

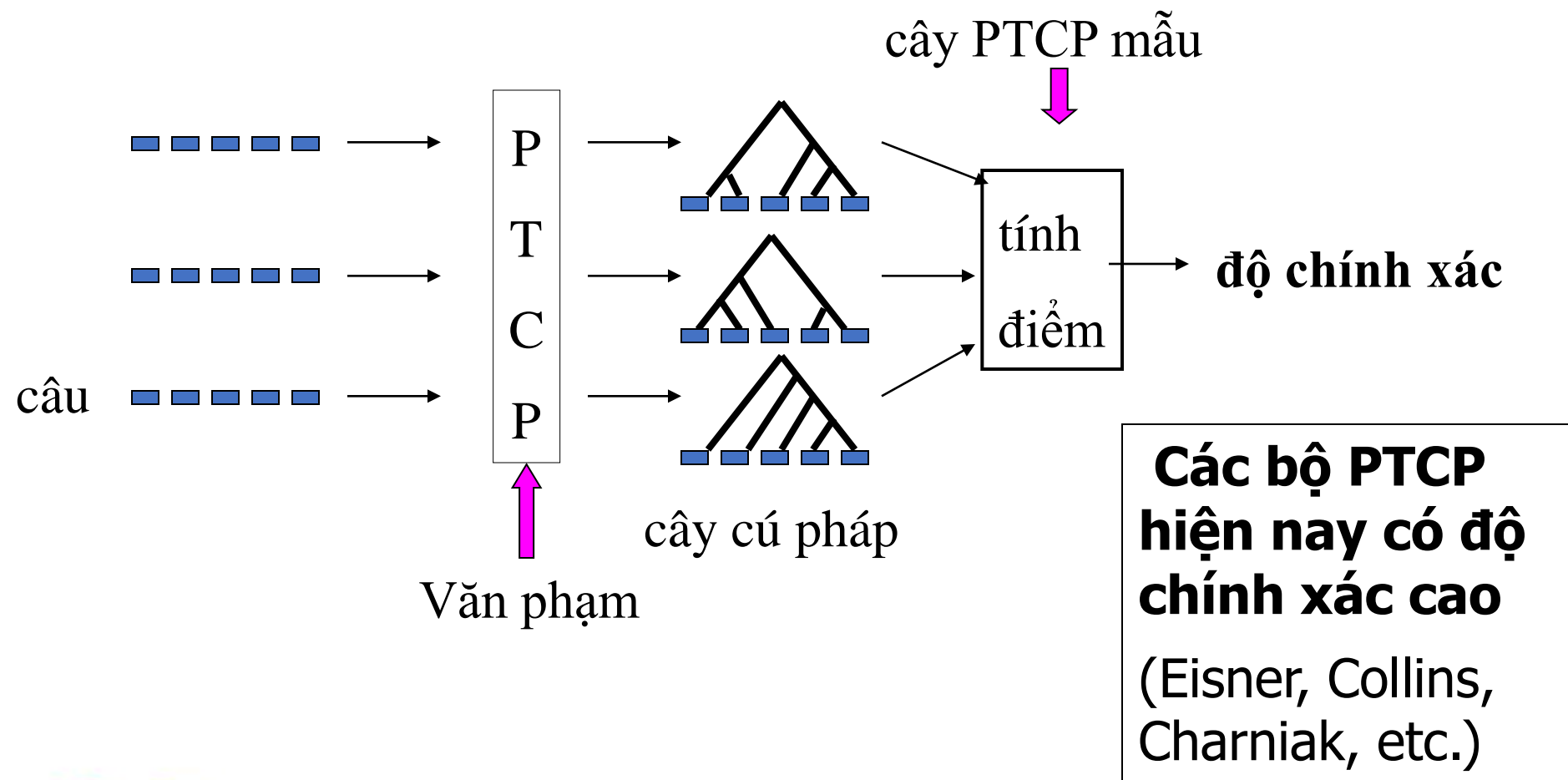


ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Phân tích cú pháp

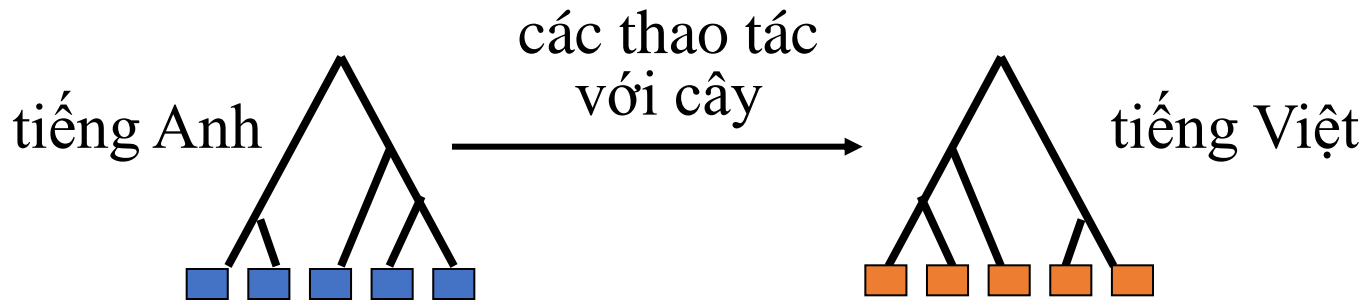
Viện Công nghệ Thông tin và Truyền thông

# Bài toán PTCP



# Các ứng dụng của PTCP

- Dịch máy (Alshawhi 1996, Wu 1997, ...)



