# Logic bậc nhất

Tô Hoài Việt
Khoa Công nghệ Thông tin
Đại học Khoa học Tự nhiên TPHCM
thviet@fit.hcmuns.edu.vn

# Tổng quan

- Logic bậc nhất (First Order Logic)
- Cú pháp và ngữ nghĩa
- Các lượng từ
- Hợp giải với logic vị từ
- Phép thế
- Thuật giải đồng nhất

# Tại sao sử dụng logic bậc nhất?

- Logic mệnh đề chỉ xử lý trên các sự kiện, có giá trị đúng hoặc sai, ví dụ “trời mưa”, “Tuấn đi xem đá banh”… Ta không thể dùng các biến để đại diện cho nhiệt độ, con người,…
- Trong logic bậc nhất, các biến giúp ta tham chiếu đến các sự vật trong thế giới và ta còn có thể lượng hoá chúng: tức là xem xét toàn bộ hay một phần của sự vật.

# Logic Bậc nhất

- Các câu không thể biểu diễn bằng logic mệnh đề nhưng có thể bằng logic bậc nhất
  - Socrates là người nên socrates chết
  - Khi sơn một hộp bằng màu xanh, nó sẽ trở thành hộp xanh
  - Một người được cho phép truy cập trang web nếu họ được cấp quyền chính thức hay quen biết với ai được phép truy cập

# Cú pháp của FOL

- Biểu thức (Term)
  - Ký hiệu hằng: Lan, Tuan, DHKHTN,…
  - Biến: x, y, a,…
  - Ký hiệu hàm áp dụng cho một hay nhiều term: f(x), tuoi(Lan), anh-cua(Tuan)…
- Câu (Sentence)
  - Một ký hiệu vị từ (predicate) áp dụng cho một hay nhiều *term*: Thuoc(Lan, DHKHTN), La-anh-em(Tuan, Lan), La-ban-be(anh-cua(Tuan), Lan),…
  - t1= t2
  - Nếu v là một biến và $\phi$ là một câu thì $\forall x.\ \phi$ và $\exists x.\ \phi$ là một câu
  - Đóng với các toán tử nối câu: $\neg\ \wedge\ \vee\ \leftarrow\ \leftrightarrow\ \rightarrow$

# Trị đúng trong Logic bậc nhất

- Các câu là đúng ứng với một mô hình và một thể hiện

- Mô hình chứa các đối tượng (các thành phần) và quan hệ giữa chúng

- Thể hiện xác định các tham chiếu cho
  các ký hiệu hằng $\rightarrow$ các đối tượng
  các ký hiệu vị từ $\rightarrow$ các quan hệ
  các ký hiệu hàm $\rightarrow$ các quan hệ hàm

- Một câu nguyên tố *predicate*$(term_1, term_2, \ldots term_n)$ là đúng nếu và chỉ nếu các đối tượng được tham chiếu bởi $term_1, term_2, \ldots term_n$ nằm trong quan hệ được tham chiếu bởi *predicate*

# Lượng từ với mọi

- $\forall$<*biến*> <*câu*>

Sinh viên CNTT thì thông minh:

$\forall$x Sinh-viên(x,CNTT) $\rightarrow$ Thông-minh(x)

- $\forall$x *P* đúng trong một mô hình *m* nếu và chỉ nếu *P* đúng với *x* trong mọi đối tượng có thể của mô hình
- Nghĩa là, tương đương với phép nối liền của các thể hiện của *P*

Sinh-viên(Lan,CNTT) $\rightarrow$ Thông-minh(Lan)

$\wedge$ Sinh-viên(Minh,CNTT) $\rightarrow$ Thông-minh(Minh)

$\wedge$ Sinh-viên(Tuấn,CNTT) $\rightarrow$ Thông-minh(Tuấn)

$\wedge$ …

# Lỗi thường gặp cần tránh

- Thông thường, $\rightarrow$ là phép nối thường đi với $\forall$
- Lỗi thường gặp: dùng $\wedge$ làm phép nối chính đi với $\forall$:

$$\forall x \text{ Sinh-viên}(x, CNTT) \wedge \text{Thông-minh}(x)$$

nghĩa là “Mọi người đều là sinh viên CNTT và mọi người đều thông minh”

# Lượng từ Tồn tại

- ∃<*biến*> <*câu*>

Có sinh viên CNTT thông minh:

∃x Sinh-viên(x,CNTT) ∧ Thông-minh(x)

