TRÍ TUỆ NHÂN TẠO
(0101001005)
Chương 3:
BIỂU DIỄN TRI THỨC BẰNG LOGIC

Tài liệu [1]: Chương 6, Chương 7, Chương 8

Tài liệu [2]: Chương 3

Giảng viên: TS. DƯƠNG VĂN HIẾU

BỘ MÔN CNTT, KHOA KTCN-TRƯỜNG ĐH TIỀN GIANG

ĐT: 0988 987 907, email: duongvanhieu@tgu.edu.vn

1. LẬP LUẬN VÀ LOGIC

1.1. Giới thiệu

- Trong thực tế, khi xảy ra các cuộc tranh luận, người chiến thắng thường là người có khả năng lập luận tốt và có khả năng thuyết phục người khác
- Con người thông minh vì có khả năng lập luận. Vậy máy tính (gồm phần cứng+phần mềm) có thông minh hay không?
- Máy tính được gọi là thông minh thì nó phải có khả năng lập luận như con người. **Vậy, làm cách nào để máy tính có thể lập luận như con người?**

1. LẬP LUẬN VÀ LOGIC

1.2. Khái niệm về lập luận

- Lập luận là hành động sinh ra một phát biểu đúng mới từ các phát biểu đúng đã biết trước
- Một máy tính biết lập luận nếu nó có thể đưa ra một phát biểu đúng khi cho nó biết trước một tập các phát biểu đúng
- Một tập các phát biểu đúng cho trước được gọi là cơ sở tri thức (KB=Knowledge Base)

1. LẬP LUẬN VÀ LOGIC

1.2. Khái niệm về lập luận

- Các phát biểu trong lập luận phải tuân theo một tập các qui tắc nhất định và cách xác định một phát biểu là đúng hay sai

1. LẬP LUẬN VÀ LOGIC

1.3. Khái niệm về logic

- Logic là tập hợp các qui tắc qui định ngữ pháp và cách xác định ngữ nghĩa của các phát biểu
- Như vậy, logic là một ngôn ngữ mà mỗi câu trong đó có ngữ nghĩa (giá trị) là đúng hoặc sai
- Câu cần chứng minh là đúng khi biết cơ sở tri thức (KB) đúng gọi là câu truy vấn (query - q).
- Nếu q là đúng khi KB là đúng thì ta nói rằng KB suy diễn ra q (ký hiệu là KB $\models$ q)

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

a) Định nghĩa

- Các phát biểu trong logic mệnh đề được hình thành từ các ký hiệu mệnh đề và các toán tử logic
- Ví dụ mệnh đề "Nếu trời mưa thì đường ướt" được biễn diễn thành A=>B. Trong đó:
  - A là "trời mưa"
  - B là "đường ướt"
- Khi học logic mệnh đề thì phải hiểu cú pháp và ngữ nghĩa của logic mệnh đề

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

b) Cú pháp

- **Ký hiệu** được sử dụng:
  - Hằng giá trị: true, false
  - Mệnh đề: P, Q, R, … (chữ cái in hoa)
  - Các phép nối logic: phủ định ($\neg$), và ($\wedge$) , hoặc ($\vee$) , kéo theo ($\Rightarrow$), tương đương ($\Leftrightarrow$)
  - Các ký hiệu ưu tiên: mở ngoặc, đóng ngoặc

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

b) Cú pháp

- **Quy tắc** xây dựng câu:
  - **Câu đơn**:
    - Hằng giá trị true, false là câu đơn
    - Mỗi ký hiệu mệnh đề (ký tự in hoa) **P**, **Q**, **R**,…là câu đơn

Ví dụ:

P="An ham học"

Q="An học giỏi"

R="An có học bổng"

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

b) Cú pháp

- **Quy tắc** xây dựng câu:
  - **Câu phức**:
    - Câu phức là sự kế hợp giữa các câu đơn P và Q bằng cách sử dụng 1 phép nối logic $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
    - Sự kết hợp giữa các câu đơn, câu phức tạo thành câu phức
  - Ví dụ:

    $(P \wedge Q) \Rightarrow R$

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- true là đúng, false là sai
- Mỗi ký hiệu mệnh đề biểu diễn 1 phát biểu hay mệnh đề trong thế giới thực:
  - ✓ Ký hiệu mệnh đề có giá trị đúng nếu phát biểu/mệnh đề đó đúng
  - ✓ Ký hiệu mệnh đề có giá trị sai nếu phát biểu/mệnh đề đó sai

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- Ngữ nghĩa và giá trị của mệnh đề phức (câu phức)
  - $\neg P$ là phủ định của mệnh đề P:
    - $\neg P$ có giá trị đúng nếu P có giá trị sai
    - $\neg P$ có giá trị sai nếu P có giá trị đúng
  - $P \wedge Q$ có nghĩa là “P và Q”:
    - $P \wedge Q$ có giá trị đúng khi cả P và Q cùng đúng
    - $P \wedge Q$ có giá trị sai đối với các trường hợp còn lại

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- Ngữ nghĩa và giá trị của mệnh đề phức (câu phức)
  - $P \vee Q$ có nghĩa là "P hoặc Q":
    - $P \vee Q$ có giá trị đúng khi chỉ cần P hoặc Q đúng.
    - $P \vee Q$ có giá trị sai khi cả P và Q cùng sai
  - $P \Rightarrow Q$ có nghĩa là "P kéo theo Q":
    - $P \Rightarrow Q$ có giá sai khi P có giá đúng mà Q sai
    - $P \Rightarrow Q$ có giá đúng đối với các trường hợp còn lại
  - $P \Leftrightarrow Q$ có nghĩa là "P kéo theo Q" và "Q kéo theo P"

Câu hỏi: Giá trị của $P \Leftrightarrow Q$ như thế nào?

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- Tóm tắt ngữ nghĩa

P	**Q**	**¬P**	**P∧Q**	**P∨Q**	**P⇒Q**	**P⇔Q**
Đúng	Đúng	Sai	Đúng	Đúng	Đúng	Đúng
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

➢ Tóm tắt ngữ nghĩa

P	**Q**	**¬P**	**P∧Q**	**P∨Q**	**P⇒Q**	**P⇔Q**
Đúng	Đúng	Sai	Đúng	Đúng	Đúng	Đúng
Đúng	Sai	Sai	Sai	Đúng	Sai	Sai
Sai	Đúng	Đúng	Sai	Đúng	Đúng	Sai
Sai	Sai	Đúng	Sai	Sai	đúng	Đúng

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- Ví dụ 1:
- Gọi:
  - A là mệnh đề "An chăm học"
  - B là mệnh đề "An thông minh"
  - C là mệnh đề "An thi Trí tuệ nhân tạo đạt điểm cao"
- Câu phức:
  - $A \Rightarrow C$ là mệnh đề "nếu An chăm học thì An thi Trí tuệ nhân tạo đạt điểm cao"
  - $(A \vee B) \Rightarrow C$ là mệnh đề "nếu An chăm học hoặc An thông minh thì An thi Trí tuệ nhân tạo đạt điểm cao"

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- ✓ A là mệnh đề "An chăm học"
- ✓ B là mệnh đề "An thông minh"
- ✓ C là mệnh đề "An thi Trí tuệ nhân tạo đạt điểm cao"

➢ Ví dụ 1:

- ✓ SV tự cho A, B, C các giá trị phù hợp
- ✓ Tìm giá trị của 2 câu sau:

  1) $A \Rightarrow C$

  2) $(A \vee B) \Rightarrow C$

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- Ví dụ 2:
- Gọi:
  - ✓ A là mệnh đề "An lười học"
  - ✓ B là mệnh đề "An mê chơi game"
  - ✓ C là mệnh đề "An thi Trí tuệ nhân tạo đạt điểm cao"
- Câu phức:
  - ✓ $A \Rightarrow \neg C$ là mệnh đề "nếu An lười học thì An thi Trí tuệ nhân tạo đạt điểm KHÔNG cao"
  - ✓ $\neg(A \wedge B) \Rightarrow C$ là mệnh đề "nếu An KHÔNG lười học và mê chơi game thì An thi Trí tuệ nhân tạo đạt điểm cao"

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

c) Ngữ nghĩa

- ✓ A là mệnh đề "An lười học"
- ✓ B là mệnh đề "An mê chơi game"
- ✓ C là mệnh đề "An thi Trí tuệ nhân tạo đạt điểm cao"

- Ví dụ 2:
  - ✓ SV tự cho A, B, C các giá trị phù hợp
  - ✓ Tìm giá trị của 2 câu sau:

    1) $A \Rightarrow \neg C$

    2) $\neg(A \wedge B) \Rightarrow C$

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

d) Mệnh đề hằng đúng

- $(A \Leftrightarrow B) \Leftrightarrow ((A \Rightarrow B) \wedge (B \Rightarrow A))$ (kéo theo 2 chiều)
- $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$
- $(\neg \neg P) \Leftrightarrow P$ (Luật phủ định kép)
- $P \vee \neg P$ (Luật loại trừ)
- $\neg (P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$ (Luật DeMorgan)
- $\neg (P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$ (Luật DeMorgan)
- $P \vee Q \Leftrightarrow Q \vee P$ (Luật giao hoán)
- $P \wedge Q \Leftrightarrow Q \wedge P$ (Luật giao hoán)

