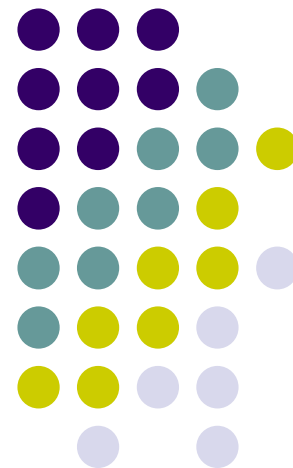


Phân tích ngữ nghĩa

Lê Thanh Hương
Bộ môn Hệ thống Thông tin
Viện CNTT & TT – Trường ĐHBKHN
Email: huonglt@soict.hust.edu.vn



Định nghĩa



- PTNN là quá trình ánh xạ từ câu NNTN sang dạng biểu diễn logic của nó.
- Mức nông: gán nhãn vai trò ngữ nghĩa (case-role) như agent, patient
- Mức sâu hơn: biểu diễn dạng logic vị từ hoặc hình thức khác cho phép suy diễn tự động.



Tại sao cần phân tích ngữ nghĩa

- Trả lời câu hỏi (bài toán hỏi đáp)
- Hội thoại (chatbot)
- Điều khiển hoạt động của robot
- Dịch máy
- Tóm lược văn bản
- ...

Thế nào được coi là hiểu?



- ... nếu có thể phản ứng phù hợp
 - VD: “cho tất cả đồ chơi vào giỏ”
- ... nếu có thể xác định 1 phát biểu là đúng hay sai
 - hiểu NP nghĩa là xác định được NP đó đề cập đến cái gì



Thế nào được coi là hiểu?

- ... nếu có thể sử dụng nội dung đó để trả lời câu hỏi
 - Dễ: Mai ăn kẹo. → Mai ăn gì?
 - Khó: Nước đi đầu tiên của quân trắng là P-Q4. → Quân đen có thể chiếu tướng không?
- ... nếu có thể dịch: phụ thuộc vào ngôn ngữ đích
 - Anh – Anh?
 - Anh – Pháp? có thể được
 - Anh – logic ? cần hiểu sâu
 - tất cả loài cá đều biết bơi
 - = $\forall x [\text{fish}(x) \Rightarrow \text{can_swim}(x)]$



Một số khái niệm cơ bản về logic

3 loại đối tượng cơ bản:

1. Giá trị - Booleans

- là giá trị ngữ nghĩa của câu

2. Thực thể - Entities

- Giá trị của NP, vd., các đối tượng như bàn, ghế, thời gian

3. Hàm

- Hàm trả về giá trị nhị phân gọi là vị từ (predicate). Vd., `frog(x)`, `green(x)`
- Hàm có thể trả về 1 hàm khác
- Hàm có thể nhận hàm khác như tham số



Logic: thuật ngữ Lambda

- λ :
 - là cách để viết 1 hàm “**bất kỳ**”
 - Không có tên hàm
 - Được dùng để định nghĩa tính chất cơ bản của hàm
 - Cho `square = λp p*p`
 - Tương đương với `int square(p) { return p*p; }`
 - Định dạng: λ <tên biến> <biểu thức>



Logic: thuật ngữ Lambda

- λ :
 - Cho $\text{square} = \lambda p \ p^*p$
 - Khi đó $\text{square}(3) = (\lambda p \ p^*p)(3) = 3^*3$
 - **Chú ý: $\text{square}(x)$ không phải là hàm, chỉ là giá trị của x^*x .**
 - Hàm: $\lambda x \ \text{square}(x) = \lambda x \ x^*x = \lambda p \ p^*p = \text{square}$

- Cho $\text{even} = \lambda p \ (p \bmod 2 == 0)$ vị từ trả về giá trị True/False
- $\text{even}(x) = \text{true}$ nếu x chẵn
- Còn $\text{even}(\text{square}(x))$?
- $\lambda x \ \text{even}(\text{square}(x)) = \text{true}$ với các số x có $\text{square}(x)$ chẵn
 - $\lambda x \ (\text{even}(x^*x)) = \lambda x \ (x^*x \bmod 2 == 0)$



