

Trí tuệ nhân tạo (Artificial Intelligence)

Tác tử logic (logical agents)

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Vinh, 3/2019

Thông tin môn học

■ Tài liệu chính

[1] Stuart Russell, Peter Norvig. Artificial Intelligence. A modern approach, 3rd ed, Prentice Hall, 2009.

■ Tài liệu khác

[2] Milos Hauskrecht, Artificial Intelligence, 2013,
people.cs.pitt.edu/~milos/courses/cs1571-Fall2013

[3] Các nguồn từ internet

Nội dung

- Tác tử dựa trên tri thức
- Logic
- Logic mệnh đề
- Chứng minh logic mệnh đề

Tác tử dựa trên tri thức

- Thành phần trung tâm của một tác tử dựa trên tri thức (knowledge-based agent) là cơ sở tri thức của nó.
- Một cơ sở tri thức (knowledge base – *KB*) là một tập các câu (sentences) trong một ngôn ngữ hình thức (formal language).
- Cần có các thủ tục để thêm một câu mới vào cơ sở tri thức và truy vấn các câu từ cơ sở tri thức – gọi chung là thủ tục suy diễn (inference).
- Suy diễn (inference) là quá trình suy luận để nhận các câu mới từ các câu đã có trong *KB*.

Tác tử dựa trên tri thức

■ Chương trình tác tử dựa trên tri thức:

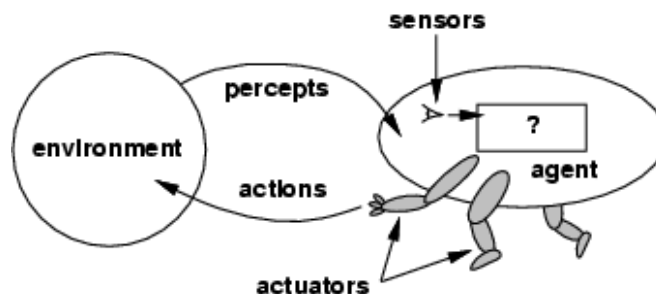
```
function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
               t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action
```

- MAKE-PERCEPT-SENTENCE: tạo một câu từ nhận thức ở thời gian *t*.
- MAKE-ACTION-QUERY: tạo một hành động từ một câu đã tạo ra.
- MAKE-ACTION-SENTENCE: tạo một câu tương ứng với hành động.
- TELL: thủ tục thêm một câu vào cơ sở tri thức.
- ASK: thủ tục truy vấn một câu từ cơ sở tri thức.

Tác tử dựa trên tri thức

- Xây dựng một tác tử dựa trên tri thức
 - Thêm từng câu cho đến khi tác tử biết toàn bộ môi trường làm việc (declarative approach).
 - Cho phép tác tử học tri thức dựa trên các tri thức đã có (learning approach).



Logic

- Một cơ sở tri thức(KB) là một tập các câu (sentences)
Câu (sentence) = cú pháp (syntax) + ngữ nghĩa (semantics)
- Cú pháp là quy tắc xây dựng các câu.
 - Ví dụ trong toán học " $x + y = 4$ " là một câu nhưng " $x4y+ =$ " không phải là một câu.
- Ngữ nghĩa là nghĩa của câu và chỉ nhận một trong hai giá trị: hoặc là "true" (đúng) hoặc là "false" (sai).
 - Ngữ nghĩa của " $x + y = 4$ " là đúng khi $x = 2$ và $y = 2$ nhưng sai khi $x = 1$ và $y = 1$.
 - 2 là số nguyên tố: ngữ nghĩa là đúng.
 - $x + 2 \geq y$ là đúng khi $x = 7, y = 1$ và sai khi $x = 0, y = 5$.