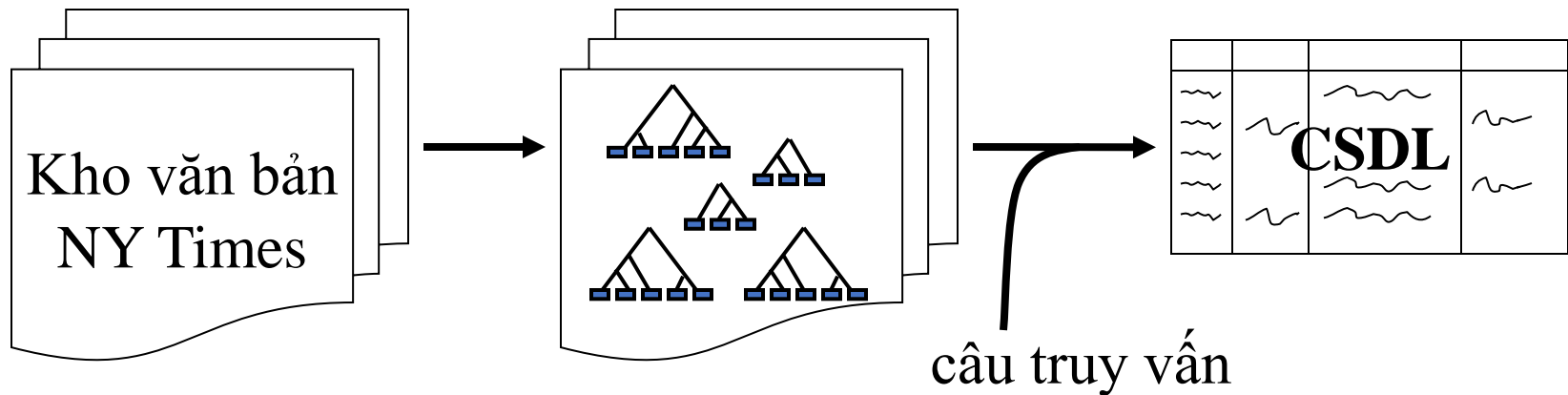
- Nhận dạng tiếng nói sử dụng PTCP (Chelba et al 1998)

Put the file in the folder.

Put the file **and** the folder.

# Các ứng dụng của PTCP

- Kiểm tra ngữ pháp (Microsoft)
- Trích rút thông tin (Hobbs 1996)



# Định nghĩa

- Văn phạm (**grammar**) là dạng biểu diễn hình thức của các cấu trúc được chấp nhận trong 1 ngôn ngữ
- Thuật toán PTCP (**parsing algorithm**) là phương pháp xác định cấu trúc câu trên cơ sở ngữ pháp đã có.
- Chương trình PTCP (**parser**) là chương trình xác định cấu trúc ngữ pháp của câu.

# Ví dụ về văn phạm

- Văn phạm: 1 tập luật viết lại
- Ký hiệu kết thúc: các ký hiệu không thể phân rã được nữa.
- Ký hiệu không kết thúc: các ký hiệu có thể phân rã được.
- Xét văn phạm G:  
     $S \rightarrow NP VP$   
     $NP \rightarrow \text{John, garbage}$   
     $VP \rightarrow \text{laughed, walks}$

G có thể sinh ra các câu sau:

*John laughed. John walks.*

*Garbage laughed. Garbage walks.*

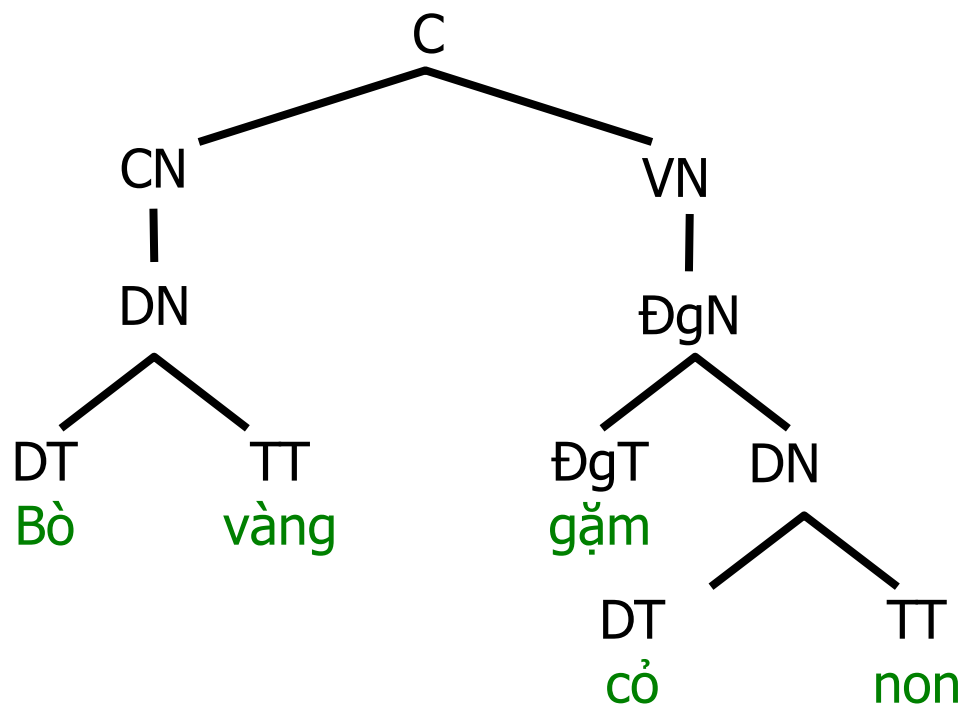
# Ví dụ về văn phạm

Phân tích câu “Bò vàng gặm cỏ non”

- Cây cú pháp:

- Tập luật

- $C \rightarrow CN\ VN$
- $CN \rightarrow DN$
- $VN \rightarrow \text{ĐgN}$
- $\text{ĐgN} \rightarrow \text{ĐgT}\ DN$
- $DN \rightarrow DT\ TT$



# Văn phạm

- Một văn phạm sản sinh là một hệ thống

$G = ( T, N, S, R )$ , trong đó

- $T$  (terminal) – tập ký hiệu kết thúc
- $N$  (non terminal) – tập ký hiệu không kết thúc
- $S$  (start) – ký hiệu khởi đầu
- $R$  (rule) – tập luật
- $R = \{ \alpha \rightarrow \beta \mid \alpha, \beta \in (T \cup N)^* \}$   
 $\alpha \rightarrow \beta$  gọi là luật sản xuất



# Ví dụ

- $G1 = (\{a,b\}, \{X\}, X, \{X \rightarrow \varepsilon, X \rightarrow aXb\})$

Xác định  $L(G1)$

- $G2 = (\{a,b\}, \{X\}, X, \{X \rightarrow \varepsilon, X \rightarrow aXb, X \rightarrow XX\})$

Xác định  $L(G2)$

# Dạng chuẩn Chomsky

- Mọi NNPNK không chứa  $\varepsilon$  đều có thể sinh từ một văn phạm trong đó mọi sản xuất đều có dạng  $A \rightarrow BC$  hoặc  $A \rightarrow a$ , với  $A, B, C \in N$  và  $a \in T$
- Ví dụ: Tìm dạng chuẩn Chomsky cho văn phạm  $G$  với  $T = \{a, b\}$ ,  $N = \{S, A, B\}$ ,  $R$  như sau:
  - $S \rightarrow bA|aB$
  - $A \rightarrow bAA|aS|a$
  - $B \rightarrow aBB|bS|b$