2. LOGIC NGỮ NGHĨA

2.1. Logic mệnh đề

d) Mệnh đề hằng đúng

- $(P \wedge Q) \vee R \Leftrightarrow (P \vee R) \wedge (Q \vee R)$ (Luật phân phối)
- $(P \vee Q) \wedge R \Leftrightarrow (P \wedge R) \vee (Q \wedge R)$ (Luật phân phối)
- $((P \Rightarrow Q) \wedge P) \Rightarrow Q$ (Tam đoạn luận)
- $(A_1 \wedge \ldots \wedge A_n) \Leftrightarrow A_1 \wedge \ldots \wedge A_n$ (Bỏ dấu ngoặc)
- $(A_1 \vee \ldots \vee A_n) \Leftrightarrow A_1 \vee \ldots \vee A_n$ (Bỏ dấu ngoặc)
- $True \vee X$

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

a) Định nghĩa

- Để lập trình xử lý cơ sở tri thức, ta cần biến đổi mệnh đề (câu) về **dạng chuẩn hội**
- **Dạng chuẩn hội** là hội của các **câu tuyển**
- **Câu tuyển** là câu có dạng $A_1 \vee A_2 \vee \ldots \vee A_n$
  - ✓ Với A_i là ký hiệu mệnh đề, được gọi là literal
  - ✓ A_i là literal dương, $\neg A_i$ là literal âm
- **Câu dạng chuẩn hội:**

$$(A_{11} \vee A_{12} \vee \ldots \vee A_{1n}) \wedge \ldots \wedge (A_{k1} \vee A_{k2} \vee \ldots \vee A_{kn})$$

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

CÁC QUY TẮC

- **Á dụng theo thứ tự QT1, QT2, QT3, QT4**

- ✓ QT1: Loại bỏ $\Leftrightarrow$: thay thế $\alpha \Leftrightarrow \beta$ bằng $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
- ✓ QT2: Loại bỏ $\Rightarrow$: Thay thế $\alpha \Rightarrow \beta$ bằng $\neg\alpha \vee \beta$
- ✓ QT3: chuyển hoặc loại bỏ dấu $\neg$ đặt trước các ký hiệu bằng các luật deMorgan và luật phủ định kép $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$; $\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$; $\neg\neg\alpha = \alpha$.
- ✓ QT4: Áp dụng luật phân phối của phép $\wedge$ đối với phép $\vee$

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

b) Biến đổi câu bất kỳ về dạng chuẩn hội

- Áp dụng các quy tắc:
  - ✓ Thay **$A \Rightarrow B$** bởi **$\neg A \vee B$**
  - ✓ Thay **$\neg\neg A$** bởi **A**
  - ✓ Áp dụng luật DeMorgan để chuyển dấu $\neg$ vào sát ký hiệu mệnh đề
  - ✓ Áp dụng luật phân phối

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

b) Biến đổi câu bất kỳ về dạng chuẩn hội

- Ví dụ 1: Đưa $(P \Rightarrow Q) \vee \neg (R \vee \neg S)$ về hội chuẩn

  Ta có $(P \Rightarrow Q) \vee$ QT$\neg (R \vee \neg S)$

  $\Leftrightarrow (\neg P \vee Q) \vee \neg (R \vee \neg S)$ QT2: thay $p \Rightarrow Q$

  $\Leftrightarrow (\neg P \vee Q) \vee (\neg R \wedge S)$ QT3: luật DeMorgan

  $\Leftrightarrow \{(\neg P \vee Q) \vee \neg R\} \wedge \{(\neg P \vee Q) \vee S)\}$ *QT4*

  $\Leftrightarrow (\neg P \vee Q \vee \neg R) \wedge (\neg P \vee Q \vee S)$

  ✓ Như vậy ta nói biểu thức $(P \Rightarrow Q) \vee \neg (R \vee \neg S)$ được chuẩn hóa về dạng hội chuẩn là

  $$(\neg P \vee Q \vee \neg R) \wedge (\neg P \vee Q \vee S)$$

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

b) Biến đổi câu bất kỳ về dạng chuẩn hội

- Ví dụ 2: Tìm dạng chuẩn hội của $\neg C \Rightarrow A \wedge B$

  (Mời sv làm)

2. LOGIC NGỮ NGHĨA

2.2. Dạng chuẩn hội

b) Biến đổi câu bất kỳ về dạng chuẩn hội

- Ví dụ 2: Tìm dạng chuẩn hội của $\neg C \Rightarrow A \wedge B$

Ta có $\neg C \Rightarrow A \wedge B$

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

*(Ap dụng thay **P=>Q bằng** **¬P ∨Q**)*

$\Leftrightarrow \quad \neg\neg C \vee (A \wedge B)$ bỏ dấu ngoặc

$\Leftrightarrow \quad C \vee (A \wedge B)$ thay $\neg\neg C$

$\Leftrightarrow \quad (C \vee A) \wedge (C \vee B)$ phân nối

✓ Như vậy ta nói biểu thức $\neg C \Rightarrow A \wedge B$ được chuẩn hóa về dạng hội chuẩn là $(C \vee A) \wedge (C \vee B)$

2. LOGIC NGỮ NGHĨA

2.3. Dạng chuẩn hội

c) Luật phân giải (resolution)

- Nếu $(P_1 \vee \ldots \vee P_{i-1} \vee P_i \vee P_{i+1} \vee \ldots \vee P_n)$
  và $(Q_1 \vee \ldots \vee Q_{j-1} \vee Q_j \vee Q_{j+1} \vee \ldots \vee Q_m)$ là đúng

với $P_i = \neg Q_j$

- Thì

$$P_1 \vee \ldots \vee P_{i-1} \vee P_{i+1} \vee \ldots \vee P_n \vee Q_1 \vee \ldots \vee Q_{j-1} \vee Q_{j+1} \vee \ldots \vee Q_m$$

cũng đúng

=> Mệnh đề mới là tuyển của các biến mệnh đề ban đầu sau khi bỏ đi P_i và Q_j

2. LOGIC NGỮ NGHĨA

2.4. Câu dạng Horn

- Trong phần dạng chuẩn hội, ta đã biết, *mọi câu trong logic mệnh đề đều có thể biểu diễn bởi dạng chuẩn hội* (hội của các câu tuyển)
- Mỗi câu tuyển có dạng $P_1 \vee P_2 \vee \ldots \vee P_n$ (với P_i là các biến mệnh đề).
- Nếu trong câu tuyển có nhiều nhất 1 biến mệnh đề dương (literal dương) thì câu tuyển đó gọi là câu dạng Horn

2. LOGIC NGỮ NGHĨA

2.4. Câu dạng Horn

- Vậy, câu dạng Horn là câu có 1 trong 3 dạng:
  1) $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n$ (không có biến dương nào)
  2) P (có 1 biến dương mà không có biến âm nào)
  3) $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n \vee Q$ (1 biến dương và ít nhất 1 biến âm)
- Một cách khác, câu dạng Horn là câu có dạng sau:
  - ✓ $\neg(P_1 \wedge P_2 \wedge \ldots \wedge P_n)$
  - ✓ P
  - ✓ $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \Rightarrow Q$

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

a) Định nghĩa

- H được xem là hệ quả logic của $G=\{G_1, \ldots, G_n\}$ nếu trong bất kỳ minh họa nào mà $\{G_1, \ldots, G_n\}$ đúng thì H cũng đúng.
- Khi cho trước KB, muốn suy ra tri thức mới từ KB đã có thì phải dùng các luật duy diễn
- Luật suy diễn có thể được biểu diễn dạng phân số:
  - ✓ Tử số là danh sách các điều kiện
  - ✓ Mẫu số là kết luận của luật

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

b) Một số luật suy diễn quan trọng:

1. Luật Modus Ponens (tam đoạn luận):

$$\frac{A \Rightarrow B,\ A}{B}$$ (*A kéo theo B, mà đã có A thì chắc chắn có B*)

2. Luật Modus Tollens

$$\frac{A \Rightarrow B,\ \neg B}{\neg A}$$ (*A kéo theo B, đã KHÔNG có B thì KHÔNG có A*)

Câu hỏi: Cho biết "Nếu trời mưa thì đường ướt"

- Nếu đường đã ướt thì kết luận trời đã mưa có đúng không? Tại sao?

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

b) Một số luật suy diễn quan trọng:

3. Luật bắc cầu

$$\frac{A \Rightarrow B, B \Rightarrow C}{A \Rightarrow C}$$ (*A kéo theo B và B kéo theo C thì A kéo theo C*)

4. Luật loại bỏ

$$\frac{P_1 \wedge P_2 \wedge \cdots \wedge P_n}{P_i}$$ (*Đã xảy ra p_1 đến p_n thì xảy ra p_i*)

Câu hỏi:

1. Có A>B và B>C thì kết luận A>C đúng hay không? Tại sao?

2. Có A là anh của B, B là anh của C. Vậy A thế nào với C?

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

b) Một số luật suy diễn quan trọng:

5. Luật đưa vào hội

$$\frac{P_1, P_2, \ldots, P_n}{P_1 \wedge P_2 \wedge \cdots \wedge P_n}$$ (*Đưa danh sách các biến vào hội các biến*)

6. Luật đưa vào tuyển

$$\frac{P_i}{P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n}$$ (*Từ 1 biến, đưa vào tuyển của các biến*)

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

b) Một số luật suy diễn quan trọng:

7. Luật phân giải

$$\frac{A \vee B,\ \neg B \vee C}{A \vee C}$$

Ví dụ:

Ta có: A="An thuộc bài";

B="An học giỏi"; C="An lười học"

Vậy:

$A \vee B$ ="An thuộc bài hoặc An học giỏi"

$\neg B \vee C$ =" An học KHÔNG giỏi hoặc An lười học"

$A \vee C$ ="An thuộc bài hoặc An lười học"

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

c) Luật suy diễn tin cậy

- Một luật suy diễn được xem là tin cậy nếu bất kỳ mô hình nào của giả thuyết của luật cũng là mô hình của kết luận. *Nói cách khác, với bất kỳ bộ giá trị nào của các biến mệnh đề mà tử số đúng thì mẫu số phải đúng*
- Chúng ta chỉ quan tâm đến luật suy diễn tin cậy
- Để chứng minh 1 luật suy diễn nào đó có phải là luật suy diễn tin cậy hay không thì người ta dùng bảng chân lý để kiểm tra.

