INTRODUCTION TO ARTIFICIAL INTELLIGENCE

PGS. TS. Nguyễn Thanh Bình

PREDICATE LOGIC

- Logic mệnh đề cho phép chúng ta biểu diễn các sự kiện.
- Mỗi ký hiệu trong logic mệnh đề được minh hoạ như những sự kiện trong thế giới thực, sử dụng các kết nối logic để tạo ra những câu phức hợp biểu diễn các sự kiện có ý nghĩa phức tạp hơn.
- Khả năng biểu diễn của logic mệnh đề chỉ giới hạn trong phạm vi các sự kiện.

PREDICATE LOGIC

- Thế giới thực bao gồm các đối tượng. Các đối tượng có tính chất riêng và có quan hệ với nhau.
- Mối quan hệ giữa các đối tượng rất đa dạng, phong phú.
- Đối tượng: một sinh viên, một cái bàn, một giáo viên...

PREDICATE LOGIC

- Tính chất: cái bàn có bốn chân, làm bằng gỗ, con số có tính chất là số thực, số hữu tỉ,...
- Quan hệ: cha con, anh em. bạn bè, thầy trò,..
- Hàm: quan hệ hàm. Ví dụ: quan hệ hàm ứng với mỗi người với ba me họ.
- Logic vị từ đóng vai trò quan trong vì khả năng biểu diễn của nó (cho phép ta biểu diễn tri thức về thế giới với các đối tượng, thuộc tính và các quan hệ của đối tượng), là cơ sở cho nhiều ngôn ngữ logic khác

PREDICATE LOGIC - FIRST ORDER

- Để mô tả các thuộc tính của đối tượng, trong logic vị từ, người ta đưa vào các vị từ *(predicate)*.
- Ngoài các kết nối logic như trong logic mệnh đề, logic vị từ cấp một còn sử dụng các lượng tử. Chẳng hạn, lượng tử $\forall$ cho phép ta tạo ra các câu nói tới mọi đối tượng trong một miền đối tượng nào đó.

PREDICATE LOGIC - FIRST ORDER

- Các ký hiệu:
  - Các ký hiệu hằng: a,b,c, John, Jerry,...
  - Các ký hiệu biến: x,y,z,u,v,w,...
  - Các ký hiệu vị từ: P, Q, R, S, Prime, Odd, Love...
- Mỗi vị từ là vị từ của n biến.
  - Ví dụ: Love là vị từ của hai biến, Prime là vị từ một biến.
- Các ký hiệu vị từ không biến là các ký hiệu mệnh đề.

PREDICATE LOGIC - FIRST ORDER

- Các ký hiệu hàm: f, g, cos, sin,...
- Mỗi hàm là hàm của n biến.
- Các ký hiệu kết nối logic giống như trong logic mệnh đề.
- Các ký hiệu lượng tử: $\forall$, $\exists$
- Các ký hiệu ngăn cách: dấu phẩy, dấu mở ngoặc, đóng ngoặc.

PREDICATE LOGIC - FIRST ORDER

- Các **hạng thức** *(term)* là các biểu thức mô tả đối tượng.
- Các hạng thức được xác định đệ quy như sau:
  - Các ký hiệu hằng hay biến là hạng thức.
  - Nếu a,b,c,...,z là n hạng thức và h là hàm n biến thì h(a,b,c,...,z) cũng là hạng thức.
  - Một hạng thức không chứa biến được gọi là hạng thức cụ thể

PREDICATE LOGIC - FIRST ORDER

- Chúng ta sẽ biễu diễn các tính chất của đối tượng và các quan hệ giữa các đối tượng bằng công thức phân tử (câu đơn).
- Các câu đơn được xác định đệ quy như sau:
  - Các ký hiệu vị từ không biến (các ký hiệu mệnh đề) là câu đơn.
  - Nếu a,b,c,...,z là n hạng thức và P là vị từ của n biến thì P(a,b,..,z) là công thức phân tử (câu đơn).
  - Ví dụ: *Mary* là một ký hiệu hằng, *Love* là một vị từ hai biến, *husband* là hàm 1 biến thì *Love(Mary,husband(Mary))* là một công thức phân tử.

