nắng và thời tiết lạnh hơn hôm qua": $\neg p \land q$
  - "Tôi sẽ đi bơi *nếu như* chiều nay trời nắng": $p \rightarrow r$
  - "*Nếu* tôi (sẽ) *không* đi bơi *thì* tôi sẽ đi đá bóng": $\neg r \rightarrow s$
  - "Nếu tôi (sẽ) đi đá bóng thì tôi sẽ về nhà vào buổi tối": $s \rightarrow t$

Mâu thuẫn và Tautology

- Một biểu thức logic định đề luôn có giá trị sai (false) trong mọi phép diễn giải (mọi mô hình) thì được gọi là một **mâu thuẫn (contradiction)**
  - Ví dụ: $(p \wedge \neg p)$
- Một biểu thức logic định đề luôn có giá trị đúng (true) trong mọi phép diễn giải (mọi mô hình) thì được gọi là một **tautology**
  - Ví dụ: $(p \vee \neg p)$

$$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$$

$$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$$

Tính thỏa mãn được và Tính đúng đắn

- Một biểu thức logic định đề là **thỏa mãn được (satisfiable)**, nếu biểu thức đó đúng trong *một mô mình* nào đó
  - Ví dụ: $A \vee B$, $A \wedge B$
- Một biểu thức là **không thể thỏa mãn được (unsatisfiable)**, nếu *không tồn tại bất kỳ mô hình nào* mà trong đó biểu thức là đúng
  - Ví dụ: $A \wedge \neg A$
- Một biểu thức là **đúng đắn (valid)**, nếu biểu thức đúng trong *mọi mô hình*
  - Ví dụ: *đúng*; $A \vee \neg A$; $A \Rightarrow A$; $(A \wedge (A \Rightarrow B)) \Rightarrow B$

Bài toán chứng minh logic

- Với một cơ sở tri trức (một tập các mệnh đề) *KB* và một mệnh đề α cần chứng minh (gọi là một định lý)
- Cơ sở tri thức *KB* có bao hàm (về mặt ngữ nghĩa) α hay không: KB $\models \alpha$?
  - Nói cách khác, α có thể được suy ra (được chứng minh) từ cơ sở tri thức KB hay không?
- Câu hỏi đặt ra: Liệu có tồn tại một thủ tục (suy diễn) có thể giải quyết được bài toán chứng minh logic, trong một số hữu hạn các bước?
  - Đối với logic định đề, câu trả lời là có!

Giải quyết bài toán chứng minh logic

- Mục đích: để trả lời câu hỏi KB $\models \alpha$?
- Có 3 phương pháp (chứng minh) phổ biến:
  - Sử dụng bảng chân lý (Truth-table)
  - Áp dụng các luật suy diễn (Inference rules)
  - Chuyển về bài toán chứng minh thỏa mãn (SAT)
    - Phương pháp chứng minh bằng phản chứng (Refutation)

Chứng minh dựa trên bảng chân lý (1)

- Bài toán chứng minh: KB $\models \alpha$?
- Kiểm tra tất cả các phép diễn giải có thể (tất cả các mô hình có thể) mà trong đó KB là đúng, để xem α đúng hay sai
- Bảng chân lý: Liệt kê các giá trị chân lý (đúng/sai) của các mệnh đề, đối với tất cả các phép diễn giải có thể
  - Các phép gán giá trị đúng/sai đối với các ký hiệu định đề

		KB		α	
p	q	$p \vee q$	$p \leftrightarrow q$	$(p \vee \neg q) \wedge q$	
đúng	đúng	đúng	đúng	đúng	← chứng minh
đúng	sai	đúng	sai	sai	
sai	đúng	đúng	sai	sai	
sai	sai	sai	đúng	sai	

Chứng minh dựa trên bảng chân lý (2)

- KB = $(p \vee r) \wedge (q \vee \neg r)$
- $\alpha = (p \vee q)$
- KB $\models \alpha$?

p	q	r	$p \vee r$	$q \vee \neg r$	KB	α
đúng	đúng	đúng	đúng	đúng	đúng	đúng
đúng	đúng	sai	đúng	đúng	đúng	đúng
đúng	sai	đúng	đúng	sai	sai	đúng
đúng	sai	sai	đúng	đúng	đúng	đúng
sai	đúng	đúng	đúng	đúng	đúng	đúng
sai	đúng	sai	sai	đúng	sai	đúng
sai	sai	đúng	đúng	sai	sai	sai
sai	sai	sai	sai	đúng	sai	sai

Chứng minh dựa trên bảng chân lý (3)

- Đối với logic định đề, phương pháp chứng minh dựa trên bảng chân lý có tính *đúng đắn (sound)* và *hoàn chỉnh (complete)*
- Độ phức tạp tính toán của phương pháp chứng minh dựa trên bảng chân lý
  - Hàm mũ đối với số lượng (n) các ký hiệu định đề: 2^n
  - Nhưng chỉ có một tập con (nhỏ) của tập các khả năng gán giá trị chân lý, mà trong đó KB và α là đúng

Chứng minh bằng các luật suy diễn (1)

- Luật suy diễn **Modus ponens**

$$\frac{p \rightarrow q,\ \ p}{q}$$

- Luật suy diễn loại bỏ liên kết VÀ (**And-Elimination**)

$$\frac{p_1 \wedge p_2 \wedge \ldots \wedge p_n}{p_i} \qquad (i=1..n)$$

- Luật suy diễn đưa vào liên kết VÀ (**And-Introduction**)

$$\frac{p_1, p_2, \ldots, p_n}{p_1 \wedge p_2 \wedge \ldots \wedge p_n}$$

- Luật suy diễn đưa vào liên kết HOẶC (**Or-Introduction**)

$$\frac{p_i}{p_1 \vee p_2 \vee \ldots \vee p_i \vee \ldots \vee p_n}$$

Chứng minh bằng các luật suy diễn (2)

- Luật suy diễn loại bỏ phủ định hai lần (**Elimination of Double Negation**)

$$\frac{\neg\neg p}{p}$$

- Luật suy diễn hợp giải (**Resolution**)

$$\frac{p \vee q, \quad \neg q \vee r}{p \vee r}$$

- Luật suy diễn hợp giải đơn (**Unit Resolution**)

$$\frac{p \vee q, \quad \neg q}{p}$$

- Tất cả các luật suy diễn trên đều có tính *đúng đắn (sound)*!