Logic

- Một diễn giải (interpretation) là một **mô hình m** của một câu α nếu α đúng trong diễn giải đó.
 - Ví dụ có x nam và y nữ ngồi chơi bài và $\alpha = x + y$. Câu $\alpha = 4$ có nghĩa là "tổng người chơi bài là 4". Diễn giải $(x = 2, y = 2)$ là một mô hình của α .
- Ký hiệu $M(\alpha)$ là tập tất cả các mô hình của α .
 - Ví dụ $M(\alpha) = \{(x = 0, y = 4), (x = 1, y = 3), (x = 2, y = 2), (x = 3, y = 1), (x = 4, y = 0)\}$.
- Một câu α kéo theo một câu β , ký hiệu $\alpha \models \beta$, nếu và chỉ nếu trong mọi mô hình trong đó α đúng thì β cũng đúng, tức là $\alpha \models \beta$ nếu và chỉ nếu $M(\alpha) \subseteq M(\beta)$.
 - Ví dụ $\beta = y + x$ thì $\alpha \models \beta$ vì $M(\alpha) = M(\beta)$.

Logic

- $KB \models \alpha$ nếu và chỉ nếu $M(KB) \subseteq M(\alpha)$, tức là trong mọi mô hình mà các câu trong KB đúng thì câu α cũng đúng.
- Một thuật toán suy diễn i nhận một câu α từ KB , ký hiệu: $KB \vdash_i \alpha$.
 - Nói cách khác, “ α nhận được từ KB bởi thuật toán i ”.
- Tính đúng đắn (soundness)
 - Thuật toán i là đúng đắn nếu bất cứ khi nào $KB \vdash_i \alpha$, thì cũng đúng với $KB \models \alpha$.
- Tính hoàn chỉnh (completeness)
 - Thuật toán i là hoàn chỉnh nếu bất cứ khi nào $KB \models \alpha$ thì cũng đúng với $KB \vdash_i \alpha$.

Logic

- Nếu KB đúng trong thế giới thực (real world) thì mọi câu α nhận được từ KB bởi một thuật toán suy diễn đúng đắn cũng đúng trong thế giới thực.
- Sự tương ứng giữa thế giới thực và biểu diễn.

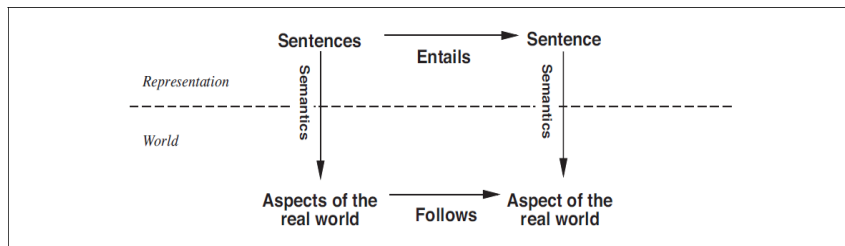


Figure 7.6 Sentences are physical configurations of the agent, and reasoning is a process of constructing new physical configurations from old ones. Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

Logic mệnh đề

- Logic mệnh đề (propositional logic) là một loại logic đơn giản nhất.

Câu = Cú pháp + Ngữ nghĩa

Sentence = Syntax + Semantics

- Câu là một mệnh đề/tuyên bố (statement) có giá trị đúng (true) hoặc sai (false).
 - “ $5 + 2 = 7$ ”: mệnh đề có giá trị đúng.
 - “2 là số nguyên tố”: mệnh đề có giá trị đúng.
 - “It is raining outside”: câu có giá trị đúng hoặc sai phụ thuộc vào thời gian.

Logic mệnh đề: cú pháp (syntax)

- Câu nguyên tử (atomic sentences) gồm:
 - Hằng True (đúng), False (sai),
 - Hoặc các ký hiệu mệnh đề, thường ký hiệu bởi các chữ hoa.
- Câu phức hợp (complex sentences): Được xây dựng từ các câu đơn giản hơn thông qua các kết nối logic.
- Các kết nối logic (connectives):
 - \neg (not - phủ định)
 - \vee (or - tuyển),
 - \wedge (and - hội),
 - \Rightarrow (if then - nếu thì),
 - \Leftrightarrow (if and only if - nếu và chỉ nếu).