PREDICATE LOGIC - FIRST ORDER

- Từ các công thức phân tử, ta sử dụng các kết nối logic và các lượng từ để xây dựng các công thức (các câu) bằng đệ quy như sau:
  - Các công thức phân tử là các công thức.
  - Nếu G, H là các công thức thì các biểu thức logic của G và H là công thức.
  - Nếu G là một công thức và x là biến thì các biểu thức (∃ x G), (∀ x G) là công thức.
  - Các công thức không phải là công thức phân tử thì được gọi là các công thức phức hợp

PREDICATE LOGIC - FIRST ORDER

- Các công thức không chứa biến thì được gọi là các công thức cụ thể.
- Lượng tử phổ dụng ∀ cho phép ta mô tả một lớp các đối tượng.
- Lượng tử tồn tại ∃ cho phép ta nói đến một đối tượng nào đó trong một lớp đối tượng.
- Một công thức phân tử hoặc phủ định công thức phân tử đượcgọi là literal.
- Một công thức là tuyển của các literal được gọi là câu tuyển.

PREDICATE LOGIC - FIRST ORDER

- Một công thức mà các biến bắt buộc xuất hiện thì gọi là công thức đóng.
- Ý nghĩa của các công thức trong một thế giớ hiện thực nào đó thì được gọi là minh hoạ.
- Trong một minh hoạ, các ký hiệu vị từ sẽ được gắn với một thuộc tính hoặc một quan hệ cụ thể nào đó.
- Khi đã xác định được ngữ nghĩa một câu đơn, ta có thể xác định được ngữ nghĩa của các câu phức hợp.

PREDICATE LOGIC - FIRST ORDER

- Hai công thức tương đương nếu như nó cùng sai hoặc cùng đúng trong mọi minh hoạ.
- Từ các câu phân tự, bằng cách sử dụng các kết nối logic và các lượng tử, ta có thể tạo ra các câu phức hợp có câu trúc phức tạp. Để dễ dàng cho việc lưu trữ các câu trong bộ nhớ và thuận lợi cho việc xây dựng các thủ tục suy diễn, ta cần chuẩn hoá các câu bằng cách đưa chúng về dạng chuẩn tắc hội (hội của các câu tuyển).

PREDICATE LOGIC - FIRST ORDER

Thủ tục chuẩn hoá các công thức:

- Loại bỏ các kéo theo
- Chuyển các phủ định tới các phân tử
- Loại bỏ các lượng tử tồn tại
- Loại bỏ các lượng tử phổ dụng
- Chuyển các tuyển tới literals
- Loại bỏ các hội
- Đặt tên lại các biến

PREDICATE LOGIC - FIRST ORDER

Các luật suy diễn:

- Luật thay thế phổ dụng
- Hợp nhất
- Luật Modus Ponens tổng quát.
- Luật phân giải tổng quát
- Luật phân giải trên các câu Horn

PREDICATE LOGIC - FIRST ORDER

- Logic vị từ cấp 1 cho phép chúng ta biểu diễn các đối tượng trong thế giới thực với các tính chất của chúng và các quan hệ của chúng.
- Để biểu diễn tri thức của chúng ta về một miền các đối tượng nào đó trong logic vị từ cấp một, chúng ta cần đưa ra các ký hiệu hằng để chỉ ra các đối tượng cụ thể, các ký hiệu biến để chỉ ra các đối tượng bất kỳ trên miền đối tượng, các ký hiệu hàm để biểu diễn quan hệ hàm, các ký hiệu vị từ biểu diễn mối quan hệ khác nhau của các đối tượng.