Chứng minh bằng luật suy diễn: Ví dụ (1)

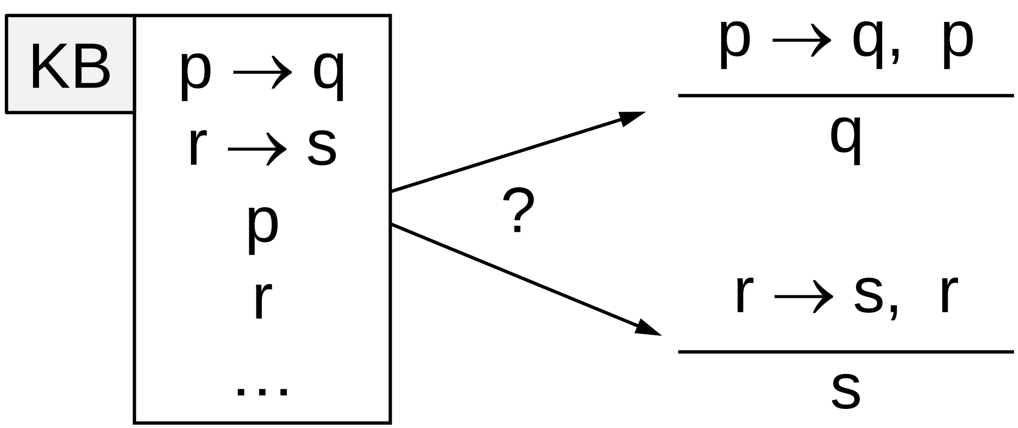
- Giả sử có tập giả thiết KB
  1) $p \land q$
  2) $p \rightarrow r$
  3) $(q \land r) \rightarrow s$
- Cần chứng minh định lý *s*
- Từ 1) và sử dụng luật And-Elimination, ta có:
  4) p
- Từ 2), 4), và sử dụng luật Modus Ponens, ta có:
  5) r

Chứng minh bằng luật suy diễn: Ví dụ (2)

- …
- Từ 1), và sử dụng luật And-Elimination, ta có:
  6) q
- Từ 5), 6), và sử dụng luật And-Introduction, ta có:
  7) $(q \wedge r)$
- Từ 7), 3), và sử dụng luật Modus-Ponens, ta có:
  8) s
- Vậy định lý (biểu thức logic) *s* được chứng minh là đúng!

Suy diễn logic và Tìm kiếm

- Để chứng minh định lý α là đúng đối với tập giả thiết *KB*, cần áp dụng một chuỗi các luật suy diễn đúng đắn
- Vấn đề: Ở mỗi bước suy diễn, có nhiều luật có thể áp dụng được
  - Chọn luật nào để áp dụng tiếp theo?
- Đây là vấn đề của bài toán tìm kiếm (search)



Chuyển đổi các biểu thức logic

- Trong logic định đề
  - Một biểu thức có thể bao gồm nhiều liên kết: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
  - Một biểu thức có thể bao gồm nhiều biểu thức con (lồng) khác
- Chúng ta có cần sử dụng tất cả các liên kết logic để biểu diễn một biểu thức phức tạp?
  - Không.
  - Chúng ta có thể viết lại (chuyển đổi) một biểu thức logic định đề thành một biểu thức tương đương *chỉ chứa các liên kết* $\neg, \wedge, \vee$

Các dạng chuẩn

- Các biểu thức trong logic định đề có thể được chuyển đổi về một trong các dạng chuẩn (Normal forms)
  - Giúp đơn giản hóa quá trình suy diễn
- **Dạng chuẩn kết hợp** (Conjunctive normal form – CNF)
  - Là kết hợp (liên kết VÀ) của các mệnh đề (clauses)
  - Mỗi mệnh đề (clause) là một liên kết HOẶC của các ký hiệu định đề đơn
  - Ví dụ: $(p \lor q) \land (\neg q \lor \neg r \lor s)$
- **Dạng chuẩn tuyển** (Disjunctive normal form – DNF)
  - Là liên kết HOẶC của các mệnh đề (clauses)
  - Mỗi mệnh đề (clause) là một liên kết VÀ của các ký hiệu định đề đơn
  - Ví dụ: $(p \land \neg q) \lor (\neg p \land r) \lor (r \land \neg s)$

Dạng chuẩn Horn

- Một biểu thức logic ở dạng chuẩn Horn nếu:
  - Biểu thức đó là một liên kết VÀ của các mệnh đề
  - Mỗi mệnh đề là một liên kết HOẶC các ký hiệu (literals), và có tối đa là 1 ký hiệu khẳng định (positive literal)
  - Ví dụ: $(p \vee \neg q) \wedge (\neg p \vee \neg r \vee s)$
- Không phải mọi biểu thức logic định đề đều có thể được chuyển về dạng chuẩn Horn!
- Biểu diễn tập giả thiết KB ở dạng chuẩn Horn
  - **Các luật (Rules)**
    - $(\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q)$
    - Tương đương với luật: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q)$
  - **Các sự kiện (Facts)**
    - p, q
  - **Các ràng buộc toàn vẹn (Integrity constraints)**
    - $(\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n)$
    - Tương đương với luật: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow \textit{sai})$

Chuyển đổi sang CNF – Ví dụ

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Loại bỏ phép $\Leftrightarrow$, thay $\alpha \Leftrightarrow \beta$ bằng $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Loại bỏ phép $\Rightarrow$, thay $\alpha \Rightarrow \beta$ bằng $\neg\alpha \vee \beta$.

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Đưa $\neg$ vào trong sử dụng luật de Morgan và phủ định kép:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Áp dụng luật phân phối đối với phép $\wedge$:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Ví dụ

$(A \vee B) \rightarrow (C \rightarrow D)$

1. Loại bỏ phép suy ra

   $\neg(A \vee B) \vee (\neg C \vee D)$

2. Chuyển phủ định vào trong ngoặc

   $(\neg A \wedge \neg B) \vee (\neg C \vee D)$

3. Phân phối

   $(\neg A \vee \neg C \vee D) \wedge (\neg B \vee \neg C \vee D)$

Bài tập

Chuyển đổi các công thức sau về dạng kết nối chuẩn:

1. $P \vee (\neg P \wedge Q \wedge R)$
2. $(\neg P \wedge Q) \vee (P \wedge \neg Q)$
3. $\neg(P \Rightarrow Q) \vee (P \vee Q)$
4. $(P \Rightarrow Q) \Rightarrow R$
5. $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \wedge S) \Rightarrow R)$
6. $(P \wedge (Q \Rightarrow R)) \Rightarrow S$
7. $P \wedge Q \Rightarrow R \wedge S$

Bài toán chứng minh thỏa mãn (SAT)

- Mục đích của bài toán chứng minh thỏa mãn (Satisfiability – SAT) là xác định một biểu thức ở dạng chuẩn kết hợp (CNF) có thể thỏa mãn được hay không
  - Tức là chứng minh biểu thức đó là đúng hay không
  - Ví dụ: $(p \lor q \lor \neg r) \land (\neg p \lor \neg r \lor s) \land (\neg p \lor q \lor \neg t)$

- Đây là một trường hợp của bài toán thỏa mãn ràng buộc (CSP)
  - Tập các biến
    - Các ký hiệu định đề (ví dụ: p, q, r, s, t)
    - Các giá trị (hằng) logic *đúng*, *sai*
  - Tập các ràng buộc
    - Tất cả các mệnh đề (được liên kết bởi phép VÀ) trong biểu thức phải đúng
    - Với mỗi mệnh đề, ít nhất một trong các định đề đơn phải đúng