Logic mệnh đề: cú pháp (syntax)

■ Dạng chuẩn BNF của câu:

$$\begin{aligned}
 \text{Sentence} &\rightarrow \text{AtomicSentence} \mid \text{ComplexSentence} \\
 \text{AtomicSentence} &\rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \dots \\
 \text{ComplexSentence} &\rightarrow (\text{Sentence}) \mid [\text{Sentence}] \\
 &\mid \neg \text{Sentence} \\
 &\mid \text{Sentence} \wedge \text{Sentence} \\
 &\mid \text{Sentence} \vee \text{Sentence} \\
 &\mid \text{Sentence} \Rightarrow \text{Sentence} \\
 &\mid \text{Sentence} \Leftrightarrow \text{Sentence}
 \end{aligned}$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

Logic mệnh đề: ngữ nghĩa (semantics)

■ Ngữ nghĩa của logic mệnh đề được xác định bởi các câu nguyên tố và các kết nối logic.

■ Bảng chân lý:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	false	true	false	false	true	true
false	true	true	false	true	true	false
true	false	false	false	true	false	false
true	true	false	true	true	true	true

■ Ví dụ xác định ngữ nghĩa của câu $(P \vee Q) \wedge \neg S$

- Xác định ngữ nghĩa của câu $P \vee Q$
- Xác định ngữ nghĩa của câu $\neg S$
- Xác định ngữ nghĩa của câu $(P \vee Q) \wedge \neg S$

Xây dựng một cơ sở tri thức

- Biểu diễn các câu thành logic mệnh đề
 - Chỉ ra các câu nguyên tử là các mệnh đề.
 - Sử dụng các kết nối logic để biểu diễn các câu phức hợp.
- Giả sử biểu diễn câu sau:
 - It is not sunny this afternoon and it is colder than yesterday.
- Định nghĩa các câu nguyên tử:
 - p = It is sunny this afternoon.
 - q = it is colder than yesterday.
- Biểu diễn thành logic mệnh đề: $\neg p \wedge q$

Xây dựng một cơ sở tri thức

- Giả sử xây dựng KB cho các câu sau:
 - It is not sunny this afternoon and it is colder than yesterday.
 - We will go swimming only if it is sunny.
 - If we do not go swimming then we will take a canoe trip.
 - If we take a canoe trip, then we will be home by sunset.
- Ký hiệu:
 - p = It is sunny this afternoon.
 - q = it is colder than yesterday.
 - r = We will go swimming.
 - s = we will take a canoe trip.
 - t = We will be home by sunset.
- Biểu diễn KB = $\{\neg p \wedge q; p \Rightarrow r; \neg r \Rightarrow s; s \Rightarrow t\}$.

Chứng minh bài toán suy diễn logic

- Làm thế nào để thiết kế một thuật toán để trả lời câu hỏi:

$$KB \models \alpha ?$$

- Ba hướng tiếp cận:
 - Sử dụng bảng chân lý (truth – table)
 - Chứng minh bằng các luật suy diễn
 - Chứng minh bằng phản chứng.

Chứng minh bằng bảng chân lý

- Mục đích là trả lời câu hỏi $KB \models \alpha ?$, tức là trong mọi mô hình mà các câu trong KB đúng thì câu α đúng ?.
- Thuật toán đầu tiên cho suy diễn là hướng tiếp cận kiểm tra mô hình (model – checking approach):
 - Sinh ra các mô hình
 - Kiểm tra α đúng trong mọi mô hình mà KB đúng.
- Ví dụ cho $KB = \{P \vee Q; P \Leftrightarrow Q\}$ và $\alpha = (P \vee \neg Q) \wedge Q$.