PREDICATE LOGIC - FIRST ORDER

- Các ký hiệu đã nêu tạo thành hệ thống từ vựng về miến đối tượng mà chúng ta quan tâm.
- Sử dụng các từ vựng đã đưa ra, chúng ta sẽ tạo ra các câu trong logic vị từ cấp một để biểu diễn tri thức của chúng ta về miền đối tượng đó.
- Tập hợp tất cả các câu được tạo thành sẽ lập nên cở sở tri thức trong hệ tri thức mong muốn.
- Ngoài ra, có thể sử dụng vị từ bằng, danh sách và các phép toán trên danh sách và tập hợp để biểu diễn tri thức mong muốn

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Loại bỏ các kéo theo:
  - Để loại bỏ các kéo theo, ta chỉ cần thay thế:

$$P \rightarrow Q \equiv \bar{P} \vee Q$$

$$P \rightarrow Q \equiv (\overline{P} \vee Q) \wedge (P \vee \overline{Q})$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Chuyển các phủ định tới các phân tử:

$$\overline{(\overline{P})} \equiv P$$

$$\overline{(\forall x P)} \equiv \exists x, \overline{P}$$

$$\overline{(\exists x P)} \equiv \forall x, \overline{P}$$

$$\overline{(P \wedge Q)} \equiv \overline{P} \vee \overline{Q}$$

$$\overline{(P \vee Q)} \equiv \overline{P} \wedge \overline{Q}$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- <u>Loại bỏ các lượng tử tồn tại:</u>
    - Giả sử **P(x,y)** có nghĩa là ***x nhỏ hơn y***. Khi đó, **∀x, ∃y P(x,y)** có nghĩa là ***"Với mọi x, tồn tại y sao cho x nhỏ hơn y"***. Ta có thể **xem y như là một hàm của x**: y=f(x). Khi đó, ta có thể loại bỏ lượng tử ∃y và công thức trở thành: **∀x, P(x,f(x))**.
    - Ví dụ: $\forall x(\exists y P(x,y)) \vee \forall u(\exists v(Q(a,v) \wedge \exists y \overline{R(x,y)}))$

      sau khi loại bỏ lượng tử tồn tại trở thành:

$$P(x,f(x)) \vee (Q(a,g(u,x)) \wedge \overline{R(x,h(x,u))})$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Loại bỏ các lượng tử phổ dụng:

$$\forall x P(x, f(x)) \vee \forall u(Q(a, g(u, x)) \wedge \overline{R(x, h(x, u))})$$

trở thành

$$P(x, f(x)) \vee (Q(a, g(u, x)) \wedge \overline{R(x, h(x, u))})$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Chuyển các tuyển tới literal:
  - Thay các công thức dạng P ∨ (Q ∧ R) thành (P ∨ Q) ∧ (P ∨ R) và các công thức dạng (P ∧ Q) ∨ R thành (P ∨ R) ∧ (Q ∨ R)
  - Khi đó:

$$P(x, f(x)) \vee (Q(a, g(u, x)) \wedge \overline{R(x, h(x, u))})$$

sẽ được chuẩn hóa thành:

$$(P(x, f(x)) \vee Q(a, g(u, x))) \wedge (P(x, f(x)) \vee \overline{R(x, h(x, u))})$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Loại các hội
  - Một câu hội là đúng nếu tất cả các thành phần của nó đều đúng. Do đó, công thức ở chuẩn tắc hội tương đương các thành phần của nó.
  - Do đó:

$$(P(x, f(x)) \vee Q(a, g(u, x))) \wedge (P(x, f(x)) \vee \overline{R(x, h(x, u))})$$

tương đương hai câu tuyển:

$$P(x, f(x)) \vee Q(a, g(u, x)) \text{ và } P(x, f(x)) \vee \overline{R(x, h(x, u))}$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Đặt tên lại các biến:
  - Đặt tên lại các biến sao cho biến ở hai câu khác nhau thì khác nhau.
  - Ví dụ:

$$P(x, f(x)) \vee Q(a, g(u, x)) \text{ và } P(x, f(x)) \vee \overline{R(x, h(x, u))}$$

có cùng tên biến là x, ta có thể đặt tên lại:

$$P(x, f(x)) \vee Q(a, g(u, x)) \text{ và } P(z, f(z)) \vee \overline{R(z, h(z, u))}$$