Giải quyết bài toán SAT

- Phương pháp **Backtracking**
  - Áp dụng chiến lược tìm kiếm theo chiều sâu (Depth-first search)
  - Xét một biến (một định đề đơn), xét các khả năng gán giá trị (đúng/sai) cho biến đó
  - Lặp lại, cho đến khi tất cả các biến được gán giá trị, hoặc việc gán giá trị cho tập con của tập tất cả các biến, làm cho **biểu thức là sai**
- Các phương pháp **tối ưu hóa lặp (Iterative optimization methods)**
  - Bắt đầu với một phép gán ngẫu nhiên các giá trị đúng/sai cho các ký hiệu định đề
  - Đổi giá trị (*đúng* thành *sai* / *sai* thành *đúng*) đối với một biến
  - Heuristic: ưu tiên các phép gán giá trị làm cho nhiều mệnh đề (hơn) đúng
  - Sử dụng các phương pháp tìm kiếm cục bộ: Simulated Annealing, Walk-SAT

Bài toán suy diễn vs. Bài toán thỏa mãn được

- **Bài toán suy diễn logic**
  - Cần chứng minh: biểu thức logic (định lý) α được bao hàm bởi tập các mệnh đề *KB*
  - Nói cách khác: với mọi phép diễn giải mà trong đó *KB* đúng, thì α có đúng?
- **Bài toán thỏa mãn được (SAT)**
  - Có tồn tại một phép gán giá trị đúng/sai cho các ký hiệu định đề (một phép diễn giải) sao cho biểu thức α là đúng?
- Mối quan hệ:

KB $\models \alpha$	nếu và chỉ nếu:
(KB $\wedge \neg\alpha$)	là **không thể thỏa mãn được**
(unsatisfiable)	

Luật suy diễn hợp giải (1)

- Luật suy diễn **hợp giải (Resolution)**

$$\frac{p \vee q, \quad \neg q \vee r}{p \vee r}$$

- Luật suy diễn hợp giải áp dụng được đối với các biểu thức logic ở dạng chuẩn CNF
- Luật suy diễn hợp giải có tính *đúng đắn (sound)*, nhưng *không có tính hoàn chỉnh (incomplete)*
  - Tập giả thiết (cơ sở tri thức) *KB* chứa biểu thức $(p \wedge q)$
  - Cần chứng minh: $(p \vee q)$?
  - Luật suy diễn hợp giải không thể suy ra được biểu thức cần chứng minh!

Luật suy diễn hợp giải (2)

- Chuyển bài toán chứng minh logic về bài toán SAT
  - Phương pháp chứng minh bằng phản chứng
  - Việc chứng minh sự mâu thuẫn của: $(KB \wedge \neg\alpha)$
  - Tương đương việc chứng minh sự bao hàm: $KB \models \alpha$
- Luật suy diễn hợp giải (Resolution rule)
  - Nếu các biểu thức trong tập *KB* và biểu thức $\neg\alpha$ đều ở dạng CNF, thì áp dụng luật suy diễn hợp giải sẽ xác định tính (không) thỏa mãn được của $(KB \wedge \neg\alpha)$

Giải thuật hợp giải của Robinson

Chứng minh bằng phản chứng: CSTT $\wedge\neg$KL không thoả mãn

```
function PL-RESOLUTION(KB, α) returns true or false
    clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
    new ← { }
    loop do
        for each Ci, Cj in clauses do
            resolvents ← PL-RESOLVE(Ci, Cj)
            if resolvents contains the empty clause then return true
            new ← new ∪ resolvents
        if new ⊆ clauses then return false
        clauses ← clauses ∪ new
```

Giải thuật hợp giải của Robinson

Chứng minh bằng phản chứng: CSTT $\wedge\neg$KL không thoả mãn

Giả sử có GT_1, GT_2,…,GT_n. Cần CM KL → phản chứng

$$\left.\begin{array}{l} GT_1 \\ \ldots \\ GT_n \\ \neg KL \end{array}\right\} \quad ><$$

Viết mỗi GT_i, $\neg$KL trên 1 dòng

Đưa GT_i, $\neg$KL về dạng chuẩn CNF

$(p_1\vee\ldots\vee p_n) \wedge (q_1\vee\ldots\vee q_n)$ (*)

Tách mỗi dòng (*) thành các dòng con:

$p_1\vee\ldots\vee p_n$

$q_1\vee\ldots\vee q_n$

Giải thuật hợp giải của Robinson

Xét 1 cặp dòng

u) $\neg p \vee q$

v) $p \vee r$

Hợp giải:

w) $q \vee r$

Vô lý xuất hiện khi

i) $\neg t$

ii) t

$\Rightarrow$ đpcm

Ví dụ

VD1:

1. a
2. $a \rightarrow b$
3. $b \rightarrow (c \rightarrow d)$
4. c

Chứng minh d

VD2:

1. $a \wedge b \rightarrow c$
2. $b \wedge c \rightarrow d$
3. a
4. b

Chứng minh d

Ví dụ

VD3:

1. p
2. $p \rightarrow q$
3. $q \wedge r \wedge s \rightarrow t$
4. $p \rightarrow u$
5. $v \rightarrow w$
6. $u \rightarrow v$
7. $v \rightarrow t$

Cho r,s. CM t

VD4:

1. $((a \vee b) \wedge c) \rightarrow (c \wedge d)$
2. $a \wedge m \wedge d \rightarrow f$
3. $m \rightarrow b \wedge c$
4. $a \rightarrow c$
5. $(a \wedge f) \rightarrow (\neg e \vee g)$
6. $(m \wedge f) \rightarrow g$

Cho a,m. CM g

Ví dụ

1. $a1 \vee a2 \Rightarrow a3 \vee a4$
2. $a1 \Rightarrow a5$
3. $a2 \wedge a3 \Rightarrow a5$
4. $a2 \wedge a4 \Rightarrow a6 \wedge a7$
5. $a5 \Rightarrow a7$
6. $a1 \wedge a3 \Rightarrow a6 \vee a7$

- Cho các mệnh đề a1, a2 đúng.
- Đưa các biểu thức logic trên về dạng chuẩn
- áp dụng phương pháp hợp giải của Robinson, chứng minh a7 đúng.

Nhận xét

- Thuật giải Robinson vẫn vấp phải sự bùng nổ tổ hợp. Có thể áp dụng các heuristics:
  - Chiến lược ưu tiên các biểu thức đơn
  - Chiến lược đơn giản hóa các biểu thức
  - Chiến lược giảm số lần hợp giải
  - Chiến lược sắp thứ tự các hợp giải
  - Chiến lược tập tựa
- Thuật giải Robinson được áp dụng trong CM định lý tự động. 2 nhược điểm:
  - con người không tư duy theo cách này
  - chúng ta bị mất ngữ nghĩa và nội dung thông tin khi chuyển về dạng câu CNF

Luật suy diễn Modus Ponens tổng quát

$$\frac{(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q),\ p_1,\ p_2,\ \ldots,\ p_n}{q}$$