$KB \models \alpha ?$		KB		α
P	Q	$P \vee Q$	$P \Leftrightarrow Q$	$(P \vee \neg Q) \wedge Q$
True	True	True	True	True
True	False	True	False	False
False	True	True	False	False
False	False	False	True	False



Chứng minh bằng bảng chân lý

```

function TT-ENTAILS?(KB, α) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
           α, the query, a sentence in propositional logic

  symbols ← a list of the proposition symbols in KB and α
  return TT-CHECK-ALL(KB, α, symbols, { })

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
  if EMPTY?(symbols) then
    if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
    else return true // when KB is false, always return true
  else do
    P ← FIRST(symbols)
    rest ← REST(symbols)
    return (TT-CHECK-ALL(KB, α, rest, model ∪ {P = true})
           and
           TT-CHECK-ALL(KB, α, rest, model ∪ {P = false}))

```

Figure 7.10 A truth-table enumeration algorithm for deciding propositional entailment. (TT stands for truth table.) PL-TRUE? returns *true* if a sentence holds within a model. The variable *model* represents a partial model—an assignment to some of the symbols. The keyword “and” is used here as a logical operation on its two arguments, returning *true* or *false*.

Chứng minh bằng bảng chân lý

- Ví dụ chứng minh $KB \models \alpha$ với KB và α như sau:

$$KB = (A \vee C) \wedge (B \vee \neg C) \quad \alpha = (A \vee B)$$

<i>A</i>	<i>B</i>	<i>C</i>	$A \vee C$	$(B \vee \neg C)$	<i>KB</i>	<i>α</i>
<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>
<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>False</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>False</i>
<i>False</i>	<i>False</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>

- Hướng tiếp cận chứng minh bằng bảng chân lý là **đúng đắn và hoàn chỉnh** nhưng độ phức tạp tính toán là 2^n .

Chứng minh bằng luật suy diễn

- Áp dụng các luật suy diễn trực tiếp cho các câu trong cơ sở tri thức để chứng minh một câu mong muốn mà không cần đến các mô hình.

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of \wedge
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of \vee
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of \wedge
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of \vee
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of \wedge over \vee
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of \vee over \wedge

Chứng minh bằng luật suy diễn

- Một số khái niệm:
 - Hai câu α và β là **tương đương** logic nếu chúng đều đúng trong cùng một tập các mô hình, ký hiệu $\alpha \equiv \beta$.
 - Nói cách khác, $\alpha \equiv \beta$ nếu và chỉ nếu $\alpha \models \beta$ và $\beta \models \alpha$.
 - Một câu là **đúng đắn (valid)** nếu nó đúng trong mọi mô hình, ví dụ $P \vee \neg P, Q \Rightarrow Q$.
 - Cho hai câu α và β , $\alpha \models \beta$ nếu và chỉ nếu $\alpha \Rightarrow \beta$ là đúng đắn, tức là $\alpha \Rightarrow \beta$ đúng trong mọi mô hình.
 - Một câu α là **thỏa mãn (satisfiable)** được nếu nó đúng trong một mô hình nào đó, ví dụ $P \vee Q, P \wedge Q$.

Chứng minh bằng luật suy diễn

- Một số khái niệm (tiếp):
 - Một câu α là **không thỏa mãn được (unsatisfiable)** nếu không tồn tại bất kỳ mô hình nào để câu α đúng, ví dụ $P \wedge \neg P$.
- Các hệ quả:
 - α là đúng dẫn nếu $\neg\alpha$ là không thỏa mãn được.
 - α là thỏa mãn được nếu $\neg\alpha$ là không đúng dẫn.
 - $\alpha \models \beta$ nếu và chỉ nếu câu $(\alpha \wedge \neg\beta)$ là không thỏa mãn được.
 - Chứng minh $\alpha \models \beta$ bằng cách chứng minh $(\alpha \wedge \neg\beta)$ gọi là **chứng minh phản chứng**.