2. LOGIC NGỮ NGHĨA

$$\frac{A \vee B,\ \neg B \vee C}{A \vee C}$$

2.5. Luật suy diễn

c) Luật suy diễn tin cậy

Ví dụ: Chứng minh luật phân giải là luật suy diễn tin cậy

- Lập bảng chân lý
- Kiểm tra nếu phần tử số $(A \vee B) \wedge (\neg B \vee C)$ đúng thì phần mẫu số $(A \vee C)$ phải đúng.

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

$$\boxed{\frac{A \vee B,\ \neg B \vee C}{A \vee C}}$$

c) Luật suy diễn tin cậy

Ví dụ: Chứng minh luật phân giải là luật suy diễn tin cậy

A	B	C	$\neg B$	$A \vee B$	$\neg B \vee C$	$(A \vee B) \wedge (\neg B \vee C)$	$A \vee C$
False	False	False	True	False	True	False	False
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

$$\frac{A \vee B,\ \neg B \vee C}{A \vee C}$$

c) Luật suy diễn tin cậy

Ví dụ: Chứng minh luật phân giải là luật suy diễn tin cậy

A	B	C	¬B	A∨B	¬B∨C	(A∨B)∧(¬B∨C)	A∨C
False	False	False	True	False	True	False	False
False	False	True	True	False	True	False	True
False	True	False	False	True	False	False	False
False	True	True	False	**True**	**True**	**True**	**True**
True	False	False	True	**True**	**True**	**True**	**True**
True	False	True	True	**True**	**True**	**True**	**True**
True	True	False	False	True	False	False	True
True	True	True	False	**True**	**True**	**True**	**True**

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

c) Luật suy diễn tin cậy

Bài tập:

- Chứng minh các luật suy diễn sau là luật suy diễn tin cậy

1) Modus Ponens $\dfrac{A \Rightarrow B,\ A}{B}$

2) Modus Tollens $\dfrac{A \Rightarrow B,\ \neg B}{\neg A}$

3) Luật bắc cầu $\dfrac{A \Rightarrow B, B \Rightarrow C}{A \Rightarrow C}$

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

d) Tiền đề, định lý, chứng minh

- Giả sử chúng ta có một tập các công thức. Các luật suy diễn cho phép ta suy ra công thức mới từ các công thức đã có bằng cách áp dụng các luật suy diễn
- Các luật quy diễn đã cho trước được gọi là **tiền đề**
- Các công thức được quy ra được gọi là **định lý**
- Dãy các luật được áp dụng để dẫn đến định lý được gọi là một **chứng minh** của định lý
- Nếu các luật suy diễn là tin cậy thì các định lý là hệ quả logic của các tiền đề

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

d) Tiền đề, định lý, chứng minh

- Ví dụ có các công thức sau:

$$Q \wedge S \Rightarrow G \vee H \qquad (1)$$

$$P \Rightarrow Q \qquad (2)$$

$$R \Rightarrow S \qquad (3)$$

$$P \qquad (4)$$

$$R \qquad (5)$$

- Chứng minh công thức $G \vee H$

2. LOGIC NGỮ NGHĨA

2.5. Luật suy diễn

$Q \wedge S \Rightarrow G \vee H$	(1)
$P \Rightarrow Q$	(2)
$R \Rightarrow S$	(3)
P	(4)
R	(5)

d) Tiền đề, định lý, chứng minh

- Chứng minh công thức $G \vee H$
- ✓ Từ (2) và (4) áp dụng quy tắc tam đoạn luận ta có Q (6)
- ✓ Từ (3) và (5) áp dụng quy tắc tam đoạn luận ta có S (7)

=> Vậy ta có $Q \wedge S$ (8)

- ✓ Từ (1) và (8) áp dụng quy tắc tam đoạn luận ta có $G \vee H$

2. LOGIC NGỮ NGHĨA

2.6. Phương pháp chứng minh bác bỏ

- Giả sử ta có tập công thức $G=\{G_1,\ldots,G_n\}$, ta cần chứng minh công thức H là hệ quả logic của G. Có nghĩa là chứng minh $G_1 \wedge \ldots \wedge G_n \Rightarrow H$ là chắc chắn
- Thay vì chứng minh $G_1 \wedge \ldots \wedge G_n \Rightarrow H$ là chắc chắn thì ta chứng minh $G_1 \wedge \ldots \wedge G_n \wedge \neg H$ là không thỏa được (hay mâu thuẫn).

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

a) Luật phân giải tổng quát

- Luật phân giải trên các câu tuyển như sau:

$$\frac{\begin{array}{c}\mathbf{A_1 \vee \ldots \vee A_n \vee C} \\ \mathbf{\neg C \vee B_1 \ldots \vee B_m}\end{array}}{\mathbf{A_1 \vee \ldots \vee A_n \vee B_1 \ldots \vee B_m}}$$

- Luật phân giải trên các câu dạng Horn:

$$\frac{\begin{array}{c}\boldsymbol{P_1 \wedge \ \ldots \wedge \mathrm{P_n} \wedge S \Rightarrow Q} \\ \boldsymbol{R_1 \wedge \ldots \wedge R_m \Rightarrow S}\end{array}}{\boldsymbol{P_1 \wedge \ \ldots \wedge \mathrm{P_n} \wedge R_1 \wedge \ldots \wedge R_m \Rightarrow Q}}$$

Trường hợp riêng:
$$\frac{\begin{array}{c}\boldsymbol{P_1 \wedge \ \ldots \wedge \mathrm{P_n} \wedge S \Rightarrow Q} \\ \boldsymbol{S}\end{array}}{\boldsymbol{P_1 \wedge \ \ldots \wedge \mathrm{P_n} \Rightarrow Q}}$$

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

a) Luật phân giải tổng quát

- Khi ta có thể áp dụng luật phân giải cho 2 câu thì:
  - 2 câu này được gọi là **hai câu phân giải được**
  - kết quả quả phân giải được gọi là **phân giải thức** của chúng
- Phân giải thức của 2 câu A và B được ký hiệu là **res(A,B)**
- Giả sử G là tập hợp các câu tuyển, ký hiệu **R(G)** là tập hợp các câu thuộc G và câu được sinh ra từ G bằng cách áp dụng luật phân giải

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

b)Định lý phân giải [2]

procedure *Resolution;*

Input: tập G các câu tuyển ;

begin

*1.**Repeat***

1.1 Chọn hai câu A và B thuộc G;

*1.2 **if** A và B phân giải được **then** tính Res(A,B);*

*1.3 **if** Res(A,B)là câu mới **then** thêm Res(A,B)vào G;*

until *nhận được [] hoặc không có câu mới xuất hiện;*

*2. **if** nhận được câu rỗng **then** thông báo G không thoả được*

else *thông báo G thoả được;*

end

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

b) Định lý phân giải [2]

- Định lý phân giải chỉ ra rằng "Nếu từ các câu thuộc G, bằng cách áp dụng luật phân giải dẫn đến câu rỗng thì G là không thỏa" *(hoặc G chứa 2 literal đối lập: P, ¬P)*

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

b) Định lý phân giải [2]

- Giả sử có G gồm 6 câu tuyển sau:

  $\neg A \vee \neg B \vee P$ (1)

  $\neg C \vee \neg D \vee P$ (2)

  $\neg E \vee C$ (3)

  A (4)

  E (5)

  D (6)

- Chứng minh P

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

$\neg A \vee \neg B \vee P$	(1)
$\neg C \vee \neg D \vee P$	(2)
$\neg E \vee C$	(3)
A	(4)
E	(5)
D	(6)
$\neg P$	(7)

b) Định lý phân giải [2]

- Cách chứng minh:
- ✓ Đầu tiên, thêm $\neg P$ vào G
  => ta có G gồm 7 câu
- ✓ Áp dụng phân giải (2) và (7) ta có $\neg C \vee \neg D$ (8)
- ✓ Áp dụng phân giải (6) và (8) ta có $\neg C$ (9)
- ✓ Áp dụng phân giải (3) và (9) ta có $\neg E$ (10)
- ✓ Vì (5) và (10) đối lặp nhau nên mâu thuẫn

=> Vậy đã chứng minh được P từ G hay P là hệ quả logic của các câu từ (1) đến (6)

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Thuật toán Harvard(1970)

- **Bước 1**: Viết lại giả thuyết và kết luận của bài toán dưới dạng chuẩn: $GT_1,\ldots,GT_n \Rightarrow KL_1,\ldots, KL_m$ (*với GT_i, KL_i được xây dựng từ các biến mệnh đề và phép nối $\wedge$, $\vee$, $\neg$*)
- **Bước 2**: Bỏ phủ định bằng cách chuyển vế GT_i sang vế kết luận, chuyển vế KL_j sang vế giả thuyết (*chuyển vế thì mất dấu $\neg$*)
- **Bước 3**: Thay dấu $\wedge$ ở GT_i và dấu $\vee$ ở KL_j bằng dấu phẩy
- **Bước 4**: Nếu GT_i còn dấu $\vee$ và KL_j còn dấu $\wedge$ thì tách chúng thành 2 dòng con