# Văn phạm phi ngữ cảnh (Context-Free Grammar)

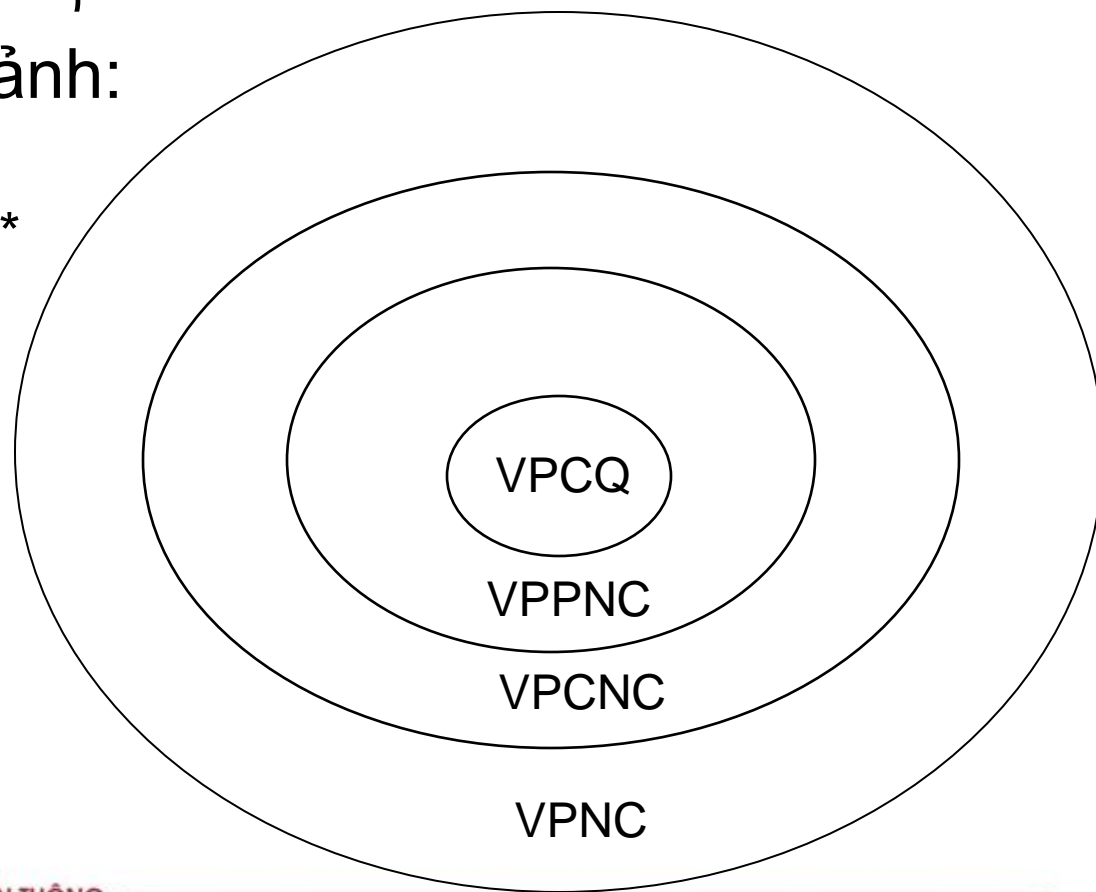
... còn gọi là văn phạm cấu trúc đoạn

- $G = \langle T, N, P, S, R \rangle$ 
  - $T$  – tập các ký hiệu kết thúc (terminals)
  - $N$  – tập các ký hiệu không kết thúc (non-terminals)
  - $P$  – ký hiệu tiền kết thúc (preterminals), khi viết lại trở thành ký hiệu kết thúc,  $P \subset N$
  - $S$  – ký hiệu bắt đầu
  - $R: X \rightarrow \gamma$ ,  $X$  là ký hiệu không kết thúc;  $\gamma$  là chuỗi các ký hiệu kết thúc và không kết thúc (có thể rỗng)
  - Văn phạm  $G$  sinh ra ngôn ngữ  $L$
- Bộ nhận dạng: trả về yes hoặc no
- Bộ PTCP: trả về tập các cây cú pháp

So với văn phạm cảm ngữ cảnh

$R: \alpha A \gamma \Rightarrow \alpha \beta \gamma$   
là ký hiệu không kết thúc;  $\gamma$  là chuỗi các ký hiệu kết thúc và không kết thúc (có thể rỗng)

- Văn phạm ngữ cấu:
  - $\alpha \rightarrow \beta$ , với  $\alpha \in V^+$ ,  $\beta \in V^*$
- Văn phạm cảm ngữ cảnh:
  - $r = \alpha \rightarrow \beta$ , với  $\alpha \in V^+$ ,  $\beta \in V^*$ ,  $|\alpha| \leq |\beta|$
  - và  $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta' \alpha_2$  với  $\beta' \neq \varepsilon$
- Văn phạm phi ngữ cảnh:
  - $A \rightarrow \theta$ ,  $A \in N$ ,
  - với  $\theta \in V^* = (T \cup N)^*$
- Văn phạm chính qui:
  - $A \rightarrow aB$ ,
  - $A \rightarrow Ba$ ,
  - $A \rightarrow a$ ,
  - với  $A, B \in N$ ,  $a \in T$ .



# Văn phạm phi ngữ cảnh

S → NP VP

NP →  $\left\{ \begin{array}{l} \text{DT NNS} \\ \text{DT NN} \\ \text{NP PP} \end{array} \right\}$

VP →  $\left\{ \begin{array}{l} \text{VP PP} \\ \text{VBD} \\ \text{VBD NP} \end{array} \right\}$

PP → IN NP

DT → *the*

NNS →  $\left\{ \begin{array}{l} \textit{children} \\ \textit{students} \\ \textit{mountains} \end{array} \right\}$

VBD →  $\left\{ \begin{array}{l} \textit{slept} \\ \textit{ate} \\ \textit{saw} \end{array} \right\}$