Chứng minh bằng luật suy diễn

- Một số luật suy diễn:
 - **Modus Ponens**: $\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$ nếu $\alpha \Rightarrow \beta$ đúng và α đúng thì β đúng.
 - **Loại bỏ hội (And - elimination)**: $\frac{\alpha \wedge \beta}{\alpha}$ nếu $\alpha \wedge \beta$ đúng thì α đúng và β đúng.
 - **Đưa vào hội (And - introduction)**: $\frac{\alpha, \beta}{\alpha \wedge \beta}$ nếu α đúng và β đúng thì $\alpha \wedge \beta$ đúng.
 - **Luật đưa vào tuyển (Or – introduction)**: $\frac{\alpha}{\alpha \vee \beta}$ nếu α đúng thì $\alpha \vee \beta$ đúng.
 - **Luật phân giải (resolution)**: $\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$ nếu $\alpha \vee \beta$ và $\neg\beta \vee \gamma$ đúng thì $\alpha \vee \gamma$ đúng.

Chứng minh bằng luật suy diễn

- Ví dụ cho $KB = \{P \wedge Q; P \Rightarrow R; (Q \vee R) \Rightarrow S\}$ và $\alpha = S$.
Chứng minh $KB \models \alpha$.
- Đánh số các câu trong KB như sau:
(1) $P \wedge Q$; (2) $P \Rightarrow R$; (3) $(Q \vee R) \Rightarrow S$
 - Từ (1), áp dụng luật And-Elim., ta có:
(4) P
(5) Q
 - Từ (2) và (4), áp dụng luật Modus Ponens, ta có:
(6) R
 - Từ (5) và (6) áp dụng luật Or – Introduction ta có:
(7) $Q \vee R$
 - Từ (7) và (3) áp dụng luật Modus Ponens ta có S.

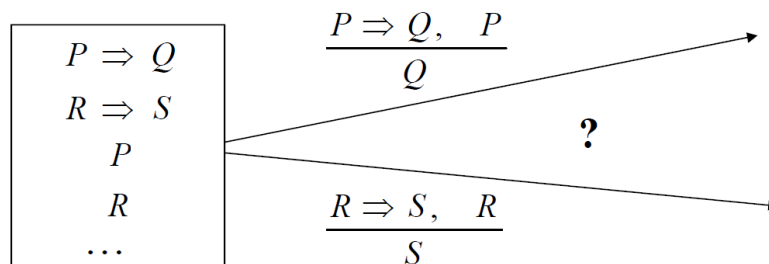
Bài tập

Sử dụng các luật suy diễn chứng minh $KB \models \alpha$:

- a) $KB = \{P \vee Q; Q \Rightarrow (R \wedge S); (P \vee R) \Rightarrow U\}$, $\alpha = U$.
- b) $KB = \{A \Rightarrow B \wedge C; C \Rightarrow E \vee F; B \Rightarrow \neg E; \neg C \vee D; D \wedge F \Rightarrow G;$
 $G \Rightarrow H \vee K; \neg H \vee I; \neg K; A\}$, $\alpha = I$.
- c) $KB = \{A \Rightarrow C; D \wedge C \Rightarrow E; B \Rightarrow D; A \wedge B \Rightarrow F; E \wedge F \Rightarrow \neg D \vee H;$
 $B \wedge D \Rightarrow I \vee \neg E; I \wedge H \Rightarrow G; A; B\}$, $\alpha = G$.
- d) $KB = \{A \wedge B \Rightarrow \neg C \vee D; C \wedge E \Rightarrow A; E \vee F; \neg F \vee G;$
 $\neg(E \wedge \neg G); G \Rightarrow B \wedge E; E \Rightarrow H; H \Rightarrow C\}$, $\alpha = D$.
- e) $KB = \{\neg A \vee B; B \wedge D \Rightarrow F; B \Rightarrow \neg(C \wedge \neg D); \neg E \vee C;$
 $C \wedge D \Rightarrow \neg F \vee H; A; E\}$, $\alpha = H$.