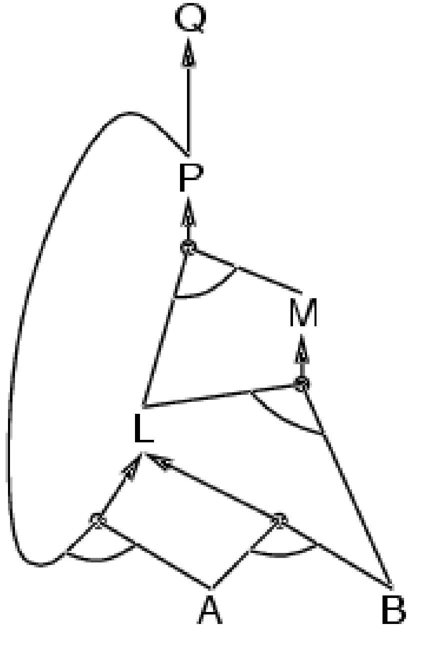
- Luật suy diễn Modus Ponens có tính *đúng đắn (sound)* và *hoàn chỉnh (complete)*, đối với các ký hiệu định đề và đối với tập các biểu thức KB ở dạng chuẩn Horn

- Luật suy diễn Modus Ponens có thể được sử dụng với cả 2 chiến lược suy diễn (suy diễn tiến và suy diễn lùi)

Suy diễn tiến (forward chaining)

- Với một tập các mệnh đề giả thiết (cơ sở tri thức) *KB*, cần suy ra mệnh đề kết luận *Q*
- Ý tưởng: Lặp lại 2 bước sau cho đến khi suy ra được kết luận
  - Áp dụng các luật có mệnh đề giả thiết được thỏa mãn trong *KB*
  - Bổ sung kết luận của các luật đó vào *KB*

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Suy diễn tiến: Ví dụ (1)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

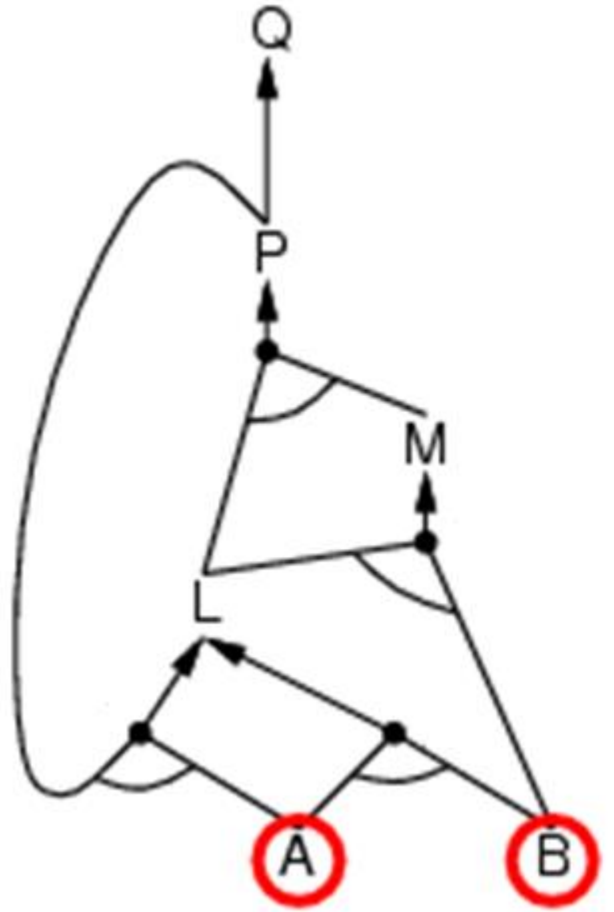
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Suy diễn tiến: Ví dụ (2)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

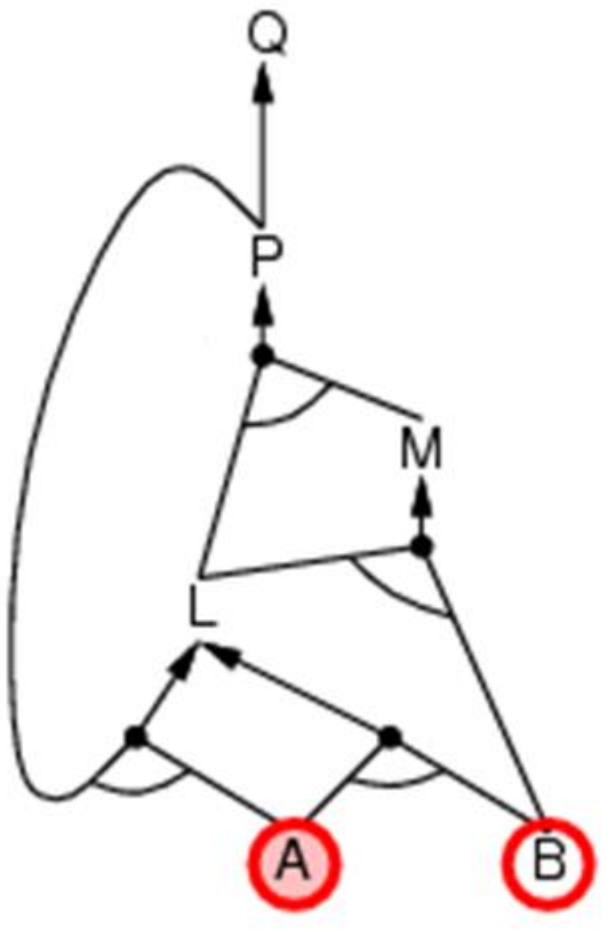
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