IN →  $\left\{ \begin{array}{l} \textit{in} \\ \textit{of} \end{array} \right\}$

NN → *cake*

# Áp dụng tập luật ngữ pháp

- S

- NP VP

- DT NNS VBD

- *The children slept*

- S

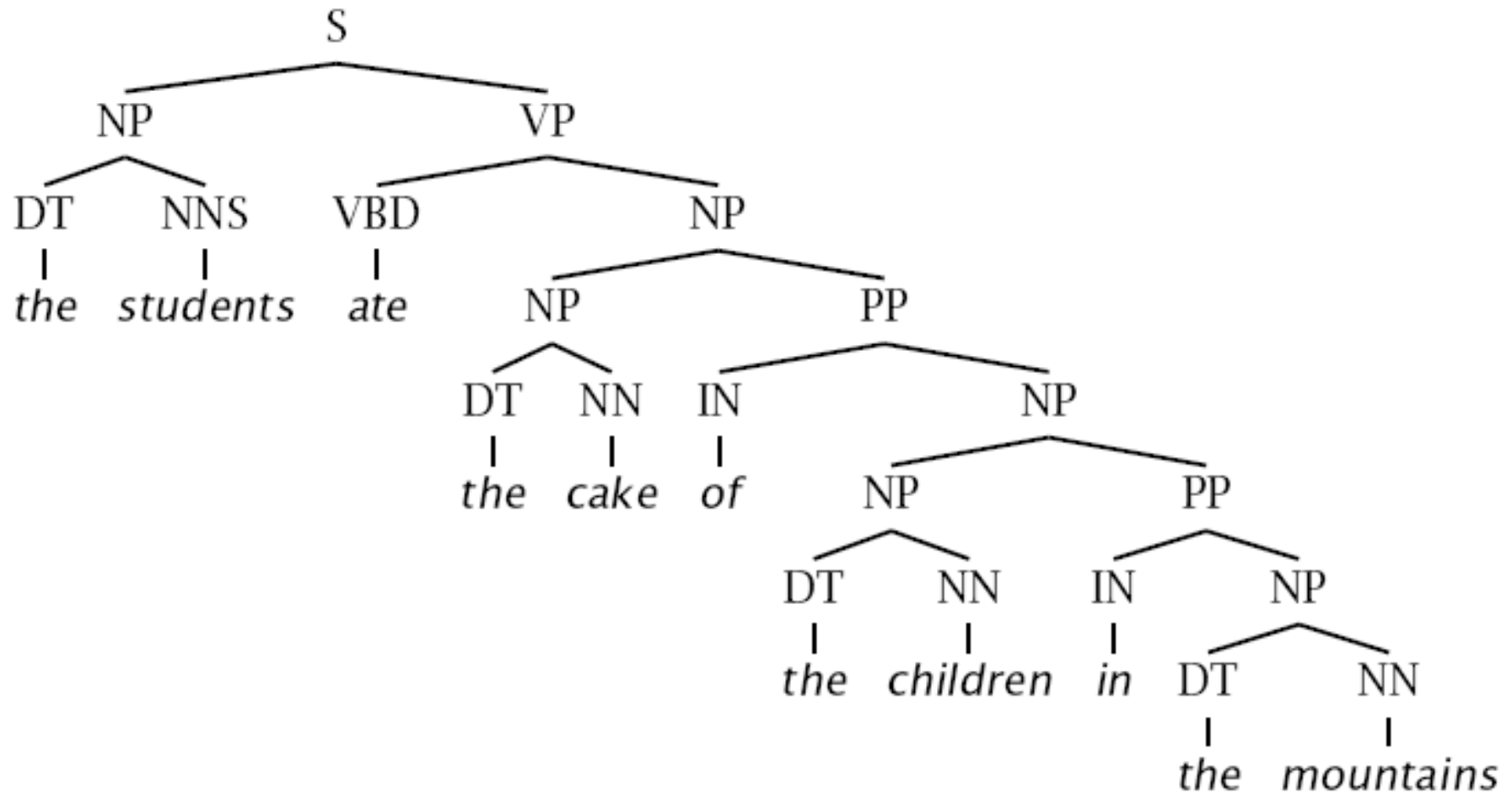
- NP VP

- DT NNS VBD NP

- DT NNS VBD DT NN

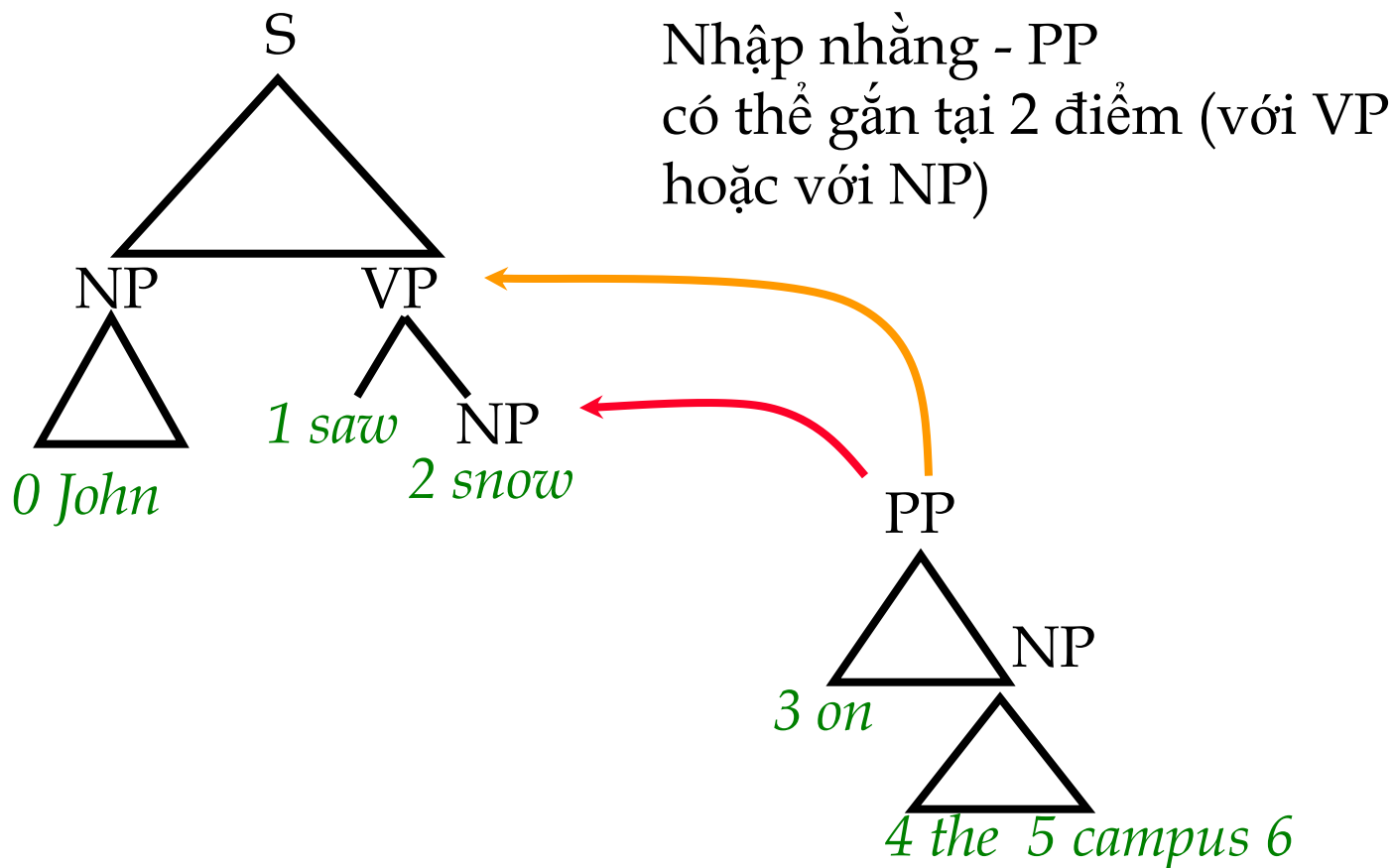
- *The children ate the cake*

# Cấu trúc đoạn đệ qui



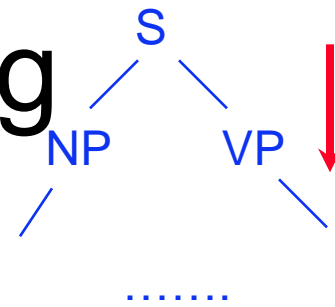
# Văn phạm cho ngôn ngữ tự nhiên có nhập nhằng

## John saw snow on the campus





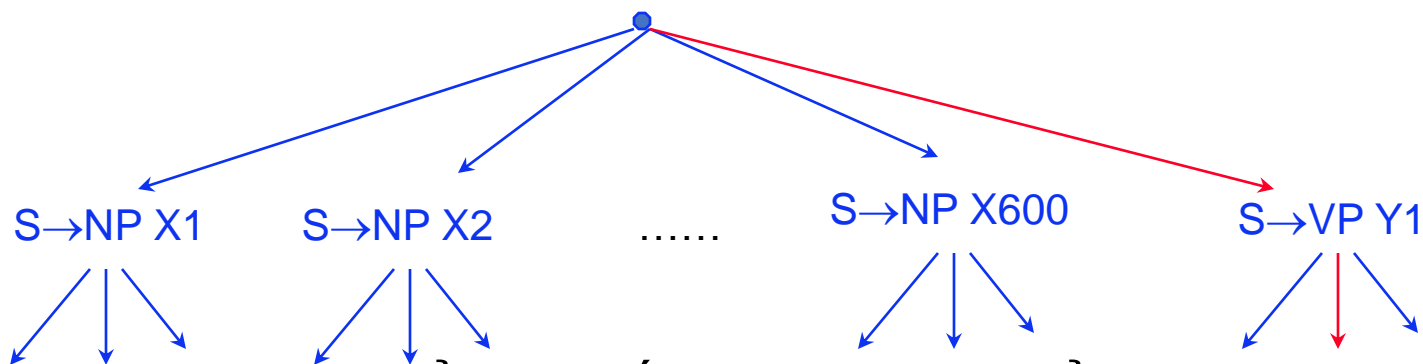
# PTCP kiểu trên xuống



- Hướng đích
- Khởi đầu với 1 danh sách các ký hiệu cần triển khai (S, NP, VP, ...)
- Viết lại các đích trong tập đích bằng cách:
  - tìm luật có vế trái trùng với đích cần triển khai
  - triển khai nó với vế phải luật, tìm cách khớp với câu đầu vào
- Nếu 1 đích có nhiều cách viết lại → chọn 1 luật để áp dụng (bài toán tìm kiếm)
- Có thể sử dụng tìm kiếm rộng (breadth-first search) hoặc tìm kiếm sâu (depth-first search)