PREDICATE LOGIC

Thủ tục chuẩn hóa các công thức

- Khi tri thức là tập hợp nào đó các công thức trong logic vị từ, bằng cách áp dụng các thủ tục vừa nêu, chúng ta xây dựng được cơ sở tri thức chỉ gồm các câu tuyển.
- Tương tư như logic mệnh đề, các câu tuyển có thể biểu diễn dưới dạng các **câu Kowalski**:

$$P_1 \wedge P_2 \wedge \cdots \wedge P_n \rightarrow Q_1 \vee Q_2 \vee \cdots \vee Q_m$$

- Một trường hợp đặc biệt của câu Kowalski là **câu Horn** (if...then)

PREDICATE LOGIC

Các luật suy diễn

- Trong logic mệnh đề, ngoài những luật quan trọng như luật Modus Ponens, luật Modus Tolens, bắc cầu..., ta đã chỉ ra được luật phân giải là luật đầy đủ cho chứng minh bác bỏ. Ta sẽ mở rộng kết quả này cho logic vị từ.
- Tất cả các luật suy diễn được đưa ra trong logic mệnh đề đều đúng trong logic vị từ cấp một.

PREDICATE LOGIC

Luật thay thế phổ dụng

- Là luật suy diễn quan trọng trong logic vị từ.
- Giả sử A là một câu, câu $\forall x$ A là đúng trong một minh họa nào đó nếu và chỉ nếu A đúng đối với tất cả các đối tượng nằm trong miền đối tượng của minh họa đó.
- Mỗi hạng thức t ứng với một đối tượng khi thế vào biến x trong câu $\forall x$ A sẽ cho ra câu đúng nếu $\forall x$ A đúng. Công thức nhận được từ công thức A bằng cách thay thế tất cả các xuất hiện của biến x bởi t ký hiệu là A[x/t].

PREDICATE LOGIC

Luật thay thế phổ dụng

- **Luật thay thế phổ dụng** (*universal instatiation*): từ công thức ∀x A, ta suy ra công thức A[x/t].
- Ví dụ:
  - ∀x Like(x, ăn chè) có nghĩa là một người đều thích ăn chè.
  - Thay x = Khoa, ta suy ra Like(Khoa, ăn chè) nghĩa là Khoa thích ăn chè.

PREDICATE LOGIC

Luật hợp nhất

- Giả sử ta có hai câu phân tử:

  ∀y Like(Nam,y) và ∀x Like(x, Football),

  Bằng cách sử dụng phép thế [x/Nam,y/Football], ta có thể hợp nhất hai câu trên thành Like(Nam,Football).
- Trong các suy diễn, ta cần sử dụng phép hợp nhất các câu bằng phép thế như ví dụ trên.

PREDICATE LOGIC

Luật hợp nhất

- Ví dụ: cho trước hai câu sau đây:
    - $\forall x$, Friend(x,Nam) → Good(x): Mọi người bạn của Nam đều tốt
    - $\forall y$, Friend(Lan,y): Lan là bạn của tất cả mọi người
- Ta có thể hợp nhất hai câu trên bằng cách thay thế [x/Lan,y/Nam]. Áp dụng luật thay thế phổ dụng, ta sinh ra hai câu mới:

  Friend(Lan,Nam) → Good(Lan) và Friend(Lan,Nam)

  Từ hai câu trên, theo luật Modus Ponens, ta suy ra Good(Lan).

PREDICATE LOGIC

Luật hợp nhất

- Không mất tính tổng quát, ta gọi phép thế θ là một dãy các cặp

$$x_i/t_i, \quad \theta = [x_1/t_1, \ldots, x_n/t_n]$$

- Trong đó x_i là các biến khác nhau, t_i là các hạng thức và x_i không có mặt trong t_i.
- Áp dụng phép thế θ vào công thức A, ta được công thức A_θ.
- Hai công thức phân tử A và B mà tồn tại phép thế θ sao cho A_θ=B_θ được gọi là hợp nhất được và phép thế θ được gọi là hợp nhất tử của A và B