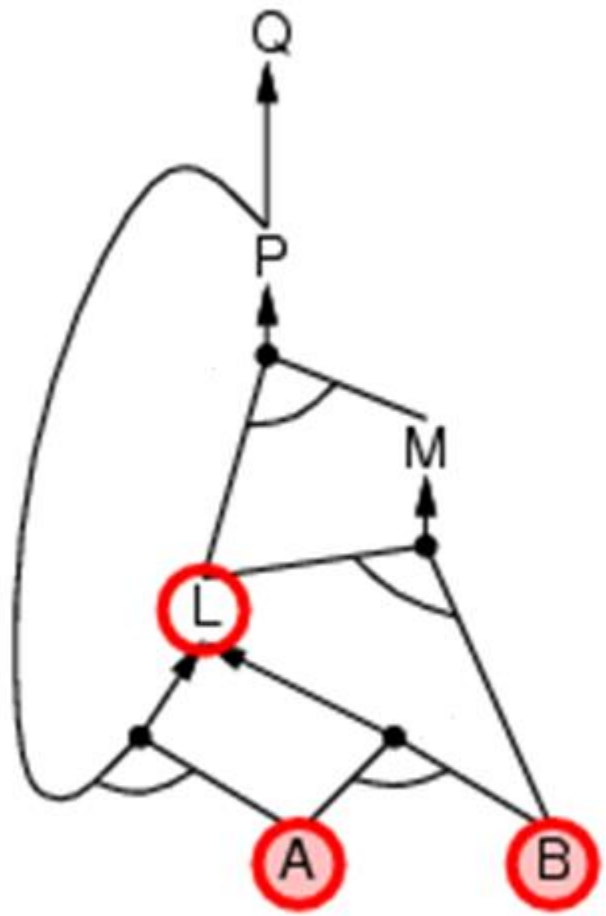
A

B



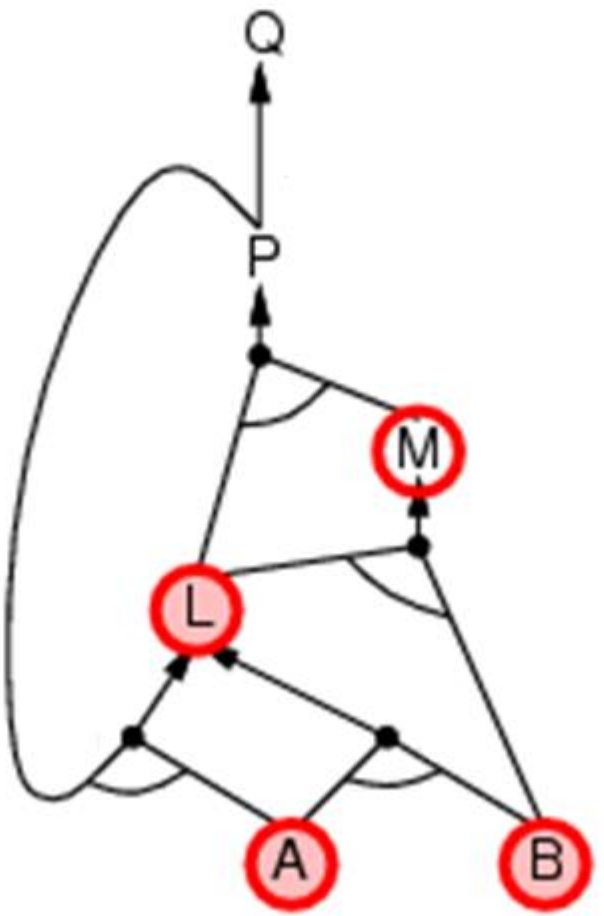
Suy diễn tiến: Ví dụ (3)

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



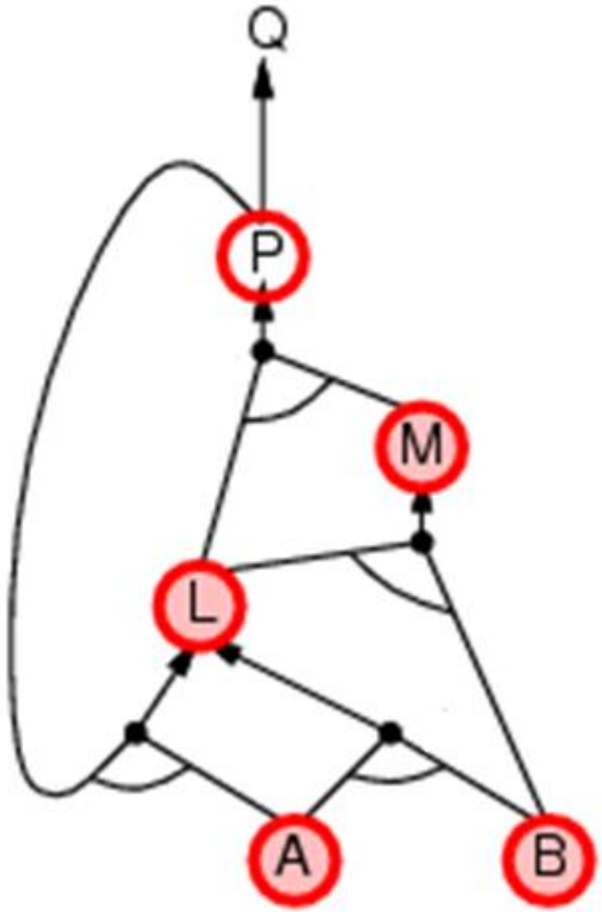
Suy diễn tiến: Ví dụ (4)

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Suy diễn tiến: Ví dụ (5)

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Suy diễn tiến: Ví dụ (6)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

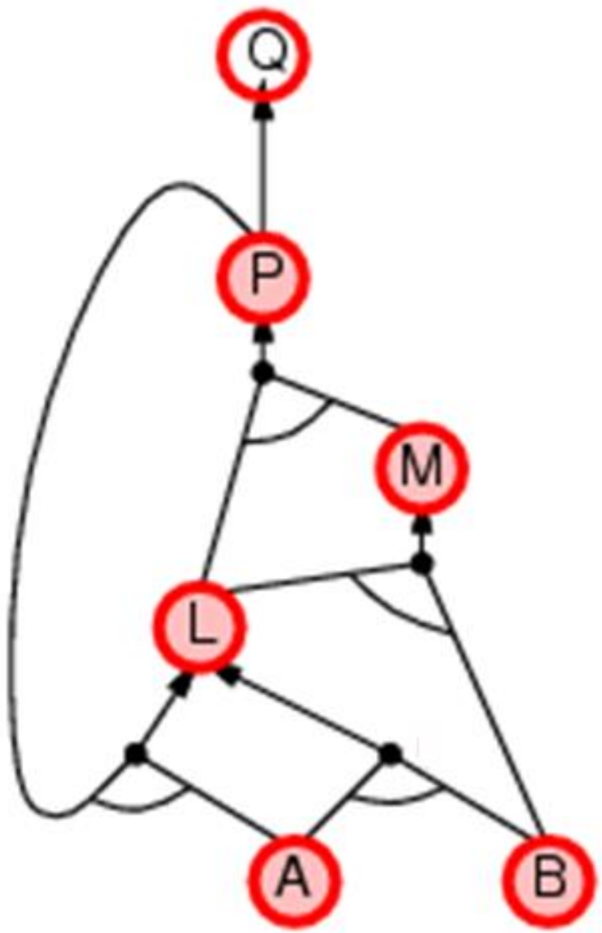
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Suy diễn tiến: Ví dụ (7)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

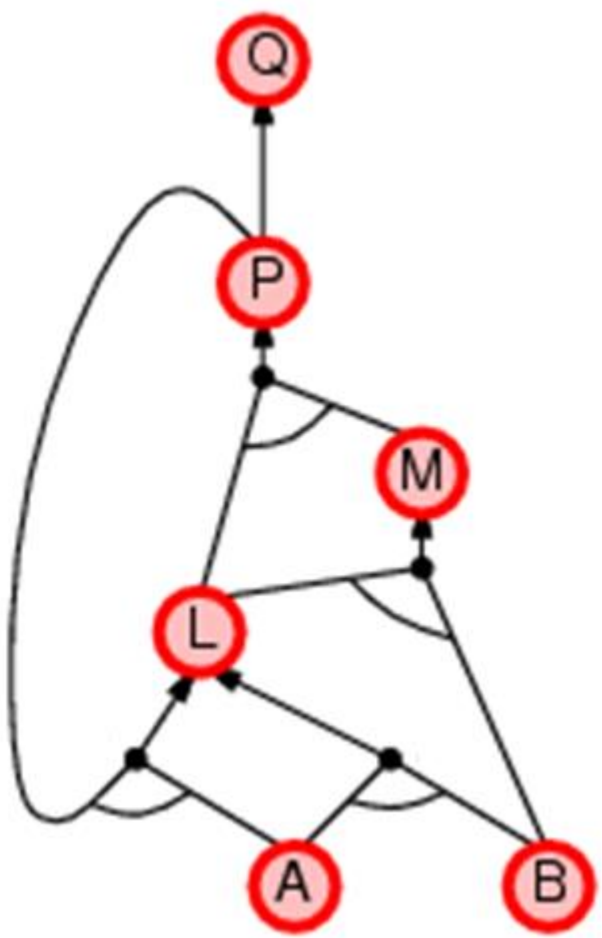
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Ví dụ