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Thuật toán Harvard(1970)

- **Bước 5**:Một dòng được chứng minh nếu tồn tại chung 1 mệnh đề ở cả 2 vế
- **Bước 6**: Bài toán được chứng minh khi và chỉ khi tất cả các dòng được chứng minh. Ngược lại thì bài toán không được chứng minh

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Ví dụ áp dụng thuật toán Harvard(1970)

Chứng minh $P\wedge (\neg P \vee Q) \Rightarrow Q$

Bước 1:Viết lại giả thuyết và kết luận của bài toán dưới dạng chuẩn, ta được: $P \wedge (\neg P \vee Q) \Rightarrow Q$ vì đã ở dạng chuẩn

Bước 2: Kết quả chuyển vế các GT_i, KL_j có dấu phủ định, ta được

$P \wedge (\neg P \vee Q) \Rightarrow Q$ vì không có nội dung cần chuyển

Bước 3: Thay dấu $\wedge$ ở GT_i và dấu $\vee$ ở KL_j bằng dấu phẩy, ta được

$P, (\neg P \vee Q) \Rightarrow Q$

Bước 4: Tách $\neg P \vee Q$ thành 2 dòng con, ta được

$P, \neg P \Rightarrow Q \quad \Leftrightarrow \quad P \Rightarrow Q, P \quad (1)$

$P, Q \Rightarrow Q \quad (2)$

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Ví dụ áp dụng thuật toán Harvard(1970)

Bước 4, có kết quả cuối cùng được 2 câu con

$P \Rightarrow Q, P$ (1)

$P, Q \Rightarrow Q$ (2)

Bước 5:

Vì (1) có chứa P ở 2 vế nên (1) được chứng minh

Vì (2) có chứa Q ở 2 vế nên (2) được chứng minh

Bước 6: Vì tất cả các câu con được chứng minh nên câu ban đầu được chứng minh

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Thuật toán Robin son (1971)

- **Bước 1**: Viết lại giả thuyết và kết luận của bài toán dưới dạng chuẩn: $GT_1,\ldots,GT_n \Rightarrow KL_1,\ldots,KL_m$ (*với GT_i, KL_i được xây dựng từ các biến mệnh đề và phép nối* $\wedge$, $\vee$, $\neg$)
- **Bước 2**: Thay dấu $\wedge$ ở GT_i và dấu $\vee$ ở KL_j bằng dấu phẩy
- **Bước 3**: Chuyển vế KL_j sang vế giả thuyết (bổ sung dấu phủ định) thành $GT_1,\ldots,GT_n$, $\neg KL_1,\ldots,\neg KL_m$
- **Bước 4**: Xây dựng một mệnh đề mới bằng cách tuyển một cặp mệnh đề từ danh sách các mệnh đề. Nếu mệnh đề mới có các biến mệnh đề đối ngẫu thì mệnh đề đó bị loại bỏ.

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Thuật toán Robin son (1971)

- **Bước 5**: Bổ sung mệnh đề mới vào danh sách và lặp lại bước 4
- **Bước 6**: Bài toán được chứng minh khi và chỉ khi còn 2 mệnh đề đối nhẫu. Ngược lại thì bài toán không được chứng minh

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Ví dụ áp dụng thuật toán Robin son (1971)

Chứng minh $P \wedge (\neg P \vee Q) \Rightarrow Q$

- **Bước 1**: Viết lại giả thuyết và kết luận của bài toán dưới dạng chuẩn, ta được $P \wedge (\neg P \vee Q) \Rightarrow Q$ vì đã ở dạng chuẩn
- **Bước 2**: Thay dấu $\wedge$ ở GT_i và dấu $\vee$ ở KL_j bằng dấu phẩy, ta được
  $P, (\neg P \vee Q) \Rightarrow Q$
- **Bước 3**: Chuyển vế KL_j sang vế giả thuyết (bổ sung dấu phủ định), ta được $P, \neg P \vee Q, \neg Q$
- **Bước 4**: Tuyển của P với $\neg P \vee Q$ thành Q, bổ sung Q vào danh sách, ta được $\neg Q, Q$

2. LOGIC NGỮ NGHĨA

2.7. Chứng minh bác bỏ bằng luật phân giải

c) Chứng minh sự đúng đắn của biểu thức

Ví dụ áp dụng thuật toán Robin son (1971)

Chứng minh $P \wedge (\neg P \vee Q) \Rightarrow Q$

- **Bước 4**: Kết quả: $\neg Q$, Q
- **Bước 6**: Bài toán được chứng minh vì chỉ còn 2 mệnh đề đối ngẫu

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

a) Thuật toán suy diễn dựa trên bảng giá trị chân lý [1]

- Thuật toán suy diễn làm việc trên cơ sở tri thức(KB) đúng đã biết
- Mục đích của thuật toán là để kiểm tra câu truy vấn q là đúng hay sai dựa trên cơ sở tri thức
- Suy luận dựa trên bảng giá trị chân lý là liệt kê tất cả các trường hợp có thể có của tập các ký hiệu mệnh đề, rồi kiểm tra xem q có đúng hay không

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

a) Thuật toán suy diễn dựa trên bảng giá trị chân lý [1]

```
Function Suydien_Lietke(KB, q) return true or false
    symbols=get_list_of_symbols(KB,q);
    n= symbols.size();
    int bộ_giá_trị[n]; //dùng để lưu bộ các giá trị logic (true:1, false:0)
    for (i=1; i≤2^n; i++) {
        bộ_giá_trị [1,..,n]=generate(i); // sinh ra bộ thứ i
        if (evaluate(KB, bộ_giá_trị)==true && evaluate(q, bộ_giá_trị)=false)
            return false
    return true;
```

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

a) Thuật toán suy diễn dựa trên bảng giá trị chân lý [1]

- Thuật toán suy diễn dựa trên bảng giá trị chân lý sinh ra toàn bộ bảng giá trị chân lý để đánh giá cơ sở tri thức (KB) và q
- Nếu tồn tại 1 trường hợp KB đúng mà q sai thì kết luận không thể suy diễn được ra q từ KB.
- Giải thuật trên có độ phức tạp thời gian là $2^n * m$, với n là số ký hiệu có trong KB, q và m độ dài câu trong KB

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

b) Thuật toán suy diễn dựa trên luật phân giải [1]

- Thuật toán suy diễn dựa trên luật phân giải có độ phức tạp thời gian bé hơn thuật toán suy diễn dựa trên bảng giá trị chân lý
- Thực hiện phân giải liên tiếp trên câu dạng chuẩn hội
- Để chứng minh $KB \vDash q$, ta chứng minh $KB \wedge \neg q \vDash []$
- Do đó, trước hết ta cần biến đổi biểu thức $KB \wedge \neg q$ về dạng chuẩn hội

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

b) Thuật toán suy diễn dựa trên luật phân giải [1]

- Sau đó áp dụng liên tiếp luật phân giải trên các cặp câu tuyển mà có ít nhất 1 biến mệnh đề là đối của nhau (ví dụ P và ¬P) để sinh ra một câu tuyển mới, câu tuyển mới này được bổ sung vào danh sách các câu tuyển
- Giải thuật dừng khi :
  - ✓ có câu [] được sinh ra thì kết luận KB⊨q
  - ✓ Không có câu nào được sinh ra thì kết luận KB không suy diễn ra q

2. LOGIC NGỮ NGHĨA

2.8. Thuật toán suy diễn

b) Thuật toán suy diễn dựa trên luật phân giải [1]

```
Function Resolution(KB, q) return true or false
    clauses=get_list_of_clauses(KB ∧ ¬q);
    new={};
    do
      for each Ci, Cj in clauses
              new_clause= resol(Ci,Cj);
              if new_clause=[] return true;
              new=new Y new_clause;
        if new ⊆ clauses return false;
        clauses=clauses Y new;
```

2. LOGIC NGỮ NGHĨA

Bài tập 1: Chứng minh các biểu thức sau là hằng đúng.