- ∃x.*P* đúng trong một mô hình *m* nếu và chỉ nếu *P* đúng với *x* trong một đối tượng có thể nào đó của mô hình
- Nghĩa là, tương đương với phép nối rời của các thể hiện của *P*

  Sinh-viên(Lan,CNTT) ∧ Thông-minh(Lan)

∨ Sinh-viên(Minh,CNTT) ∧ Thông-minh(Minh)

∨ Sinh-viên(Tuấn,CNTT) ∧ Thông-minh(Tuấn)

∨ …

# Lỗi thường gặp khác cần tránh

- Thông thường, $\wedge$ là phép nối chính với $\exists$
- Lỗi thường gặp: dùng $\rightarrow$ làm phép nối chính với $\exists$:

$$\exists x\ \text{Sinh-viên}(x,\text{CNTT}) \rightarrow \text{Thông-minh}(x)$$

đúng nếu có bất kỳ ai không là sinh viên CNTT!

# Viết FOL như thế nào

- Mèo là động vật có vú [Mèo$^1$, Động-vật-có-vú$^1$]

  $\forall x.\text{Mèo}(x) \rightarrow \text{Động-vật-có-vú}(x)$
- Lan là sinh viên học giỏi [Sinh-viên$^1$, Học-giỏi$^1$,Lan]

  $\text{Sinh-viên}(\text{Lan}) \wedge \text{Học-giỏi}(\text{Lan})$
- Cháu là con của anh em [Cháu$^2$, Anh-em$^2$, Con$^2$]

  $\forall x,y.\text{Cháu}(x,y) \leftrightarrow \exists z.(\text{Anh-em}(z,y) \wedge \text{Con}(x,z))$
- Bà ngoại là mẹ của mẹ [các hàm: bà-ngoại, mẹ]

  $\forall xy.\ x = \text{bà-ngoại}(y) \leftrightarrow \exists z.(x = \text{mẹ}(z) \wedge z = \text{mẹ}(y))$
- Mọi người đều yêu ai đó [Yêu$^2$]

  $\forall x, \exists y.\text{Yêu}(x, y)$

  $\exists x, \forall y.\text{Yêu}(x, y)$

# Viết FOL như thế nào (tt)

- Không ai yêu Lan

  $\forall x. \neg Yêu(x, Lan)$

  $\neg\exists x. Yêu(x,Lan)$

- Ai cũng có một cha

  $\forall x, \exists y. Cha(y,x)$

- Ai cũng có một cha và một mẹ

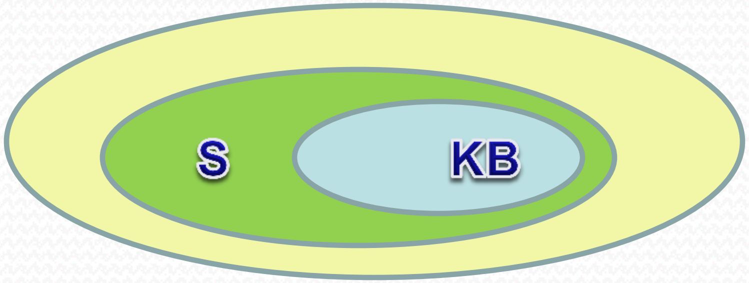
  $\forall x, \exists yz. Cha(y,x) \wedge Mẹ(z,x)$

- Bất kỳ ai có một cha cũng có một mẹ

  $\forall x. [[\exists y.Cha(y,x)] \rightarrow [\exists y.Mẹ(y,x)]]$

# Suy dẫn trong FOL

- KB suy dẫn S: với mọi thể hiện I, nếu KB thoả trong I thì S cũng thoả trong I




- Nói chung tính toán suy dẫn là không khả thi vì có nhiều vô số thể hiện có thể.
- Ngay cả việc tính toán tính thoả cũng không khả thi đối với các thể hiện có tập hợp vô hạn

# Chứng minh và Suy dẫn

- Suy dẫn xuất phát từ khái niệm tổng quát của phép "kéo theo"
- Không thể tính toán trực tiếp bằng cách liệt kê khái niệm
- Do đó, ta sẽ làm theo cách chứng minh
- Trong FOL, nếu KB suy dẫn được S thì có một tập hữu hạn các chứng minh của S từ KB

# Hợp giải Bậc nhất

$\forall x, P(x) \rightarrow Q(x)$

$P(A)$

---

$Q(A)$

Tam đoạn luận:
Mọi người đều chết
Socrates là người

---

Socrates chết

Hai vấn đề mới:

- biến đổi FOL thành dạng mệnh đề (clausal form)
- hợp giải với biến

$\forall x, \neg P(x) \vee Q(x)$

$P(A)$

---

$Q(A)$

Tương đương theo định nghĩa của phép Suy ra

$\neg P(A) \vee Q(A)$

$P(A)$

---

$Q(A)$

Thay A vào x, vẫn đúng

khi đó

Hợp giải Mệnh đề

# Dạng mệnh đề (Clausal Form)

- cấu trúc ngoài giống CNF
- không có lượng từ

$$\forall x.\ \exists y.\ P(x) \Rightarrow R(x,y)$$

$$\Downarrow$$

$$\neg P(x) \vee R(x,y)$$

# Biến đổi thành dạng mệnh đề

1. Loại bỏ các dấu mũi tên

$$\alpha \leftrightarrow \beta \Rightarrow (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$$

$$\alpha \rightarrow \beta \Rightarrow \neg\alpha \vee \beta$$

2. Phân phối phủ định

$$\neg\neg\alpha \Rightarrow \alpha$$

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

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

$$\neg\forall x.\ \alpha \Rightarrow \exists x.\neg\alpha$$

$$\neg\exists x.\ \alpha \Rightarrow \forall x.\neg\alpha$$

3. Đổi tên các biến thành phần

$\forall x.\exists y.(\neg P(x) \vee \exists x.\ Q(x,y)) \Rightarrow \forall x_{1.}\exists y_2.(\neg P(x_1) \vee \exists x_3.\ Q(x_3,y_2))$

# Skolem hoá

4. Skolem hoá
   - thay tên mới cho tất cả lượng từ tồn tại

$$\exists x.\ P(x) \Rightarrow P(Lan)$$

$$\exists x,y.R(x,y) \Rightarrow R(Thing1, Thing2)$$

$$\exists x.\ P(x) \wedge Q(x) \Rightarrow P(Fleep) \wedge Q(Fleep)$$

$$\exists x.\ P(x) \wedge \exists x.\ Q(x) \Rightarrow P(Frog) \wedge Q(Grog)$$

$$\exists y, \forall x.\ Loves(x,y) \Rightarrow \forall x.Loves(x, Englebert)$$

   - thay hàm mới cho tất cả các lượng từ tồn tại ở tầm vực ngoài

$$\forall x\ \exists y.\ Loves(x,y) \Rightarrow \forall x.Loves(x, beloved(x))$$

# Biến đổi thành dạng mệnh đề

5. Bỏ các lượng từ với mọi

$$\exists y, \forall x.\ Loves(x,y) \Rightarrow Loves(x, beloved(x))$$

6. Phần phối *or* vào *and;* trả về các mệnh đề

$$P(z) \vee (Q(z,w) \wedge R(w,z)) \Rightarrow$$

$$\{P(z) \vee Q(z,w), P(z) \vee Q(w,z)\}$$

7. Đổi tên các biến trong từng mệnh đề

$$\{P(z) \vee Q(z,w), P(z) \vee Q(w,z)\} \Rightarrow$$

$$\{P(z_1) \vee Q(z_1,w_1), P(z_2) \vee Q(w_2,z_2)\}$$

# Hợp giải Bậc nhất

$$\frac{\forall x, P(x) \rightarrow Q(x) \quad P(A)}{Q(A)}$$

Tam đoạn luận:
Mọi người đều chết
Socrates là người

---

Socrates chết

$$\frac{\forall x, \neg P(x) \vee Q(x) \quad P(A)}{Q(A)}$$

Tương đương theo định nghĩa của phép Suy ra

Điều chủ yếu là tìm các phép thế đúng đắn cho các biến

$$\frac{\neg P(A) \vee Q(A) \quad P(A)}{Q(A)}$$

Thay A vào x, vẫn đúng

khi đó

Hợp giải Mệnh đề

# Phép thế

P(x, f(y), B): một câu nguyên tố

| **Các thể hiện** | **Phép thế** $\{v_1/t_1, v_2/t_2 \ldots\}$ | **Ghi chú** |
|---|---|---|
| P(z, f(w), B) | {x/z, y/w} | Đổi tên biến |
| P(x, f(A), B) | {y/A} | |
| P(g(z), f(A), B) | {x/g(z), y/A} | |
| P(C, f(A), B) | {x/C, y/A} | Phép thế cơ sở |