Logic: Một số vị từ

- **most** – 1 vị từ trên 2 vị từ khác
 - **most(pig, big)** = “most pigs are big”
 - tương đương với, **most(λx pig(x), λx big(x))**
 - trả về true nếu đa số giá trị thỏa vị từ đầu tiên cũng thỏa vị từ thứ 2
- tương tự:
 - **all(pig, big)** (tương đương với $\forall x$ pig(x) \Rightarrow big(x))
 - **exists(pig, big)** (tương đương với $\exists x$ pig(x) AND big(x))



Cách biểu diễn vị từ

- Gilly swallowed a goldfish
 - `swallowed(Gilly, goldfish)`
- `goldfish` không phải là tên một đối tượng duy nhất như là Gilly
- Vấn đề
Gilly swallowed a goldfish and Milly swallowed a goldfish
chuyển thành:
`swallowed(Gilly, goldfish) AND swallowed(Milly, goldfish)`
nhưng không phải cùng 1 goldfish ...



Sử dụng lượng từ

- Gilly swallowed a goldfish
 - `swallowed(Gilly, goldfish)`
- Tốt hơn: $\exists g$ goldfish(g) AND `swallowed(Gilly, g)`
- Hoặc sử dụng vị từ lượng từ:
 - `exists(λg goldfish(g), λg swallowed(Gilly,g))`
 - tương đương: `exists(goldfish, swallowed(Gilly))`
 - “trong tập cá có 1 con cá bị nuốt bởi Gilly”

- Mai likes small cats.
- Mai likes the cat whose name is Tom.

Thời



- Gilly swallowed a goldfish
 - $\text{exists}(\text{goldfish}, \lambda g \text{ swallowed}(\text{Gilly}, g))$
- Cải tiến:
 - $\text{swallowed}(\text{Gilly}, g)$ chuyển thành $\text{swallow}(t, \text{Gilly}, g)$, t là thời gian
 - $\exists t \text{ past}(t) \text{ AND } \text{exists}(\lambda g \text{ goldfish}(g), \lambda g \text{ swallow}(t, \text{Gilly}, g))$
 - “Có lúc nào đó trong quá khứ, con cá là 1 trong các vật bị Gilly nuốt”



Các thuộc tính về sự kiện

- Gilly swallowed a goldfish
 - $\exists t \text{ past}(t) \text{ AND exists}(\text{goldfish}, \text{swallow}(t, \text{Gilly}))$
- Một phát biểu có thể có các thuộc tính khác nữa:
 - [Gilly] swallowed [a goldfish] [on a dare] [in a telephone booth] [with 30 other freshmen] [after many bottles of vodka had been consumed].
 - xác định ai? làm gì? tại sao? khi nào? (who what why when)
- Thay biến thời gian t với biến sự kiện e
 - $\exists e \text{ past}(e), \text{act}(e, \text{swallowing}), \text{swallower}(e, \text{Gilly}), \text{exists}(\text{goldfish}, \text{swallowee}(e)), \text{exists}(\text{booth}, \text{location}(e)), \dots$
 - Có thể biểu diễn quá khứ bằng $\lambda e \exists t \text{ before}(t, \text{now}), \text{ended-at}(e, t)$



Trật tự của lượng từ

- Ví dụ
 - In this country a woman gives birth every 15 min. Our job is to find that woman and stop her.
 - $\exists \text{woman} (\forall 15\text{min gives-birth-during}(\text{woman}, 15\text{min}))$
 - $\forall 15\text{min} (\exists \text{woman gives-birth-during}(15\text{min}, \text{woman}))$



Các thuộc tính về ý chí

- Willy wants a unicorn
 - $\exists e \text{ act}(e, \text{wanting}), \text{wanter}(e, \text{Willy}), \text{exists}(\text{unicorn}, \lambda u \text{ wantee}(e, u))$
 - “there is a unicorn u that Willy wants”
 - wantee là một cá thể
 - $\exists e \text{ act}(e, \text{wanting}), \text{wanter}(e, \text{Willy}), \text{wantee}(e, \lambda u \text{ unicorn}(u))$
 - “Willy wants any entity u that satisfies the unicorn predicate”
 - wantee là 1 loại thực thể
- Willy wants Lilly to get married
 - $\exists e \text{ present}(e), \text{act}(e, \text{wanting}), \text{wanter}(e, \text{Willy}), \text{wantee}(e, \lambda e' [\text{act}(e', \text{marriage}), \text{marrier}(e', \text{Lilly})])$
 - “Willy wants any event e' in which Lilly gets married”
 - Sentence doesn't claim that such an event exists
- Các động từ chỉ ý chí: hope, doubt, believe, ...