1) $(P \wedge Q) \Rightarrow Q$
2) $(P \wedge Q) \Rightarrow P$
3) $P \Rightarrow (\neg P \Rightarrow P)$
4) $P \Rightarrow (Q \Rightarrow (P \wedge Q))$

Bài tập 2: Chứng minh mệnh đề sau là hằng sai

1) $(\neg((r \vee q) \wedge q) \vee \neg p) \wedge ((\neg p \vee \neg q) \Rightarrow (p \wedge q \wedge r))$
2) $\neg p \wedge \neg(p \wedge q) \wedge \neg(p \wedge \neg r) \wedge (((\neg q \Rightarrow r) \vee \neg(q \vee (r \vee s) \vee (r \wedge \neg s))) \wedge p)$

Bài tập 3: Sử dụng thuật toán Harvard, Robin son để chứng minh

1) $(P \vee Q) \wedge (\neg P \vee R) \Rightarrow (Q \vee R)$
2) $(\neg P \vee Q) \wedge (\neg Q \vee R) \wedge (\neg R \vee S) \wedge (\neg U \vee \neg S) \Rightarrow \neg P \vee \neg U$

3. LOGIC VỊ TỪ CẤP 1

3.1. Giới thiệu

- Sau khi xem xét các thuật toán suy diễn trên logic mệnh đề, chúng ta thấy rằng độ phức tạp rất lớn và rất khó cài đặt trên cơ sở tri thức lớn có nhiều biến mệnh đề
- Để khắc phục hạn chế trên, người ta dùng logic vị từ

3. LOGIC VỊ TỪ CẤP 1

3.1. Giới thiệu

- Logic vị từ là mở rộng của logic mệnh đề
- Logic vị từ cho phép mô tả thế giới các đối tượng (ví dụ An, Bình), các thuộc tính của đối tượng và các mối quan hệ giữa các đối tượng (ví dụ An thích Bình)
- Sử dụng các biến để chỉ các đối tượng
- Sử dụng các vị từ để mô tả thuộc tính của đối tượng, mối quan hệ giữa các đối tượng
- Bổ sung thêm lượng từ tồn tại, lượng từ với mọi

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

a) Ký hiệu

- Ký hiệu hằng:
  - true, false là hằng của ngôn ngữ
  - tên của các đối tượng (ví dụ An, Bình)
- Ký hiệu biến:
  - Biến đối tượng (ví dụ x,y,z,t,u)
- Ký hiệu vị từ:
  - P,Q, Sinhvien, Yeu, father, ...

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

a) Ký hiệu

- Ký hiệu hàm:
  - sin, cos, log, father, … Ví dụ: love(An,Binh)
- Ký hiệu kết nối logic:
  - ¬, ∧, ∨, ⇒, ⇔
- Ký hiệu lượng từ:
  - ∀ (với mọi), ∃ (tồn tại)
- Các ký hiệu khác:
  - Mở ngoặc, đóng ngoặc, dấu phẩy

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

b) Quy tắc xây dựng câu

- **Câu đơn**:
  - True là câu đơn hằng đúng, false là câu đơn hằng sai
  - Ký_hiệu_vị_từ(hạng_thức_1, hạng_thức_2,...) là một câu đơn. Trong đó, hạng_thức_i là biểu thức của các đối tượng hoặc hàm của các tham số $f(t_1, t_2, ..,t_n)$
  - Ví dụ:
    - mother(Lucky, Gấu) *Lucky là mother của Gấu*
    - mother(Gấu, Cafe) *Gấu là mother của Cafe*

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

b) Quy tắc xây dựng câu

- **Câu phức:**
  - Nếu A, B là các câu và x là một ký hiệu biến thì các công thức sau là câu phức:
    - $\neg A$
    - $A \land B$
    - $A \lor B$
    - $A \Rightarrow B$
    - $A \Leftrightarrow B$
    - $\forall x, A$
    - $\exists x, A$

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

b) Quy tắc xây dựng câu

- Quy ước khác:
  - ✓ Hạng thức không chứa biến thì gọi là hạng thức nền
  - ✓ Một câu đơn cũng có tên gọi là câu phân tử hay công thức phân tử
  - ✓ Một câu đơn hoặc phủ định của một câu đơn thì gọi là literal
  - ✓ Miền giá trị của một biến là tập hợp các giá trị/đối tượng mà biến đó có thể nhận

3. LOGIC VỊ TỪ CẤP 1

3.2. Cú pháp

b) Quy tắc xây dựng câu

- Quy ước khác:
  - Trong công thức có ký hiệu lượng tử ($\forall x$, A hoặc $\exists x$, A) các biến x trong A gọi là biến buộc (biến lượng tử), biến nào trong A không phải là biến lượng tử thì gọi là biến tự do.
  - Các câu mà không có biến tự do gọi là câu đóng.
  - Trong môn học này, chúng ta chỉ quan tâm đến các câu đóng (xác định được tính đúng/sai của nó)

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Đối với câu đơn:
  - ✓ Mỗi câu đơn đóng (không chứa biến) tương ứng với một mệnh đề (phát biểu, sự kiện, thông tin) trong thế giới thực.
  - ✓ Giá trị chân lý của câu đơn phụ thuộc vào giá trị của mệnh đề (phát biểu, sự kiện, thông tin) mà nó biểu diễn

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Đối với câu đơn:
  - ✓ Ví dụ có vị từ **sinh_vien(x)** *có nghĩa "x là sinh viên"*
  - ✓ Nếu An là 1 thanh niên thì
    - **sinh_vien(An)** có chân trị đúng nếu thực tế An là 1 sinh viên
    - **sinh_vien(An)** có chân trị sai nếu thực tế An KHÔNG phải là 1 sinh viên

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Đối với câu phức:
  - ✓ $\neg A$ có giá trị true nếu A có giá trị false và ngược lại
  - ✓ $A \wedge B$ có giá trị true nếu A, B cùng true
  - ✓ $A \vee B$ có giá trị false nếu A, B cùng false
  - ✓ $A \Rightarrow B$ có giá trị false khi A là true và B là false
  - ✓ $A \Leftrightarrow B$ có giá trị true khi A, B có giá trị giống nhau

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Đối với câu phức:
  - Ví dụ có vị từ **sinh_vien(x)**, **cong_nhan(y)**
  - Nếu An và Bình là 2 thanh niên thì
    - **sinh_vien(An)** $\wedge$ **cong_nhan(Bình)** có chân trị đúng nếu thực tế An là 1 sinh viên và Bình là 1 công nhân
    - **sinh_vien(An)** $\vee$ **cong_nhan(Bình)** có chân trị đúng nếu thực tế An là 1 sinh viên hoặc Bình là 1 công nhân

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Đối với câu phức:
  - ✓ ∀x A có giá trị true nếu tất cả các câu sinh ra từ A bằng cách thay x bởi một giá trị/đối tượng cụ thể thuộc miền giá trị biến x đều là true
  - ✓ ∃x A có giá trị true khi có một giá trị x_0 trong miền giá trị của biến x làm cho A true

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Như vậy:
  - Tính đúng/sai của một câu đơn (vị từ) phụ thuộc vào tính đúng sai của sự kiện/thông tin mà nó biểu diễn
  - Tính đúng sai của câu phức phụ thuộc vào các qui tắc trên qua hệ giữa các vị từ.
  - Trong nhiều trường hợp, tính đúng/sai của các câu đơn có thể lập luận ra từ các các câu phức đã biết đúng/sai và các qui tắc chuyển đổi tính đúng/sai giữa các câu đơn và câu phức

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Như vậy:
  - Công thức $\forall x$ P là đúng khi và chỉ khi P đúng cho tất cả các giá trị trong miền giá trị của x
  - Công thức $\exists x$ P là đúng khi và chỉ khi 1 trong các công thức nhận được từ P bằng cách thay x bằng một đối tượng trong miền giá trị là đúng..

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Ví dụ:
  - Vị từ Sinh_viên(An) *có nghĩa "An là sinh viên"*
  - Vị từ Chăm_học(An) *có nghĩa "An chăm học"*
  - Vị từ Cha(Nam, Hoàng) *có nghĩa "Nam là cha của Hoàng"*
  - Câu phức $\forall$x Sinh_viên(x) $\Rightarrow$ Chăm_học (x)

    $\forall$x {Sinh_viên(x) $\Rightarrow$ Chăm_học (x)}

    có nghĩa "Mọi sinh viên đều chăm học"
  - Câu phức $\exists$x Sinh_viên(x) $\wedge$ Cha(Nam, x)

    $\exists$x {Sinh_viên(x) $\wedge$ Cha(Nam, x)}

    có nghĩa "Có 1 sinh viên có cha tên Nam

3. LOGIC VỊ TỪ CẤP 1

3.3. Ngữ nghĩa

- Ví dụ:
  - Vị từ Sinh_viên(x) *có nghĩa "x là sinh viên"*
  - Vị từ Học(x, TTNT) *có nghĩa "x học TTNT"*
  - Câu phức $\exists$x (Sinh_viên(x) $\wedge$ Học(x,TTNT))

    có nghĩa "Có ít nhất 1 sinh viên học TTNT"

3. LOGIC VỊ TỪ CẤP 1

3.4. Câu hằng đúng

- $\forall x\ P(x) \Leftrightarrow \forall y\ P(y)$ (quy tắc đổi tên)
- $\exists x\ P(x) \Leftrightarrow \exists y\ P(y)$ (quy tắc đổi tên)
- $\forall x \forall y\ P(x,y) \Leftrightarrow \forall y \forall x\ P(x,y)$ (quy tắc giao hoán)
- $\exists x \exists y\ P(x,y) \Leftrightarrow \exists y \exists x\ P(x,y)$ (quy tắc giao hoán)
- $\neg(\forall x\ P(x)) \Leftrightarrow \exists x \neg P(x)$ (luật DeMorgan)
- $\neg(\exists x\ P(x)) \Leftrightarrow \forall x \neg P(x)$ (luật DeMorgan)
- $\forall x\ P(x) \Rightarrow P(a)$ quy tắc loại bỏ $\vee$
- $\exists x\ P(x) \Rightarrow P(e)$ quy tắc loại bỏ $\exists$

3. LOGIC VỊ TỪ CẤP 1

3.4. Câu hằng đúng

- **$\neg(\forall x\ P(x)) \Leftrightarrow \exists x \neg P(x)$** (luật DeMorgan)
- **$\neg(\exists x\ P(x)) \Leftrightarrow \forall x \neg P(x)$** (luật DeMorgan)
- **Ví dụ 1:**
- **P(x)="x là ngày chủ nhật"**
- **$\forall x$ P(x)="mọi ngày là ngày chủ nhật" là sai**
- **$\neg(\forall x\ P(x))$ là đúng**
- **$\exists x \neg P(x)$="tồn tại 1 ngày không phải ngày chủ nhật", đúng**
- => Vậy cả 2 vế cùng đúng