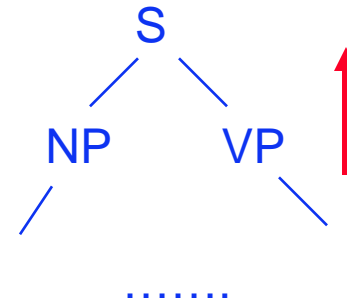
# Khó khăn với PTCP trên xuống

- Các luật đệ quy trái
- PTCP trên xuống rất bất lợi khi có nhiều luật có cùng vế trái



- Nhiều thao tác thừa: triển khai tất cả các nút có thể phân tích trên xuống
- PTCP trên xuống sẽ làm việc tốt khi có chiến lược điều khiển ngữ pháp phù hợp
- PTCP trên xuống không thể triển khai các ký hiệu tiền kết thúc thành các ký hiệu kết thúc. Trên thực tế, người ta thường sử dụng phương pháp dưới lên để làm việc này.
- Lặp lại công việc: bất cứ chỗ nào có cấu trúc giống nhau

# PTCP dưới lên



- Hướng dữ liệu
- Khởi tạo với xâu cần phân tích
- Nếu chuỗi trong tập đích phù hợp với vế phải của 1 luật → thay nó bằng vế trái của luật.
- Kết thúc khi tập đích = {S}.
- Nếu vế phải của các luật khớp với nhiều luật trong tập đích, cần lựa chọn luật áp dụng (bài toán tìm kiếm)
- Có thể sử dụng tìm kiếm rộng (breadth-first search) hoặc tìm kiếm sâu (depth-first search)

# Khó khăn với PTCP dưới lên

- Không hiệu quả khi có nhiều nhập nhằng mức từ vựng
- Lặp lại công việc: bất cứ khi nào có cấu trúc con chung
- Cả PTCP TD (LL) và BU (LR) đều có độ phức tạp là hàm mũ của độ dài câu.

# Thuật toán CKY (bộ nhận dạng)

- **Vào:** xâu  $n$  từ
- **Ra:** yes/no
- **Cấu trúc ngữ pháp:** bảng  $n \times n$  (chart table)
  - hàng đánh số 0 đến  $n-1$
  - cột đánh số 1 đến  $n$
  - cell  $[i,j]$  liệt kê tất cả các nhãn cú pháp giữa  $i$  và  $j$