Danh ngữ



- Expert $\lambda_g \text{ expert}(g)$
- big fat expert $\lambda_g \text{ big}(g), \text{ fat}(g), \text{ expert}(g)$
- Baltimore expert (white-collar expert, TV expert ...)
 - $\lambda_g \text{ Related}(\text{Baltimore}, g), \text{ expert}(g) - \text{expert from Baltimore}$

Câu



- Ý nghĩa của câu là gì?
 - Phụ thuộc vào dấu chấm câu.
 - Billy likes Lili. → `assert(like(B,L))`
 - Billy likes Lili? → `ask(like(B,L))`
 - Hoặc chính xác hơn, “Does Billy like Lili?”
 - Billy, like Lili! → `command(like(B,L))`

Câu



- What did Gilly swallow?
 - $\text{ask}(\lambda x \exists e \text{ past}(e), \text{act}(e, \text{swallowing}), \text{swallower}(e, \text{Gilly}), \text{swallowee}(e, x))$
- Eat your fish!
 - $\text{command}(\lambda f \text{ act}(f, \text{eating}), \text{eater}(f, \text{Hearer}), \text{eatee}(\dots))$
- I ate my fish.
 - $\text{assert}(\exists e \text{ past}(e), \text{act}(e, \text{eating}), \text{eater}(e, \text{Speaker}), \text{eatee}(\dots))$

Phân tích ngữ nghĩa câu



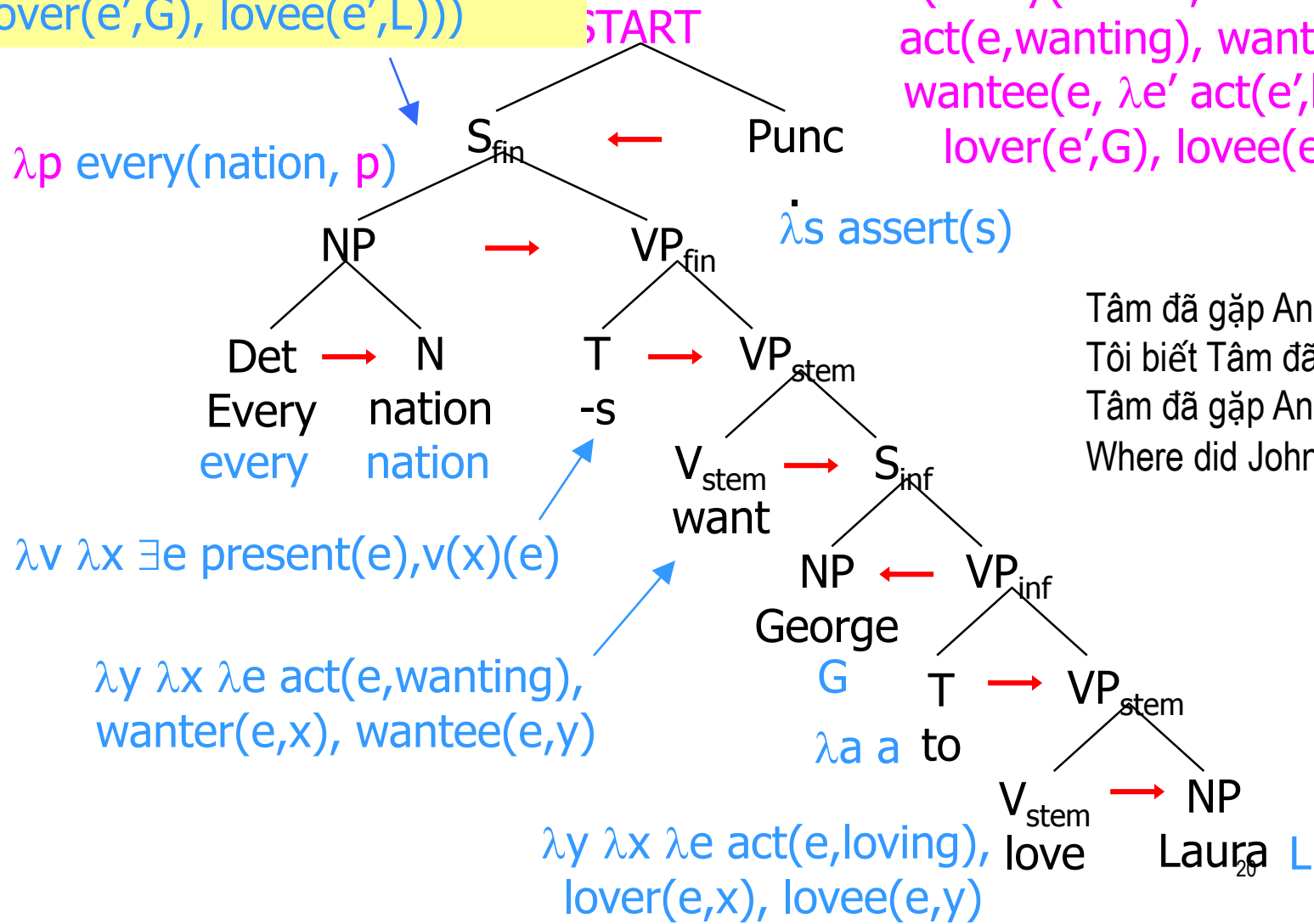
1. Phân tích cú pháp
2. Tìm ngữ nghĩa của từng từ
3. Xác định ngữ nghĩa cho mỗi thành phần ngữ pháp, thực hiện từ dưới lên