VD1. Cho GT = {a,b,m_a}. Tìm KL ={h_c}

1. a,b,m_a $\rightarrow$c
2. a,b,c $\rightarrow$ A
3. b,A $\rightarrow$ h_c
4. a,b,c $\rightarrow$ B
5. a,b,c $\rightarrow$C
6. a,B $\rightarrow$$h_c$
7. A,B $\rightarrow$C
8. B,C $\rightarrow$A
9. A,C $\rightarrow$B

Bài tập tại lớp

So sánh stack và queue

BT1. Cho GT={a }, KL={u}

1. $a \rightarrow b$
2. $b \rightarrow c$
3. $c \rightarrow d$
4. $a \rightarrow u$

BT2. GT={a}, KL={u}

1. $a \rightarrow b$
2. $d \rightarrow c$
3. $c \rightarrow u$
4. $a \rightarrow m$
5. $b \rightarrow n$
6. $m \rightarrow p$
7. $p \rightarrow q$
8. $q \rightarrow u$

Thuật toán

Vào:

- Tập các mệnh đề/vị từ đã cho (ở dạng chuẩn Horn)
- Tập các luật RULE dạng p→q
- Tập các mệnh đề/vị từ kết luận KL

Ra:

- Thông báo "Thành công" nếu KL có thể suy ra từ GT

PP: /*Tgian là tập các mệnh đề/vị từ đúng cho đến thời điểm đang xét*/

Thuật toán

```
{1 Tgian = GT;
   Thoa = Loc(Tgian,R);
   while Thoa <>0 and KL∉Tgian do
   {2    r ← get(Thoa); /* r: left → q  */
         R = R \ {r}; Vet = Vet ∪ {r};
         Tgian = Tgian ∪ {q};
         Thoa = Loc(Tgian,R)
   }2
   if KL ⊆ Tgian then exit("Thành công")
   else   exit("Không thành công")
}1
```

Suy diễn lùi (backward chaining)

Ý tưởng: suy diễn lùi từ kết luận KL

- kiểm tra xem KL đã được biết chưa, nếu không
- chứng minh bằng quay lui sử dụng các luật dẫn đến q

- Tránh lặp vô tận:
  - lưu trữ các đích đã được chứng minh
  - trước khi chứng minh kiểm tra xem đích cần chứng minh đã có trong goal stack chưa?

- Tránh lặp lại công việc: kiểm tra xem KL mới
  - đã ở trong tập đã được chứng minh chưa
  - đã làm nhưng thất bại chưa

Suy diễn lùi: Ví dụ (1)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

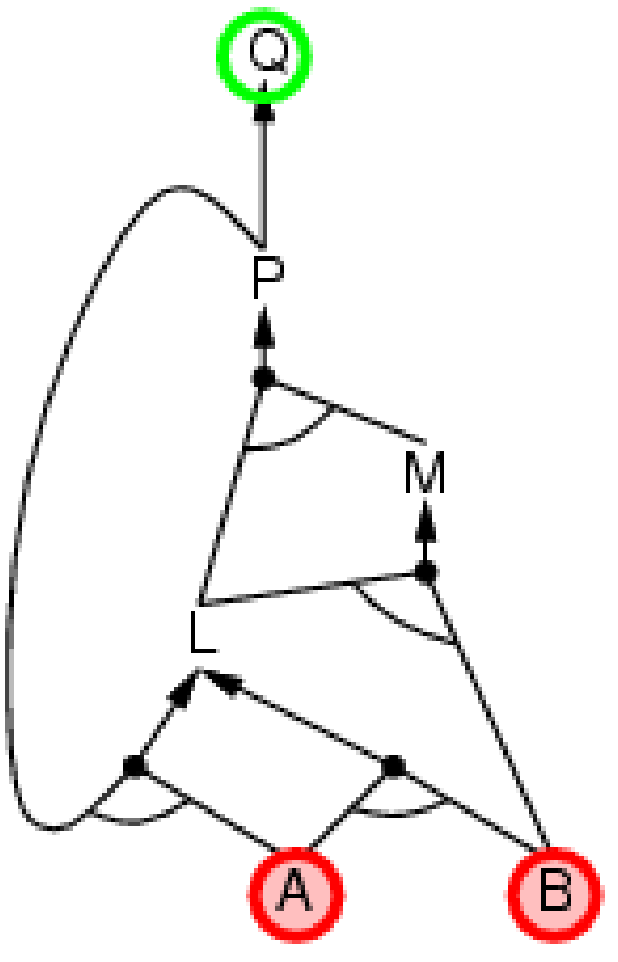
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

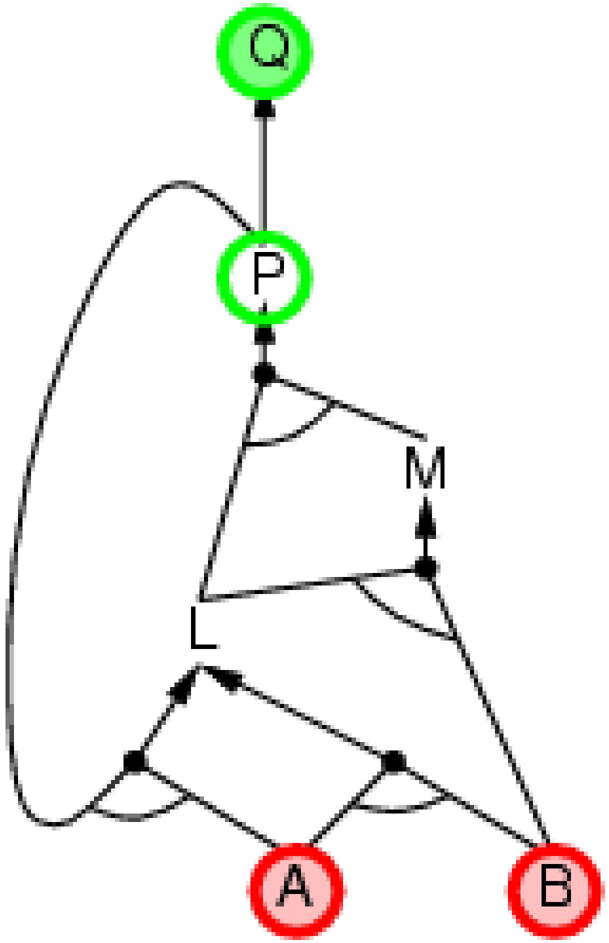
A

B



Suy diễn lùi: Ví dụ (2)

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Suy diễn lùi: Ví dụ (3)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