Các dạng chuẩn

- Vấn đề: ở mỗi bước có quá nhiều luật suy diễn được sử dụng để chứng minh, nên chọn luật nào?
- Ví dụ:



Các dạng chuẩn

- Các câu trong logic mệnh đề có thể được chuyển về các dạng chuẩn (normal forms).
- **Dạng chuẩn hội (conjunctive normal form - CNF):** hội của các mệnh đề trong đó mỗi mệnh đề là tuyển của các mệnh đề đơn (literals).
 - Ví dụ: $(P \vee Q) \wedge (\neg P \vee \neg R \vee S)$
- **Dạng chuẩn tuyển (disjunctive normal form – DNF):** tuyển của các mệnh đề, trong đó mỗi mệnh đề là hội của các mệnh đề đơn.
 - Ví dụ: $(P \wedge \neg Q) \vee (\neg P \wedge R) \vee (R \wedge \neg S)$

Chuyển câu về dạng chuẩn hội

- Ví dụ chuyển về dạng chuẩn hội: $\neg(A \Rightarrow B) \vee (C \Rightarrow A)$

1. Loại bỏ \Rightarrow

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

2. Chuyển phủ định vào sát ký hiệu mệnh đề dùng luật De Morgan:

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

3. Áp dụng luật kết hợp và phân phối:

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

và

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

Thuật toán phân giải

- Ta có $KB \models \alpha$ nếu và chỉ nếu $(KB \wedge \neg\alpha)$ không thỏa mãn, tức là chứng minh phản chứng.

```
function PL-RESOLUTION( $KB, \alpha$ ) returns true or false
  inputs:  $KB$ , the knowledge base, a sentence in propositional logic
          $\alpha$ , the query, a sentence in propositional logic

  clauses  $\leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg\alpha$ 
  new  $\leftarrow \{\}$ 
  loop do
    for each pair of clauses  $C_i, C_j$  in clauses do
      resolvents  $\leftarrow$  PL-RESOLVE( $C_i, C_j$ )
      if resolvents contains the empty clause then return true
      new  $\leftarrow$  new  $\cup$  resolvents
    if new  $\subseteq$  clauses then return false
  clauses  $\leftarrow$  clauses  $\cup$  new
```

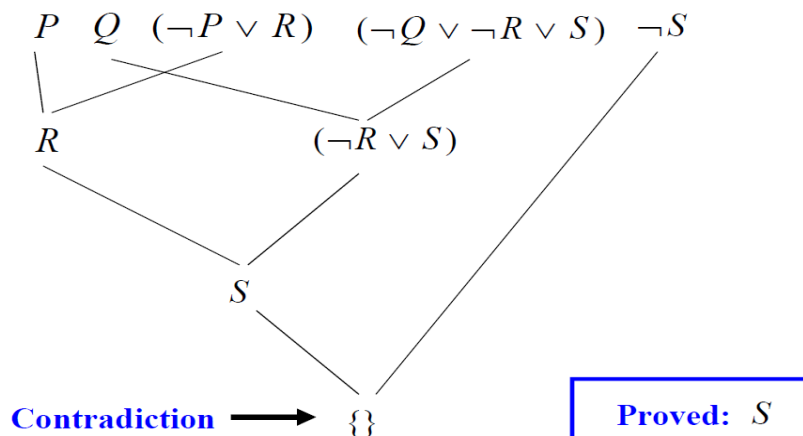
Figure 7.12 A simple resolution algorithm for propositional logic. The function PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

Thuật toán phân giải

- Thuật toán phân giải là **đúng đắn (sound)** và **hoàn chỉnh (complete)**.
- Ví dụ: $KB = \{(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]\}$ và $\alpha = S$. Chứng minh $KB \models \alpha$.
- Chuyển KB thành CNF
 - $P \wedge Q \equiv P \wedge Q$
 - $P \Rightarrow R \equiv (\neg P \vee R)$
 - $(Q \wedge R) \Rightarrow S \equiv \neg(Q \wedge R) \vee S \equiv \neg Q \vee \neg R \vee S$
 - $KB = \{P, Q, \neg P \vee R, \neg Q \vee \neg R \vee S\}$
- Bổ sung $\neg S$ vào KB
 - $KB = \{P, Q, \neg P \vee R, \neg Q \vee \neg R \vee S, \neg S\}$