every(nation, $\lambda x \exists e$ present(e),
 act(e,wanting), wanter(e,x),
 wantee(e, $\lambda e'$ act(e',loving),
 lover(e',G), lovee(e',L)))

phần

assert(every(nation, $\lambda x \exists e$ present(e),
 act(e,wanting), wanter(e,x),
 wantee(e, $\lambda e'$ act(e',loving),
 lover(e',G), lovee(e',L))))



Tâm đã gặp An.
 Tôi biết Tâm đã gặp An.
 Tâm đã gặp An ở trường.
 Where did John see Mary?



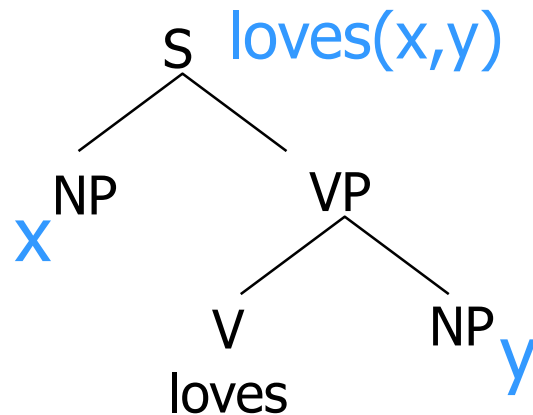
- Where did John see Mary?

```
(ROOT
  (SBARQ
    (WHADVP (WRB Where) )
    (SQ (VBD did)
      (NP (NNP John) )
      (VP (VB see)
        (NP (NNP Mary) ) ) )
    (. ?) ) ) )
```



Ngữ nghĩa thành phần

- Thêm thuộc tính “sem” cho mỗi luật phi ngữ cảnh
 - $S \rightarrow NP \text{ loves } NP$
 - $S[\text{sem}=\text{loves}(x,y)] \rightarrow NP[\text{sem}=x] \text{ loves } NP[\text{sem}=y]$
 - Nghĩa của S phụ thuộc vào nghĩa của NP
- TAG version:



- Điền mẫu: $S[\text{sem}=\text{showflights}(x,y)] \rightarrow$
I want a flight from $NP[\text{sem}=x]$ to $NP[\text{sem}=y]$