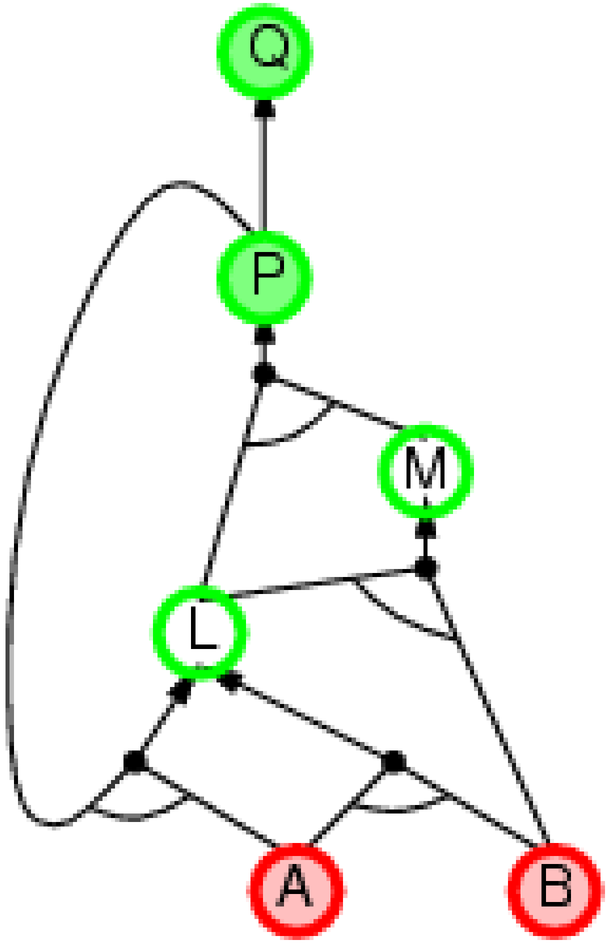
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Suy diễn lùi: Ví dụ (4)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

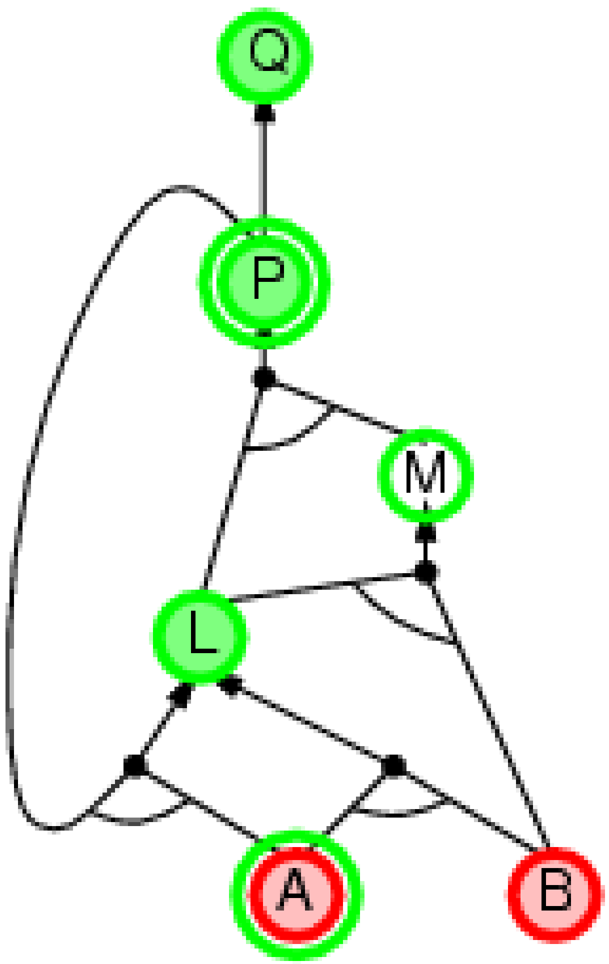
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Suy diễn lùi: Ví dụ (5)

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

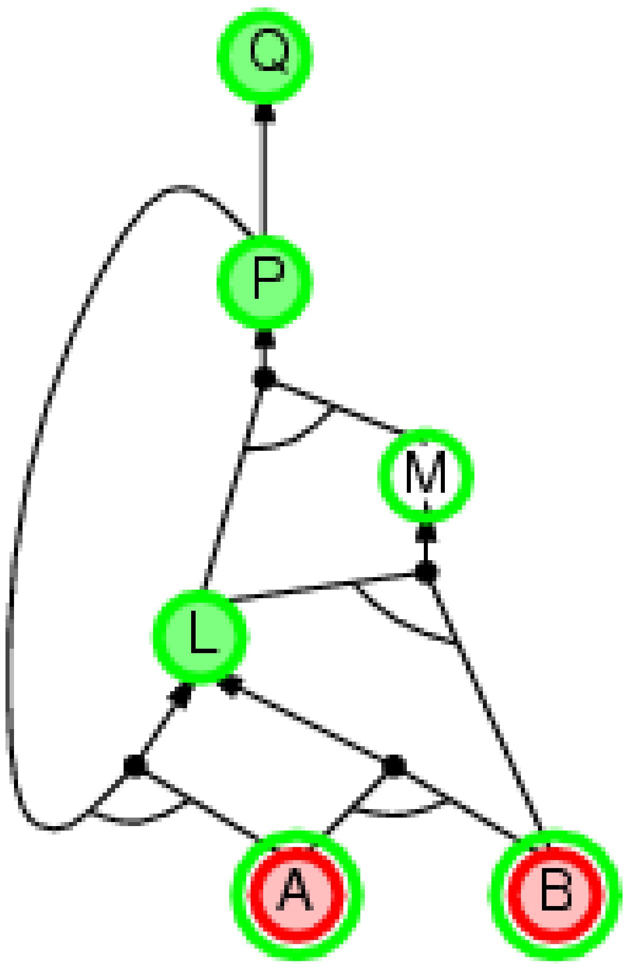
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Suy diễn lùi

VD:

1. $A,C \rightarrow B$
2. $a,b,m_a \rightarrow c$
3. $a,b,c \rightarrow A$
4. $a,b,c \rightarrow B$
5. $a,b,c \rightarrow C$
6. $a,B \rightarrow h_c$
7. $b,A \rightarrow h_c$
8. $c,S \rightarrow h_c$
9. $a,b,c \rightarrow S$