Thuật toán phân giải

- Áp dụng thuật toán phân giải cho KB
 - $KB = \{P, Q, \neg P \vee R, \neg Q \vee \neg R \vee S, \neg S\}$



Mệnh đề xác định và mệnh đề Horn

- Mệnh đề xác định (definite clause)
 - Là câu TUYỂN của các mệnh đề đơn (literals), trong đó có chỉ một mệnh đề dương (positive).
 - Ví dụ: $(\neg P \vee \neg Q \vee R)$ là mệnh đề xác định, nhưng $(\neg P \vee Q \vee R)$ không phải là mệnh đề xác định.
 - Chú ý mệnh đề $(Q \wedge R) \Rightarrow S$ cũng là mệnh đề xác định vì $(Q \wedge R) \Rightarrow S \equiv \neg(Q \wedge R) \vee S \equiv \neg Q \vee \neg R \vee S$.
- Mệnh đề Horn
 - Là tuyển của các mệnh đề đơn, trong đó có nhiều nhất một mệnh đề dương.
 - Ví dụ $(\neg P \vee \neg Q \vee R)$ và $(\neg P \vee \neg Q \vee \neg R)$ là các mệnh đề Horn.
 - Mệnh đề xác định cũng là mệnh đề Horn.

Mệnh đề xác định và mệnh đề Horn

$$\begin{aligned}
 CNFSentence &\rightarrow Clause_1 \wedge \dots \wedge Clause_n \\
 Clause &\rightarrow Literal_1 \vee \dots \vee Literal_m \\
 Literal &\rightarrow Symbol \mid \neg Symbol \\
 Symbol &\rightarrow P \mid Q \mid R \mid \dots \\
 HornClauseForm &\rightarrow DefiniteClauseForm \mid GoalClauseForm \\
 DefiniteClauseForm &\rightarrow (Symbol_1 \wedge \dots \wedge Symbol_l) \Rightarrow Symbol \\
 GoalClauseForm &\rightarrow (Symbol_1 \wedge \dots \wedge Symbol_l) \Rightarrow False
 \end{aligned}$$

Figure 7.14 A grammar for conjunctive normal form, Horn clauses, and definite clauses. A clause such as $A \wedge B \Rightarrow C$ is still a definite clause when it is written as $\neg A \vee \neg B \vee C$, but only the former is considered the canonical form for definite clauses. One more class is the k -CNF sentence, which is a CNF sentence where each clause has at most k literals.

Mệnh đề xác định và mệnh đề Horn

- Mệnh đề xác định và mệnh đề Horn có thể được biểu diễn theo dạng:

$$\begin{aligned} (\text{HỘI của các mệnh đề}) &\Rightarrow (\text{mệnh đề}) \\ (\text{body}) &\Rightarrow (\text{head}) \end{aligned}$$

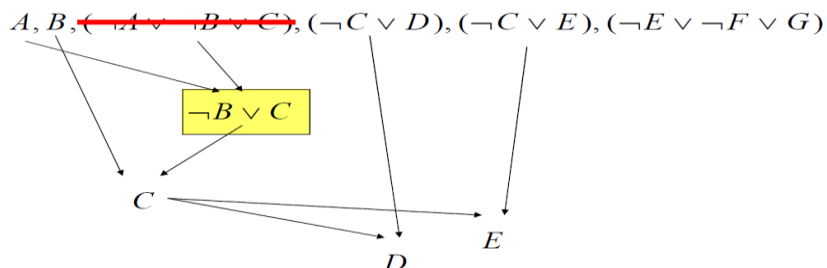
□ Ví dụ: $(Q \wedge R) \Rightarrow S \equiv \neg(Q \wedge R) \vee S \equiv \neg Q \vee \neg R \vee S$.

- Suy diễn với mệnh đề Horn có thể được thực hiện bởi các thuật toán suy diễn tiến (forward - chaining) và suy diễn lùi (backward - chaining) – là dạng suy diễn cơ bản của lập trình logic.
- Suy diễn với mệnh đề Horn với luật phân giải được thực hiện với thời gian tuyến tính của kích thước.