3. LOGIC VỊ TỪ CẤP 1

3.4. Câu hằng đúng

- $\neg(\forall x\ P(x)) \Leftrightarrow \exists x \neg P(x)$ (luật DeMorgan)
- $\neg(\exists x\ P(x)) \Leftrightarrow \forall x \neg P(x)$ (luật DeMorgan)
- **Ví dụ 2:**
- **P(x)="người x phải chết"**
- **$\forall x$ P(x)="mọi con người đều chết" là đúng**
- **$\neg(\forall x\ P(x))$ là sai**
- **$\exists x \neg P(x)$="tồn tại 1 con người không chết", sai**
- => Vậy cả 2 vế cùng sai

3. LOGIC VỊ TỪ CẤP 1

3.4. Câu hằng đúng

- **$P(a) \Rightarrow \exists x\, P(x)$** quy tắc đưa ký hiệu ∃ vào
- **$\forall x\, P(x) \Leftrightarrow \neg\exists x \neg P(x)$** (chuyển đổi giữa ∀ và ∃)

(P(x) đúng với mọi x có nghĩa là không tồn tại giá trị x làm p(x) sai và ngược lại)

- **$\exists x\, P(x) \Leftrightarrow \neg\forall x \neg P(x)$** (chuyển đổi giữa ∀ và ∃)

(tồn tại ít nhất 1 giá trị x làm P(x) đúng có nghĩa là không phải tất cả giá trị x làm p sai)

- Luật tam đoạn luận
- Luật phân giải

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

a) Xét ví dụ:

- Cho câu "An là con trai. Thủy là con gái. Tóc của con gái dài hơn tóc của con trai"
- Yêu cầu chứng minh "Tóc của Thủy dài hơn tóc của An"

❖ Xây dựng cơ sở tri thức: **Tóc_dài_hơn(y,x)**

- Con_trai(An) (1)
- Con_gái(Thủy) (2)
- $\forall x \forall y$ Con_trai(x) $\wedge$ Con_gái(y) $\Rightarrow$ Tóc_dài_hơn(y,x) (3)

❖ Cần chứng minh **Tóc_dài_hơn(Thủy, An)**

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

a) Xét ví dụ:

- Cho câu "Năm 2024 là năm nhuần, năm 2023 là năm thường, năm thuần có nhiều ngày hơn năm thường"
- Yêu cầu chứng minh "Năm 2024 có nhiều ngày hơn năm 2023"

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

a) Xét ví dụ

- Xây dựng cơ sở tri thức:
  - ✓ Con_trai(An) (1)
  - ✓ Con_gái(Thủy) (2)
  - ✓ $\forall x \forall y$ Con_trai(x) $\wedge$ Con_gái(y) $\Rightarrow$ Tóc_dài_hơn(y,x) (3)

- Cách giải
  - ✓ Bước 1: Từ (1) và (2), áp dụng quy tắc A, B $\Rightarrow$ A $\wedge$ B ta có

    Con_trai(An) $\wedge$ Con_gái(Thủy) (4)

  - ✓ Bước 2: Áp dụng quy tắc loại bỏ $\forall$ trong công thức (3); thay x bằng An, thay y bằng Thủy ta có

    Con_trai(An) $\wedge$ Con_gái(Thủy) $\Rightarrow$ Tóc_dài_hơn(Thủy, An) (5)

  - ✓ Bước 3: Áp dụng luật tam đoạn luận đối với (4) và (5) ta được

    Tóc_dài_hơn(Thủy, An) đpcm

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

b) Cách giải bài toán lập luận

- Giống như trong logic mệnh đề, bài toán lập luận có thể xem là bài toán tìm đường đi như sau:
  - *Trạng thái đầu*: KB
  - *Các phép chuyển trạng thái:* Mỗi phép chuyển trạng thái là một lần áp dụng luật trong logic vị từ trên tập câu trong KB. Mỗi luật $\neg$ áp dụng cho KB sinh ra câu mới $\neg$(KB), bổ sung câu mới này vào KB được trạng thái mới KB $\wedge$ $\neg$(KB)
  - *Trạng thái đích:* trạng thái KB chứa q
  - *Chi phí cho mỗi phép chuyển:* 1

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

b) Cách giải bài toán lập luận

- Bài toán trên có thể tìm được lời giải bằng cách áp dụng các thuật toán tìm kiếm. Tuy nhiên, không gian tìm kiếm lời giải của bài toán này là rất lớn.
- Cũng giống như trong logic mệnh đề, nếu cơ sở tri thức (KB) và câu truy vấn (q) được biểu diễn bằng các câu có dạng thích hợp, thì chúng ta có thể chỉ cần áp dụng một loại luật của logic mệnh đề để chứng minh rằng KB $\models$ q.

3. LOGIC VỊ TỪ CẤP 1

3.5. Lập luận trong logoc vị từ

b) Cách giải bài toán lập luận

- **Nếu KB và q biểu diễn được bằng các câu dạnh Horn** thì **chỉ cần áp dụng liên tiếp các luật tam đoạn luận** là chứng minh được KB $\models$ q
- **Nếu KB và q biểu diễn bằng các câu dạng chuẩn hội** thì ta **chỉ cần liên tiếp áp dụng các luật phân giải** là thực hiện được việc suy diễn KB $\models$ q

=> CHUYỂN CÁC CÂU TRONG KB và q về dạng Horn hoặc về dạng chuẩn hội

3. LOGIC VỊ TỪ CẤP 1

3.6. Phép đồng nhất 2 vị từ, thuật giải đồng nhất

a) Phép đồng nhất

- Khi áp dụng luật trong logic vị từ, ta thường xuyên gặp phải việc đối sách các vị từ trong hai câu xem chúng có thể đồng nhất được với nhau không.
- Ví dụ khi áp dụng Luật loại bỏ ký hiệu ∀ trong câu có tính phổ biến (3) để được câu cụ thể trên bộ giá trị (x=An, y=Thuy), ta phải đối sánh các cặp vị từ <Con_trai(An) và Con_trai(x)>, <Con_gái(Thủy) và Con_gái(y)> để tìm ra giá trị x=An, y=Thủy để cho các cặp vị từ đó là hoàn toàn như nhau.

3. LOGIC VỊ TỪ CẤP 1

3.6. Phép đồng nhất 2 vị từ, thuật giải đồng nhất

a) Phép đồng nhất

- Việc đối sánh 2 vị từ để tìm ra 1 bộ giá trị cho các biến sao cho hai vị từ là đồng nhất được gọi là phép đồng nhất
- Ví dụ:
  - Đồng nhất (Con_trai(An), Con_trai(y)) = {y/An}
  - Đồng nhất (Yêu(An,x), Yêu(y,Binh)) = {x/Binh; y/An}
  - Đồng nhất (Yêu(An,x), Yêu(y, Em_gái(Hoa)) = {x/Em_gái(Hoa); y/An}

  (chú ý: trong trường hợp này Em_gái(x) là một hàm – em gái của x, không phải là vị từ)

3. LOGIC VỊ TỪ CẤP 1

3.6. Phép đồng nhất 2 vị từ, thuật giải đồng nhất

a) Phép đồng nhất

- Ví dụ:
  - Đồng nhất (Yeu(An,x), Yeu(An,y)) = {y/x }
  - Đồng nhất (Ban(An,x), Ban(y, Em_gái(y))={x/Em_gái(An); y/An}
  - Đồng nhất (P(a,X), P(X,b)) = **failure** ???
  - Đồng nhất [parents(x, father(x), mother(Jane)), parents(Bill, father(y), mother(y))]= **failure**???

3. LOGIC VỊ TỪ CẤP 1

3.6. Phép đồng nhất 2 vị từ, thuật giải đồng nhất

b) Giải thuật đồng nhất

- Input: hai literal *p* và *q*
- Output: Sự thay thế gán giá thay thế các biến *theta*

3. LOGIC VỊ TỪ CẤP 1

3.6. Phép đồng nhất 2 vị từ, thuật giải đồng nhất

b) Giải thuật đồng nhất

Procedure Đồng_nhất(p, q, theta) *return* true or false

(r,s)=hạng thức đầu tiên không nhất quán giữa (p,q);

if ((r,s)=empty) return theta; thành công

if (là_biến(r))

theta = theta Y {r/s}

Đồng_nhất(thaythe(theta,p), thaythe(theta,q), theta)

elseif (là_biến(s))

theta = theta Y {s/r}

Đồng_nhất(thaythe(theta,p), thaythe(theta,q), theta)

else return **failure**

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

a) Câu dạng chuẩn hội

- Cũng giống như trong logic mệnh đề, câu dạng chuẩn hội trong logic vị từ cấp một có dạng sau (là hội của các tuyển)

$$(A_{11} \vee A_{12} \vee \ldots \vee A_{1n}) \wedge (A_{21} \vee A_{22} \vee \ldots \vee A_{2m}) \wedge \ldots \wedge (A_{k1} \vee A_{k2} \vee \ldots \vee A_{kr})$$

- Trong đó:
  - ✓ $(A_{11} \vee A_{12} \vee \ldots \vee A_{1n})$, ..., $(A_{k1} \vee A_{k2} \vee \ldots \vee A_{kr})$ là các câu tuyển
  - ✓ A_{ij} là các literal