# Phép thế

P(x, f(y), B): một câu nguyên tố

| **Các thể hiện** | **Phép thế** **$\{v_1/t_1, v_2/t_2 \ldots\}$** | **Ghi chú** |
|---|---|---|
| P(z, f(w), B) | {x/z, y/w} | Đổi tên biến |
| P(x, f(A), B) | {y/A} | |
| P(g(z), f(A), B) | {x/g(z), y/A} | |
| P(C, f(A), B) | {x/C, y/A} | Phép thế cơ sở |

Áp dụng một phép thế

P(x, f(y), B) {y/A} = P(x, f(A), B)

P(x, f(y), B) {y/A, x/y} = P(A, f(A), B)

# Đồng nhất

- Hai biểu thức $\omega_1$ và $\omega_2$ là đồng nhất được (unifiable) khi vào chỉ khi tồn tại thế s sao cho $\omega_1 s = \omega_2 s$
- Gọi ω1 = x và ω2 = y, dưới đây là các phép đồng nhất:

| s | $\omega_1 s$ | $\omega_2 s$ |
|---|---|---|
| {y/X} | x | X |
| {x/Y} | Y | y |
| {x/f(f(A)), y/f(f(A))} | f(f(A)) | f(f(A)) |
| {x/A, y/A} | A | A |

# Đồng nhất tổng quát nhất

- Để đồng nhất *Knows(John,x)* và *Knows(y,z)*, ta có các phép thế

  θ = {y/John, x/z } hay θ = {y/John, x/John, z/John}

- Phép đồng nhất đầu tiên tổng quát hơn cái thứ hai.

# Đồng nhất tổng quát nhất

- g là phép đồng nhất tổng quát nhất (most general unifier - MGU) của $\omega_1$ và $\omega_2$ khi và chỉ khi với mọi phép đồng nhất s, tồn tại s' sao cho $\omega_1.s = (\omega_1.g)s'$ và $\omega_2.s = (\omega_2.g)s'$

| $\omega_1$ | $\omega_2$ | MGU |
|---|---|---|
| P(x) | P(A) | {x/A} |
| P(f(x), y, g(x)) | P(f(x), x, g(x)) | {y/x} hay {x/y} |
| P(f(x), y, g(y)) | P(f(x), z, g(x)) | {y/x, z/x} |
| P(x, B, B) | P(A, y, z) | {x/A, y/B, z/B} |
| P(g(f(v)), g(u)) | P(x, x) | {x/g(f(v)), u/f(v)} |
| P(x, f(x)) | P(x, x) | Không có MGU! |

# Thuật toán đồng nhất

```
unify(Expr x, Expr y, Subst s){
    if s = fail, return fail
    else if x = y, return s
    else if x là một biến, return unify-var(x, y, s)
    else if y là một biến, return unify-var(y, x, s)
    else if x là một vị từ hay một hàm,
        if y có cùng toán tử,
            return unify(args(x), args(y), s)
        else    return fail
    else                              ; x và y là các danh sách
        return unify(rest(x), rest(y), unify(first(x), first(y), s))
    return fail;
}
```

# Thủ tục đồng nhất biến

Thế vào var và x khi còn có thể, tiếp đó thêm vào ràng buộc mới.

```
unify-var(Variable var, Expr x, Subst s){
        if var đã được gắn với giá trị val trong s,
                return unify(val, x, s)
        else if x được gắn với giá trị val trong s,
                return unify(var, val, s)
        else if var xuất hiện đâu đó trong x, return fail
        else    return add({var/x}, s)
}
```

# Một số ví dụ về đồng nhất

| $\omega_1$ | $\omega_2$ | **MGU** |
|---|---|---|
| A(B,C) | A(x,y) | {x/B, y/C} |
| A(x, f(D,x)) | A(E, f(D, y)) | {x/E, y/E} |
| A(x, y) | A(f(C, y), z) | {x/f(C,y), y/z} |
| P(A, x, f(g(y))) | P(y, f(z), f(z)) | {y/A, x/f(z), z/f(g(y)} |
| P(x, g(f(A)), f(x)) | P(f(y), z, y) | Không có |
| P(x, f(y)) | P(z, g(w)) | Không có |

# Điều cần nắm

- Ý nghĩa của logic bậc nhất
- Cú pháp của logic bậc nhất
- Có thể biểu diễn một vấn đề dưới dạng logic bậc nhất
- Chuyển đổi KB thành dạng mệnh đề
- Hiểu được phép thế và phép đồng nhất