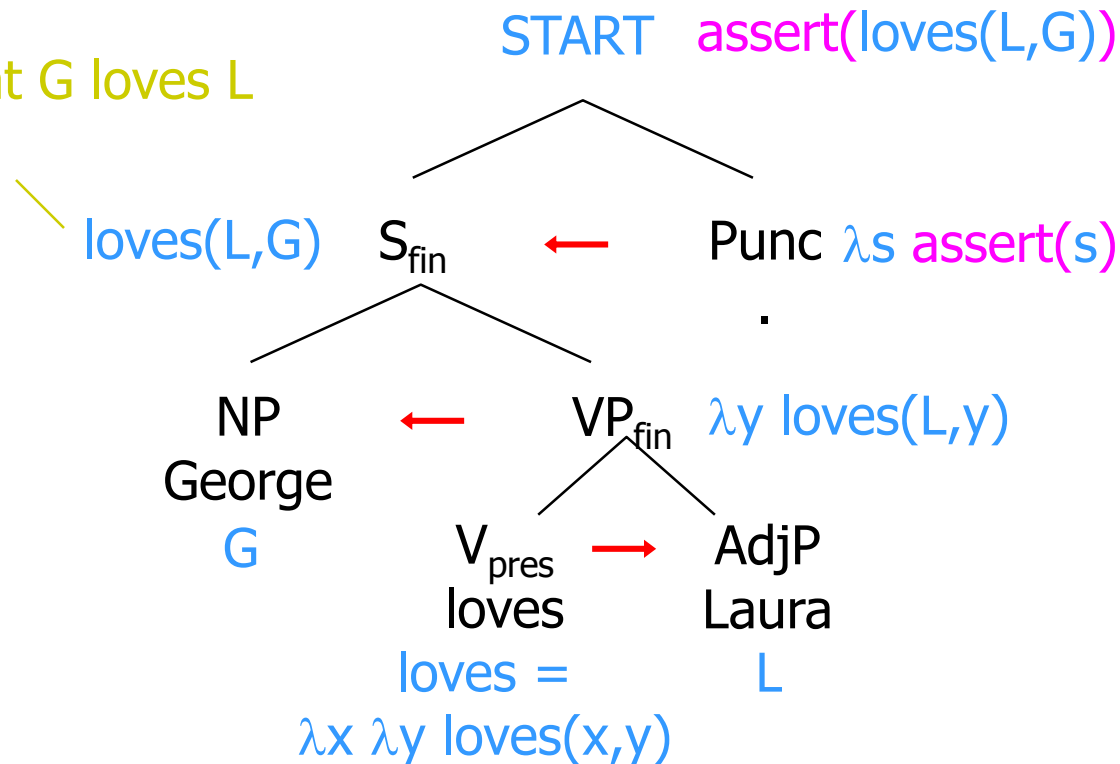
Ngữ nghĩa thành phần

- Thay $S \rightarrow NP \text{ loves } NP$
 - $S[\text{sem}=\text{loves}(x,y)] \rightarrow NP[\text{sem}=x] \text{ loves } NP[\text{sem}=y]$
- Luật tổng quát $S \rightarrow NP VP$:
 - $V[\text{sem}=\text{loves}] \rightarrow \text{loves}$
 - $VP[\text{sem}=\text{v}(\text{obj})] \rightarrow V[\text{sem}=\text{v}] NP[\text{sem}=\text{obj}]$
 - $S[\text{sem}=\text{vp}(\text{subj})] \rightarrow NP[\text{sem}=\text{subj}] VP[\text{sem}=\text{vp}]$
- George loves Laura **CÓ** $\text{sem}=\text{loves}(\text{Laura})(\text{George})$
- Trong phần này, ta:
 - tính ngữ nghĩa từ dưới lên
 - Ngữ pháp ở dạng chuẩn Chomsky
 - Mỗi nút có 2 con: 1 hàm và 1 tham số
 - **Để lấy ngữ nghĩa của nút, áp dụng hàm vào tham số**