1'. $h_a,c \rightarrow B$

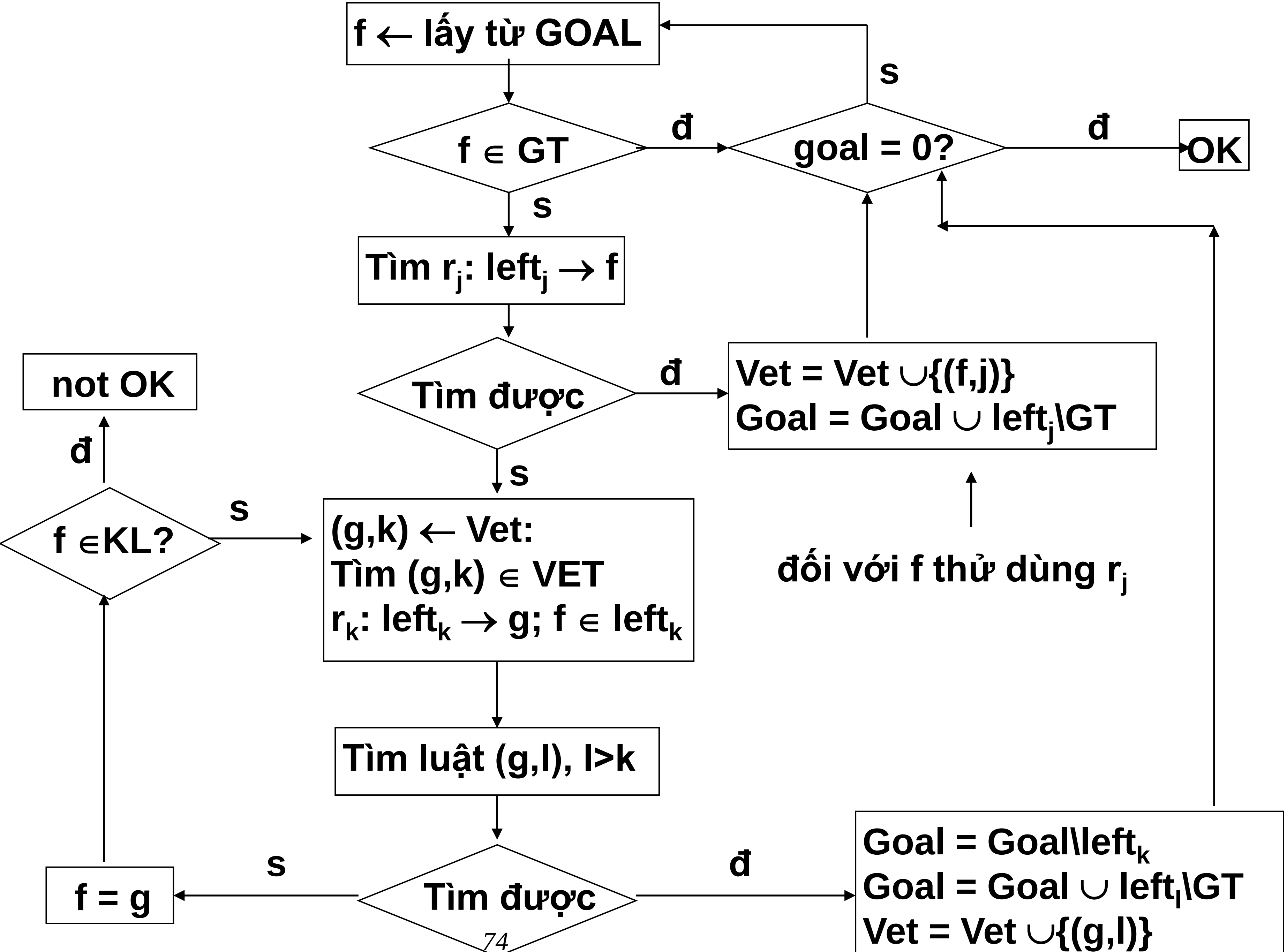
$GT=\{a,b,m_a\}$; $KL=\{h_c\}$

Suy diễn lùi

Đầu:

- Goal = tập các sự kiện cần CM=KL
- Goal = {f| f cần CM cho đến thời điểm hiện tại}
- Vet ={(f,j)| để CM f thì dùng luật j: $left_j \rightarrow f$}
- Cờ Back = true khi quay lui

  false không quay lui


f ← lấy từ GOAL
f ∈ GT
đ
goal = 0?
s
đ
OK
s
Tìm r_j: $left_j \rightarrow f$
Tìm được
đ
Vet = Vet ∪{(f,j)}
Goal = Goal ∪ $left_j$\GT
not OK
đ
s
f ∈KL?
s
(g,k) ← Vet:
Tìm (g,k) ∈ VET
r_k: $left_k \rightarrow g$; f ∈ $left_k$
đối với f thử dùng r_j
Tìm luật (g,l), l>k
s
đ
f = g
Tìm được
Goal = Goal\$left_k$
Goal = Goal ∪ $left_l$\GT
Vet = Vet ∪{(g,l)}

Suy diễn lùi

1. Quá trình SD lùi tương tự quá trình tìm cây/đồ thị lời giải trong đồ thị V/H
2. Để tăng hiệu quả của thủ tục SDL, có thể đưa vào 2 tập:
   - Tập Đúng chứa các sự kiện đã được khẳng định là đúng (đã xác định)
   - Tập Sai chứa các sự kiện đã được khẳng định là sai (không thể xác định)

Bài tập tại lớp

BT1. Cho GT={a,b,m_a}, KL={h_c}

1. a,b,$m_a \rightarrow$ c
2. a,b,C $\rightarrow$ s
3. a,s $\rightarrow h_a$
4. b,s $\rightarrow h_b$
5. c,s $\rightarrow h_c$
6. a,B $\rightarrow h_c$
7. a,b,c $\rightarrow$ B

BT2. GT={a}, KL={u}

1. a $\rightarrow$ b
2. d $\rightarrow$ c
3. c $\rightarrow$ u
4. a $\rightarrow$ m
5. b $\rightarrow$ n
6. m $\rightarrow$ p
7. p $\rightarrow$ q
8. q $\rightarrow$ u

So sánh SD tiến và SD lùi

- SD tiến hướng dữ liệu, tự động, không định hướng. Ví dụ, nhận dạng đối tượng, xác định hành trình
- Có thể làm rất nhiều việc không liên quan đến KL

- SD lùi hướng KL, thích hợp cho các bài toán giải quyết vấn đề. Ví dụ, tìm chìa khoá, lập kế hoạch thi TOEFL
- Độ phức tạo của SD lùi thường nhỏ hơn rất nhiều so với kích thước của CSTT.

Logic định đề: Ưu và nhược điểm

- (+) Logic định đề cho phép dễ dàng phát biểu (biểu diễn) cơ sở tri thức bằng tập các mệnh đề
- (+) Logic định đề cho phép làm việc với các thông tin ở dạng phủ định, dạng tuyển (disjunctive)
- (+) Logic định đề có tính cấu tạo (kết cấu)
  - Ngữ nghĩa của mệnh đề ($S_1 \land S_2$) được suy ra từ ngữ nghĩa của S_1 và ngữ nghĩa của S_2
- (+) Ngữ nghĩa trong logic định đề không phụ thuộc ngữ cảnh (context-independent)
  - Không như trong ngôn ngữ tự nhiên (ngữ nghĩa phụ thuộc vào ngữ cảnh của các câu nói)
- (-) Khả năng diễn đạt (biểu diễn) của logic định đề là rất hạn chế
  - Logic định đề không thể diễn đạt được (như trong ngôn ngữ tự nhiên): "Nếu X là cha của Y, thì Y là con của X"
  - Logic định đề phải liệt kê (xét) mọi khả năng gán giá trị chân lý (đúng/sai) cho X và Y

Giới hạn của Logic định đề

- Hãy xét ví dụ sau đây:
  - Tuấn là một sinh viên của HUST
  - Mọi sinh viên của HUST đều học môn Đại số
  - Vì Tuấn là một sinh viên của HUST, nên Tuấn học môn Đại số
- Trong logic định đề:
  - Định đề *p*: "Tuấn là một sinh viên của HUST"
  - Định đề *q*: "Mọi sinh viên của HUST đều học môn Đại số"
  - Định đề *r*: "Tuấn học môn Đại số"
  - Nhưng: (trong logic định đề) *r* không thể suy ra được từ *p* và *q*!