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Qui tắc 1. Loại bỏ dâu $\Leftrightarrow$:

  Thay thế $A \Leftrightarrow B$ bằng $(A \Rightarrow B) \wedge (B \Rightarrow A)$

- Qui tắc 2. Loại bỏ dấu $\Rightarrow$:

  Thay thế $A \Rightarrow B$ bằng $\neg A \vee B$

- Qui tắc 3. Áp dụng luật DeMorgan và phủ định kép:

  Thay $\neg(A \vee B)$ bằng $\neg A \wedge \neg B$; $\neg(A \wedge B)$ bằng $\neg A \vee \neg B$; $\neg\neg A$ bằng A;

  Thay $\neg \forall x P(x)$ bằng $\exists x \neg P(x)$; $\neg \exists x P(x)$ bằng $\forall x \neg P(x)$

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Qui tắc 4. Chuẩn hóa các biến (kèm theo lượng từ):

Thay $\forall x P(x) \vee \exists x Q(x)$ bằng $\forall x P(x) \vee \exists y Q(y)$

- Qui tắc 5. Chuyển các lượng từ về đầu câu:

Thay $\forall x P(x) \vee \exists y Q(y)$ bằng $\forall x \exists y P(x) \vee Q(y)$

- Qui tắc 6. Loại bỏ $\exists$ bằng giá trị vô danh e

Thay $\exists x$ P(x) thành P(e) với e là hằng vô danh không trùng với các ký hiệu đã có trong KB

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Qui tắc 6. Loại bỏ ∃ bằng giá trị vô danh e (tiếp tục)

Thay $\forall x \exists y \forall z P(x,y,z)$ thành $\forall x \forall z P(x,f(y),z)$ với f là hàm vô danh không trùng với ký hiệu trong KB

- Qui tắc 7. Bỏ qua các lượng từ ∀
- Qui tắc 8. Áp dụng luật phân phối của phép ∧ đối với phép ∨

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

Phát biểu:

"Tất cả con chó đều sủa về ban đêm. Hễ nhà ai có mèo thì nhà người đó đều không có chuột. Những ai khó ngủ thì đều không nuôi bất cứ con gì mà sủa về ban đêm. Bà Bình có mèo hoặc có chó"

Nhận xét: Đoạn phát biểu trên có 4 câu. Do đó, khi biểu diễn dạng logic vị từ thì mỗi câu trong đoạn được biểu diễn thành 1 câu trong logic vị từ.

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

Phát biểu "Tất cả con chó đều sủa về ban đêm. Hễ nhà ai có mèo thì nhà người đó đều không có chuột. Những ai khó ngủ thì đều không nuôi bất cứ con gì mà sủa về ban đêm. Bà Bình có mèo hoặc có chó"

Đặt:

$\forall x\ (\text{Là\_chó}(x) \Rightarrow \text{Sủa\_về\_đêm}(x))$ (1)

$\forall x\ \forall y(\text{Có}(x,y) \wedge \text{Là\_mèo}(y) \Rightarrow \neg\exists z(\text{Có}(x,z) \wedge \text{Là\_chuột}(z)))$ (2)

$\forall x\ (\text{Khó\_ngủ}(x) \Rightarrow \neg\exists z(\text{Có}(x,z) \wedge \text{Sủa\_về\_đêm}(z)))$ (3)

$\exists x(\text{Có}(\text{Bà Bình}, x) \wedge (\text{Là\_mèo}(x) \vee \text{Là\_chó}(x))$ (4)

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

$\forall x$ (Là_chó(x) $\Rightarrow$ Sủa_về_đêm(x))	(1)
$\forall x\ \forall y$(Có(x,y)$\wedge$Là_mèo(y) $\Rightarrow \neg\exists z$(Có(x,z)$\wedge$Là_chuột(z)))	(2)
$\forall x$ (Khó_ngủ(x) $\Rightarrow \neg\exists z$(Có(x,z)$\wedge$Sủa_về_đêm(z))	(3)
$\exists x$(Có(Bà Binh, x) $\wedge$(Là_mèo(x) $\vee$Là_chó(x))	(4)

Biến đổi câu (1):

- Áp dụng các quy tắc: Bỏ lượng từ mọi, loại bỏ dấu $\Rightarrow$`
- Do đó, câu (1) tương đương với câu sau:

$\neg$ Là_chó(x) $\vee$ Sủa_về_đêm(x)

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

$$\forall x\ \forall y(Có(x,y) \land Là\_mèo(y) \Rightarrow \neg \exists z(Có(x,z) \land Là\_chuột(z))) \qquad (2)$$

Biến đổi câu (2):

- Áp dụng quy tắc loại bỏ dấu $\Rightarrow$ ta được

$\forall x\ \forall y(\neg(Có(x,y) \land Là\_mèo(y)) \lor (\neg \exists z(Có(x,z) \land Là\_chuột(z)))$

- Áp dụng luật DeMorgan, ta được

$\forall x\ \forall y(\ (\neg\ Có(x,y) \lor \neg\ Là\_mèo(y)) \lor (\neg \exists z(Có(x,z) \lor \neg\ Là\_chuột(z))))$

$\forall x\ \forall y(\ (\neg\ Có(x,y) \lor \neg\ Là\_mèo(y)) \lor (\forall z\ \neg(Có(x,z) \lor \neg\ Là\_chuột(z))))$

$\forall x\ \forall y\ \forall z(\neg\ Có(x,y) \lor \neg\ Là\_mèo(y) \lor \neg(Có(x,z) \lor \neg\ Là\_chuột(z))$

- Bỏ lượng từ $\forall$: $\neg Có(x,y) \lor \neg Là\_mèo(y) \lor \neg Có(x,z) \lor \neg Là\_chuột(z))$

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

$\boxed{\forall x\ (Khó\_ngủ(x) \Rightarrow \neg\exists z(Có(x,z) \wedge Sủa\_về\_đêm(z)) \qquad (3)}$

Biến đổi câu (3):

- Áp dụng quy tắc (A⇒B)⇔(¬A∨B), ta có

$\forall x\ \{\neg\ Khó\_ngủ(x) \vee (\neg\exists z(Có(x,z) \wedge (Sủa\_về\_đêm(z))\}$

- Áp dụng luật DeMorgan, ta có

$\forall x\{\neg\ Khó\_ngủ(x) \vee (\forall z(\neg Có(x,z) \vee \neg\ (Sủa\_về\_đêm(z))\}$

$\forall x\ \forall z(\neg\ Khó\_ngủ(x) \vee \neg Có(x,z) \vee \neg\ (Sủa\_về\_đêm(z))$

- Bỏ lượng tự với mọi, ta có

$\neg\ Khó\_ngủ(x) \vee \neg Có(x,z) \vee \neg\ Sủa\_về\_đêm(z)$

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

b) Chuyển câu bất kỳ sang dạng chuẩn hội

- Ví dụ: Biểu diễn câu trong logic vị từ và biến đổi câu

$\boxed{\exists x(\text{Có}(\text{Bà Binh}, x) \land (\text{Là\_mèo}(x) \lor \text{Là\_chó}(x)) \qquad (4)}$

Biến đổi câu (4):

- Áp dụng quy tắt loại bỏ lượng từ tồn tại bằng giá trị vô danh a

Có(Bà Bình, a) ∧ (Là_mèo (a) ∨ Là_chó(a))

=> Kết luận: Bà Bình có mèo hoặc có chó

3. LOGIC VỊ TỪ CẤP 1

3.7. Câu dạng chuẩn hội, luật phân giải tổng quát

c) Luật phân giải

- Nếu chúng ta có 2 câu tuyển sau là đúng:

$(P_1 \vee \ldots \vee P_{i-1} \vee P_i \vee P_{i+1} \vee \ldots \vee P_n) \wedge$

$(Q_1 \vee \ldots \vee Q_{j-1} \vee Q_j \vee Q_{j+1} \vee \ldots \vee Q_m)$ và có phép thay thế theta sao cho thaythe(theta, P_i)=$\neg$thaythe(theta, Q_j) thì chúng ta cũng có câu tuyển sau là đúng

thaythe(theta, $P_1 \vee \ldots \vee P_{i-1} \vee P_{i+1} \vee \ldots \vee P_n \vee Q_1 \vee \ldots$ $Q_{j-1} \vee Q_{j+1} \vee \ldots \vee Q_m$)

Câu tuyển mới là tuyển của các literal trong 2 câu tuyển ban đầu nhưng bỏ đi P_i và Q_j

3. LOGIC VỊ TỪ CẤP 1

3.8. Câu dạng Horn và tam đoạn luận tổng quát

a) Câu dạng Horn

- Tất cả các câu trong logic vị từ cấp 1 đều có thể biểu diễn được dưới dạng chuẩn hội.
- Nếu trong câu tuyển có nhiều nhất 1 literal dương thì câu tuyển đó được gọi là câu dạng Horn.
- Câu dạng Horn là 1 trong 3 dạng:
  - ✓ $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n$ (không có literal dương nào)
  - ✓ P (chỉ có 1 literal dương)
  - ✓ $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n \vee Q$ (có 1 literal dương và ít nhất 1 literal âm)

3. LOGIC VỊ TỪ CẤP 1

3.8. Câu dạng Horn và tam đoạn luận tổng quát

b) Tam đoạn luận tổng quát [1]

- Nếu chúng ta có các câu Horn dương sau là đúng:
  - P'_1,
  - P'_2,
  - …
  - P'_n
  - $P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow Q$

Và có phép thay thế theta sao cho thaythe(theta, P'_i)=thaythe(theta, P_i)

Thì câu thaythe(theta, Q) là đúng

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Hai câu tuyển có một literal dương và một literal âm mà đồng nhất với nhau được thì sẽ sinh ra câu tuyển mới là tuyển các literal còn lại của cả hai câu sau khi bỏ đi hai literal đồng nhất này.
- Kết quả của phép phân giải cũng là câu dạng tuyển và khi bổ sung vào KB thì kết quả KB cũng là dạng chuẩn hội.
- Vì vậy mà trước khi áp dụng giải thuật phân giải ta phải chuyển KB $\wedge \neg q$ sang dạng chuẩn hội

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Giống như giải thuật phân giải trong logic mệnh đề, giải thuật phân giải trong loc vị từ cấp một cũng thực hiện liên tiếp các phép phân giải hai clause trong biểu diễn dạng chuẩn hội của KB ∧ ¬q, bổ sung clause mới vào KB và lặp lại đến khi hoặc sinh ra câu rỗng ([]) hoặc không kết quả phân giải không bổ sung thêm clause nào vào KB được nữa.