Ngữ nghĩa thành phần



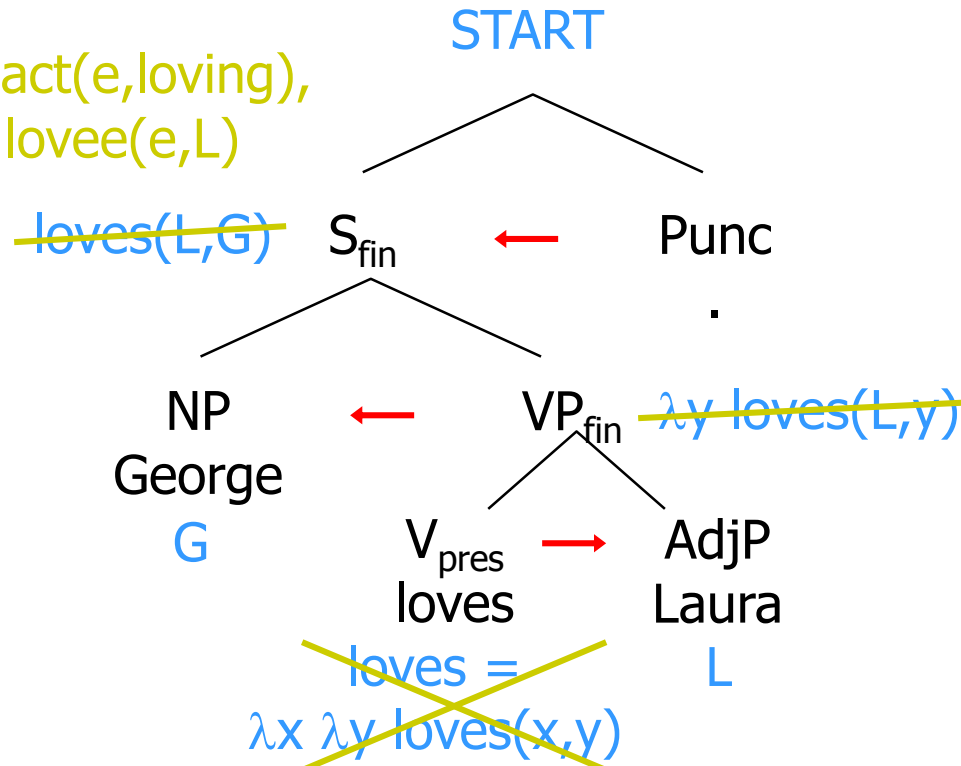
Muốn diễn đạt G loves L



Ngữ nghĩa thành phần



$\exists e \text{ present}(e), \text{ act}(e, \text{loving}),$
 $\text{lover}(e, G), \text{ lovee}(e, L)$



$\lambda y \exists e \text{ present}(e),$
 $\text{ act}(e, \text{loving}),$
 $\text{lover}(e, y), \text{ lovee}(e, L)$

$\lambda x \lambda y \exists e \text{ present}(e),$
 $\text{ act}(e, \text{loving}),$
 $\text{lover}(e, y), \text{ lovee}(e, x)$



Cách biểu diễn ngữ nghĩa cơ bản

- Sử dụng “Event”
- `(EVENT :condition1 val1 :condition2 val2...
:condn valn)`
- Ví dụ:
- `(see :agent John :patient Mary :tense past)`



Luật cú pháp/ngữ nghĩa

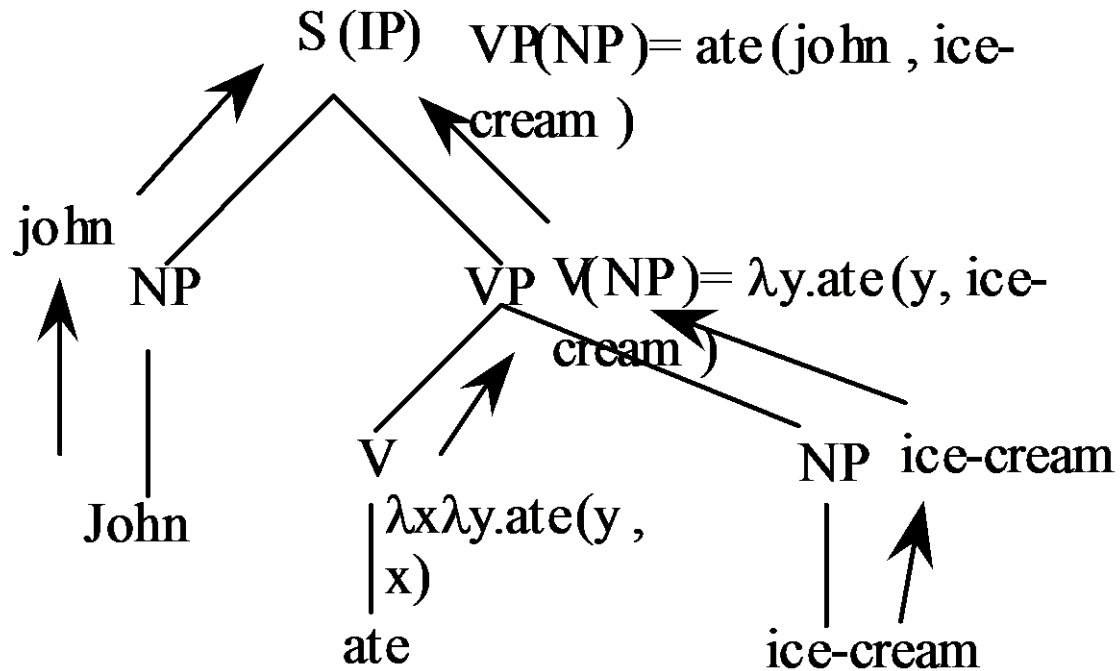
<u>Thành phần/luật</u>	<u>Dịch nghĩa</u>
Verb <i>ate</i>	$\lambda x \lambda y. ate(y, x)$
N	N
V	V
S	$S^* = VP^*(NP^*)$
NP	N^*
VP	$V^*(NP^*)$

Ý nghĩa của câu



- λ form ứng với VP gắn với λ form đi với NP
- Từ là các giá trị
- Cho cây cú pháp, phân tích từ dưới lên để có ngữ nghĩa của câu *ate(John, ice-cream)*
- Vị từ này có thể được đánh giá dựa trên CSDL để trả về 1 giá trị hoặc T/F.