Mệnh đề xác định và mệnh đề Horn

- Kích thước của mệnh đề: số các ký hiệu có trong mệnh đề.
- Kích thước của KB theo dạng chuẩn Horn: tổng các kích thước của các mệnh đề.

□ Ví dụ $A, B, (A \wedge B \Rightarrow C), (C \Rightarrow D), (C \Rightarrow E), (E \wedge F \Rightarrow G)$, hoặc
 $A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$
 có kích thước là 12.



Suy diễn tiến

- Suy diễn tiến (forward-chaining algorithm): suy diễn từ KB cho đến khi có được mệnh đề kết luận.

```

function PL-FC-ENTAILS?( $KB, q$ ) returns true or false
  inputs:  $KB$ , the knowledge base, a set of propositional definite clauses
            $q$ , the query, a proposition symbol
   $count \leftarrow$  a table, where  $count[c]$  is the number of symbols in  $c$ 's premise
   $inferred \leftarrow$  a table, where  $inferred[s]$  is initially false for all symbols
   $agenda \leftarrow$  a queue of symbols, initially symbols known to be true in  $KB$ 

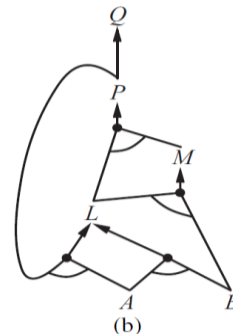
  while  $agenda$  is not empty do
     $p \leftarrow \text{POP}(agenda)$ 
    if  $p = q$  then return true
    if  $inferred[p] = \text{false}$  then
       $inferred[p] \leftarrow \text{true}$ 
      for each clause  $c$  in  $KB$  where  $p$  is in  $c$ .PREMISE do
        decrement  $count[c]$ 
        if  $count[c] = 0$  then add  $c$ .CONCLUSION to  $agenda$ 
  return false
  
```

Suy diễn tiến

- Suy diễn tiến (forward-chaining algorithm): suy diễn từ KB cho đến khi có được mệnh đề kết luận.
- Ví dụ cho KB ở hình (a), chứng minh Q . (xem class13, tài liệu [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

(a)



(b)

Figure 7.16 (a) A set of Horn clauses. (b) The corresponding AND-OR graph.

Suy diễn lùi

- Suy diễn lùi (backward-chaining)
 - Suy diễn từ mệnh kết luận để xác định các mệnh đề được sử dụng trong KB .
 - Nếu tất cả các mệnh đề được sử dụng là đúng thì mệnh đề kết luận được chứng minh.
- Các thuật toán suy diễn tiến và suy diễn lùi là các thuật toán đúng đắn (sound) và hoàn chỉnh (complete).

Bài tập

Sử dụng thuật toán phân giải và thuật toán suy diễn tiến chứng minh $KB \models \alpha$ với KB và α như sau:

- a) $KB = \{P \vee Q; Q \Rightarrow (R \wedge S); (P \vee R) \Rightarrow U\}$, $\alpha = U$.
- b) $KB = \{A \Rightarrow B \wedge C; C \Rightarrow E \vee F; B \Rightarrow \neg E; \neg C \vee D; D \wedge F \Rightarrow G;$
 $G \Rightarrow H \vee K; \neg H \vee I; \neg K; A\}$, $\alpha = I$.
- c) $KB = \{A \Rightarrow C; D \wedge C \Rightarrow E; B \Rightarrow D; A \wedge B \Rightarrow F; E \wedge F \Rightarrow \neg D \vee H;$
 $B \wedge D \Rightarrow I \vee \neg E; I \wedge H \Rightarrow G; A; B\}$, $\alpha = G$.
- d) $KB = \{\neg A \vee B; B \wedge D \Rightarrow F; B \Rightarrow \neg(C \wedge \neg D); \neg E \vee C;$
 $C \wedge D \Rightarrow \neg F \vee H; A; E\}$, $\alpha = H$.