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

```
Function Resolution(KB, q) return true or false
    KB = KB ∧ ¬q
    clauses=get_list_of_clauses(KB);
    while ([] not in KB)
      (Ci,Cj)=get_resolvable_pair(KB); // lấy hai câu mà chứa cặp literals
                                        //có thể đồng nhất với nhau được,
                                        //nhưng dấu ngược nhau
       if  (Ci,Cj)=empty return "failure“
       else
            resolvent = resolution-rule(S1, S2);
            KB = KB ∧ resolvent;
    return “success”;
```

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Mỗi lần thực hiện phép phân giải là một phép chuyển trạng thái từ KB sang trạng thái mới KB ∧ resolvent (với resolvent là kết quả của phép phân giải).

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Ở một trạng thái bất kỳ, có nhiều cặp câu tuyển có thể phân giải được với nhau, hay nói cách khác có nhiều phép chuyển trạng thái; việc lựa chọn phép chuyển trạng thái nào là dựa trên chiến lược lựa chọn, chúng ta có thể chọn theo chiều rộng, hoặc chọn theo chiều sâu như các chiến lược tìm kiếm theo chiều rộng hoặc theo chiều sâu

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Việc chứng minh KB∧¬q ╞[] cũng có thể thực hiện bằng chiến lược chứng minh lùi (tìm kiếm lùi), xuất phát từ ¬q (là đích của bài toán gốc KB ╞q chứ không phải đích []) ta tìm các câu trong KB có thể phân giải được với ¬q, áp dụng luật phân giải theo chiều rộng, đến khi nào [] được sinh ra thì dừng. Giải thuật phân giải theo cách này gọi là giải thuật phân giải lùi.

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Việc chứng minh KB∧¬q ╞[] cũng có thể thực hiện bằng chiến lược chứng minh lùi (tìm kiếm lùi), xuất phát từ ¬q (là đích của bài toán gốc KB ╞q chứ không phải đích []) ta tìm các câu trong KB có thể phân giải được với ¬q, áp dụng luật phân giải theo chiều rộng, đến khi nào [] được sinh ra thì dừng. Giải thuật phân giải theo cách này gọi là giải thuật phân giải lùi.

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Ví dụ:
- Giả sử chúng ta có cơ sở tri thức như cho trong ví dụ trước.
- Hãy chứng minh "Nếu bà Bình là người khó ngủ thì nhà bà ấy không có chuột".
- Câu cần chứng minh này tương đương với câu sau trong logic vị từ cấp một (q):

$$\text{Khó\_ngủ}(\text{Bà Binh}) \Rightarrow \neg \exists z(\text{Có}(\text{Bà Binh}, z) \land \text{Là\_Chuột}(z))$$

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

- Ví dụ:

Ta có q: Khó_ngủ(Bà Binh) ⇒ ¬∃z(Có(Bà Binh, z) ∧ Là_Chuột(z))

Bước 1. Lấy phủ định của q, ta có

¬(Khó_ngủ(Bà Binh) ⇒ ¬∃z(Có(Bà Binh, z) ∧ Là_Chuột(z)))

Áp dụng luật (A⇒B)⇔(¬A∨B) cho q, ta có

¬(¬ Khó_ngủ(Bà Binh) ∨ ¬∃z(Có(Bà Binh, z) ∧ Là_Chuột(z)))

Áp dụng luật DeMorgan, ta có

Khó_ngủ(Bà Binh) ∧ ∃z(Có(Bà Binh, z) ∧ Là_Chuột(z))

Bỏ lượng tử ∃, thay z bằng giá trị vô danh b, ta có

Khó_ngủ(Bà Binh) ∧ (Có(Bà Binh, b) ∧ Là_Chuột(b))

3. LOGIC VỊ TỪ CẤP 1

3.9. Giải thuật phân giải [1]

$\forall x$ (Là_chó(x) $\Rightarrow$ Sủa_về_đêm(x)) (1)

$\forall x$ $\forall y$(Có(x,y)$\wedge$Là_mèo(y) $\Rightarrow\neg\exists z$(Có(x,z)$\wedge$Là_chuột(z))) (2)

$\forall x$ (Khó_ngủ(x) $\Rightarrow$ $\neg\exists z$(Có(x,z)$\wedge$Sủa_về_đêm(z)) (3)

$\exists x$(Có(Bà Binh, x) $\wedge$(Là_mèo(x) $\vee$Là_chó(x)) (4)

Bước 2. **Bước 2**. KB$\wedge\neg$q, ta có

- ¬(Là_chó(x) ∨ Sủa_về_đêm(x)) (1)
- ¬Có(x,y)∨¬Là_mèo(y) ∨¬Có(x,z)∨Là_chuột(z) (2)
- ¬ (Khó_ngủ(x) ∨ ¬Có(x,z)∨¬Sủa_về_đêm(z) (3)
- Có(Bà Binh, a) (4)
- (Là_mèo(x) ∨Là_chó(x)) (5)
- Khó_ngủ(Bà Binh) (6)
- Có(Bà Binh, b) (7)
- Là_Chuột(b) (8)

4. BIỂU DIỄN TRI THỨC BẰNG LOGIC VỊ TỪ CẤP 1

4.1. Giới thiệu

- Để biểu diễn tri thức bằng logic vị từ cấp 1, ta cần đưa ra:
  - Các ký hiệu hằng để chỉ hằng các đối tượng cụ thể
  - Các ký hiệu biến để chỉ các đối tượng bất kỳ
  - Các ký hiệu hàm để biểu diễn các quan hệ hàm
  - Các ký hiệu vị từ để biểu diễn quan hệ giữa các đối tượng

  => Các ký hiệu được đưa ra tạo thành hệ thống từ vựng
- Sử dụng các từ vựng được đưa ra để tạo ra các câu trong logic vị từ cấp 1 để biểu diễn tri thức mà chúng ta muốn xây dựng
- Sử dụng các ký hiệu hằng, biến, hàm để tạo ra các hạng thức biểu diễn đối tượng
- Để biểu diễn thuận lợi, chúng ta cần 1 cấu trúc đặc biệt là danh sách

4. BIỂU DIỄN TRI THỨC BẰNG LOGIC VỊ TỪ CẤP 1

4.2. Vị từ hằng [2]

- Vị từ hằng (dùng ký hiệu dấu =) biểu diễn mối quan hệ đồng nhất. Giả sử T_1 và T_2 là 2 hạng thức bất kỳ, công thức $T_1=T_2$ là đúng trong mộ minh họa nếu trong minh họa đó T_1 và T_2 ứng với cùng 1 đối tượng
- Ví dụ father(An)=Bình là đúng trong 1 minh họa nếu người ứng với father(An) và người ứng với Bình là một người
- Ví dụ chị_gái(x,y) có nghĩa là "x là chị gái của y" thì câu "An có ít nhất 2 chị gái" được biểu diễn như sau:

$$\exists x,y\ (\text{chị\_gái}\ (x, An) \wedge \text{chị\_gái}(y, An) \wedge \neg(x=y))$$

4.3. Danh sách và các phép toán trên danh sách [2]

- Danh sách là một dãy gồm n đối tượng bất kỳ
- Danh sách được biểu diễn bằng cặp dấu [] chứa các đối tượng bên trong, các đối tượng bên trong danh sách cách nhau bởi dấu phẩy
- Ký hiệu:
  - [] là danh sách rỗng
  - [An, Bình, 2024, date(24, Feb,2024)] là danh sách có 4 đối tượng khác nhau
  - [[An, [Thịnh, Vượng]], 2024] là danh sách có chứa danh sách con
  - [phần_đầu | phần_đuôi] là danh sách gồm phần đầu và phần đuôi

4. BIỂU DIỄN TRI THỨC BẰNG LOGIC VỊ TỪ CẤP 1

4.3. Danh sách và các phép toán trên danh sách [2]

- XEM NỘI DUNG TRONG TÀI LIỆU [2]

Chuẩn bị kiểm tra

- **Bài toán tìm kiếm**
  - ✓ Làm lại bài toán tìm kiếm theo chiều rộng, chiều sâu
  - ✓ Xem minh họa ở trang 20, 22 của tài liệu [1]
- **Logic mệnh đề và logic vị từ**
  - ✓ Chứng minh mệnh đề (Làm các bài tập ở Slide 63)
  - ✓ Chứng minh q là hệ quả logic của KB (xem cách chứng minh trong logic vị từ)