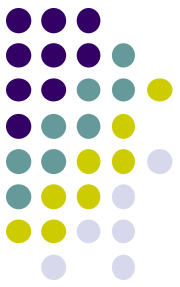
Dịch ngữ nghĩa





Ví dụ

- Phân tích ngữ nghĩa của các câu sau. Chỉ rõ tập luật ngữ nghĩa đã áp dụng (dạng VP[sem=v(obj)] → V[sem=v] NP[sem=obj]).
 - Tâm đã gặp An.
 - Tôi biết Tâm đã gặp An.
 - Tâm đã gặp An ở trường.



Cách thực hiện

- λ ở mức cao nhất gọi đến VP. Giá trị VP này được xác định ở mức lá bằng cách sử dụng tham số NP
- Nói cách khác, để tìm ý nghĩa của câu, ta gọi VP sử dụng tham số là NP
- Tại nút lá, mỗi từ cũng đi kèm thêm một số thông tin ngữ nghĩa

Bài tập



1. Đưa ra tất cả các cách biểu diễn ngữ nghĩa cho các câu sau:
 - Mai likes small cats.
 - Mai likes the cat whose name is Tom.
2. Đưa ra cách biểu diễn ngữ nghĩa dựa trên sự kiện cho các câu sau:
 - Willy wants Lilly to get married.



Mai likes the cat whose name is Tom.

```
(ROOT
  (S
    (NP (NNP Mai))
    (VP (VBZ likes)
      (NP
        (NP (DT the) (NN cat))
        (SBAR
          (WHNP (WP$ whose) (NN name))
          (S
            (VP (VBZ is)
              (NP (NNP Tom)))))))
      (. .)))
```



Willy wants Lilly to get married.

```
(ROOT
  (S
    (NP (NNP Willy))
    (VP (VBZ wants)
      (S
        (NP (NNP Lilly))
        (VP (TO to)
          (VP (VB get)
            (ADJP (JJ married))))))
      (. .)))
```

Bài tập



Với $f(6) = 6 * 6$, thì $f = \lambda x \ x * x$

1. Với $f(\text{John}) = \text{loves}(\text{Mary}, \text{John})$, thì $f = ?$
2. Với $f(\text{John}) = (\forall x \text{ woman}(x) \rightarrow \text{loves}(x, \text{John}))$ thì $f = ?$
3. Với $f(\lambda x \text{ loves}(\text{Mary}, x)) = (\lambda x \text{ Obviously}(\text{loves}(\text{Mary}, x)))$. $f = ?$ Sử dụng nó để biểu diễn ngữ nghĩa câu "Sue obviously loves Mary?"
4. Với $f(\text{Mary})(\text{John}) = (\lambda e \text{ act}(e, \text{loving}), \text{lovee}(e, \text{Mary}), \text{lover}(e, \text{John}))$.
 $f = ?$

Bài tập



5. Cho f giống bài trước. Giả sử

$g(f(\text{Mary})(\text{John})) = (\lambda e \text{ act}(e, \text{loving}), \text{lovee}(e, \text{Mary}), \text{lover}(e, \text{John}), \text{manner}(e, \text{passionate}))$. $g = ?$

Gợi ý: viết $f(\text{Mary})$, nghĩa là “loves Mary”. $g(f(\text{Mary}))$ nghĩa là “passionately loves Mary.”

Ứng dụng của phân tích ngữ nghĩa



(top-level)

Shall I clear the database? (y or n) y

>John saw Mary in the park.

OK.

>Where did John see Mary?

IN THE PARK.

>John gave Fido to Mary.

OK.

>Who gave John Fido?

I DON'T KNOW

>Who gave Mary Fido?

JOHN

>John saw Fido.

OK.

>Who did John see?

FIDO AND MARY