# Thuật toán CKY (bottom-up)

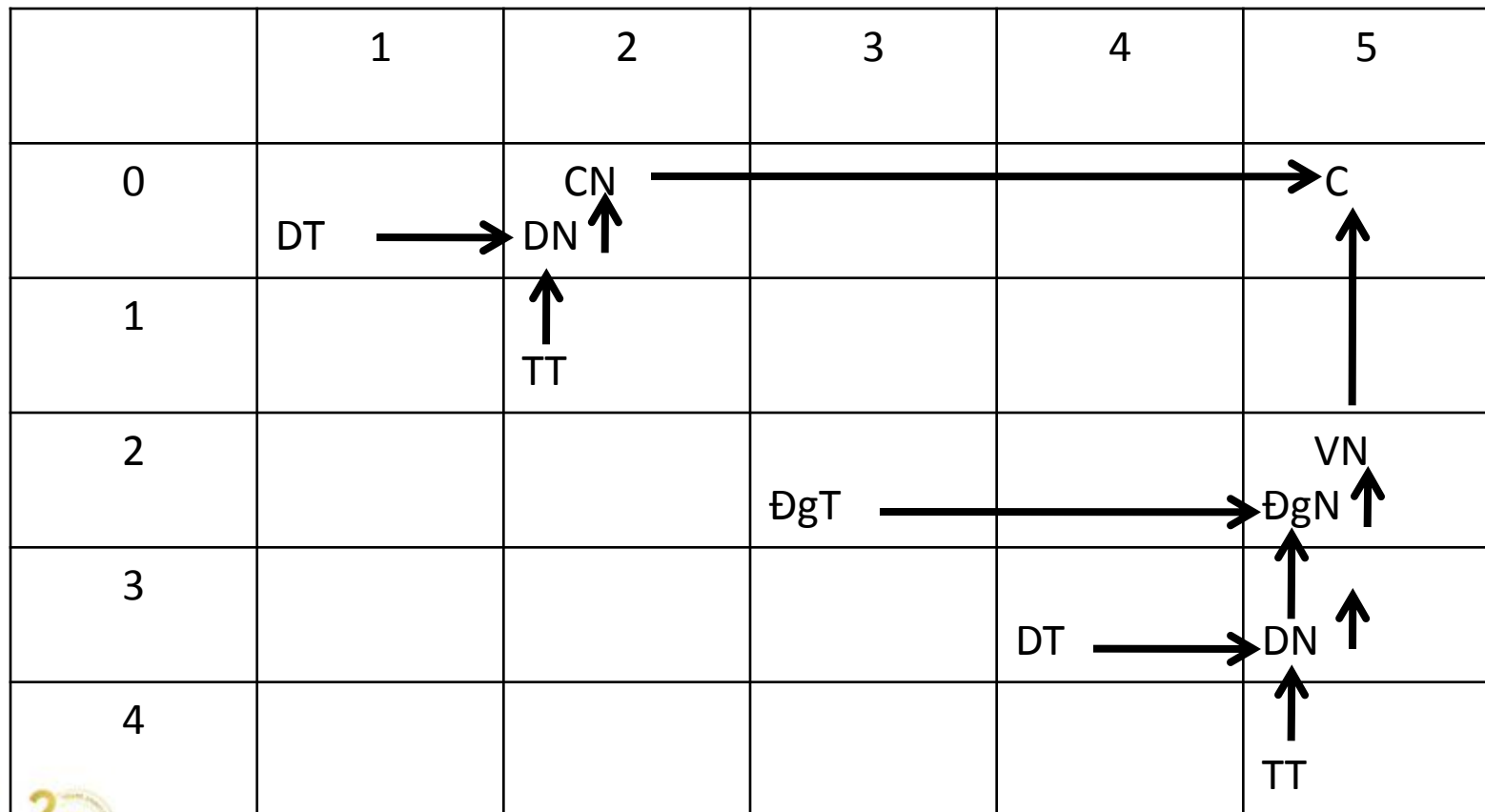
- **for**  $i := 1$  to  $n$ 
  - Thêm tất cả từ loại của từ thứ  $i$  vào ô  $[i-1, i]$
- **for**  $\text{width} := 2$  to  $n$ 
  - **for**  $\text{start} := 0$  to  $n - \text{width}$ 
    - $\text{end} := \text{start} + \text{width}$
    - **for**  $\text{mid} := \text{start} + 1$  to  $\text{end} - 1$ 
      - **for** mọi nhãn cú pháp  $X$  trong  $[\text{start}, \text{mid}]$
      - **for** mọi nhãn cú pháp  $Y$  trong  $[\text{mid}, \text{end}]$
      - **for** mọi cách kết hợp  $X$  và  $Y$  (nếu có)
      - Thêm nhãn kết quả vào  $[\text{start}, \text{end}]$  nếu chưa có nhãn này

# Ví dụ

$C \rightarrow CN \text{ VN}$   
 $CN \rightarrow DN$   
 $VN \rightarrow ĐgN$

$ĐgN \rightarrow ĐgT \text{ DN}$   
 $DN \rightarrow DT \text{ TT}$

## Bò vàng găm cỏ non



# Văn phạm phi ngữ cảnh

1. Start  $\rightarrow$  S
2. S  $\rightarrow$  NP VP
3. NP  $\rightarrow$  Det Noun
4. NP  $\rightarrow$  Name
5. NP  $\rightarrow$  NN PP
6. PP  $\rightarrow$  Prep NP
7. VP  $\rightarrow$  V NP
8. VP  $\rightarrow$  V NP PP
9. V  $\rightarrow$  *ate*
10. Name  $\rightarrow$  *John*
11. Name  $\rightarrow$  *ice-cream, snow*
12. Noun  $\rightarrow$  *ice-cream, pizza*
13. Noun  $\rightarrow$  *table, guy, campus*
14. Det  $\rightarrow$  *the*
15. Prep  $\rightarrow$  *on*



# Luật kết hợp

- Ô  $\text{Cell}[i,j]$  chứa nhãn  $X$  nếu
  - Có luật  $X \rightarrow YZ$ ;
  - $\text{Cell}[i,k]$  chứa nhãn  $Y$  và ô  $\text{Cell}[k,j]$  chứa nhãn  $Z$ , với  $k$  nằm giữa  $i$  và  $j$ ;
- VD:  $\text{NP} \rightarrow \text{DT}[0,1] \text{NN}[1,2]$

# CKY phải sử dụng luật nhị phân

- Chuyển  $VP \rightarrow V \text{ NP PP}$  thành:
  - 8.a.  $VP \rightarrow V \text{ Arguments}$
  - 8.b.  $\text{Arguments} \rightarrow \text{NP PP}$

1. Start  $\rightarrow$  S
2. S  $\rightarrow$  NP VP
3. NP  $\rightarrow$  DT NN
4. NP  $\rightarrow$  Name
5. NP  $\rightarrow$  NN PP
6. PP  $\rightarrow$  Prep NP
7. VP  $\rightarrow$  V NP
- 8.a. VP  $\rightarrow$  V Arguments
- 8.b. Arguments  $\rightarrow$  NP PP

“The guy ate the ice-cream on the table”

	1	2	3	4	5	6	7	8
0	DT							
1		NN						
2			VBD					
3				DT				
4					NN			
5						Prep		
6							DT	
7								NN

# Áp dụng thao tác ‘dán’

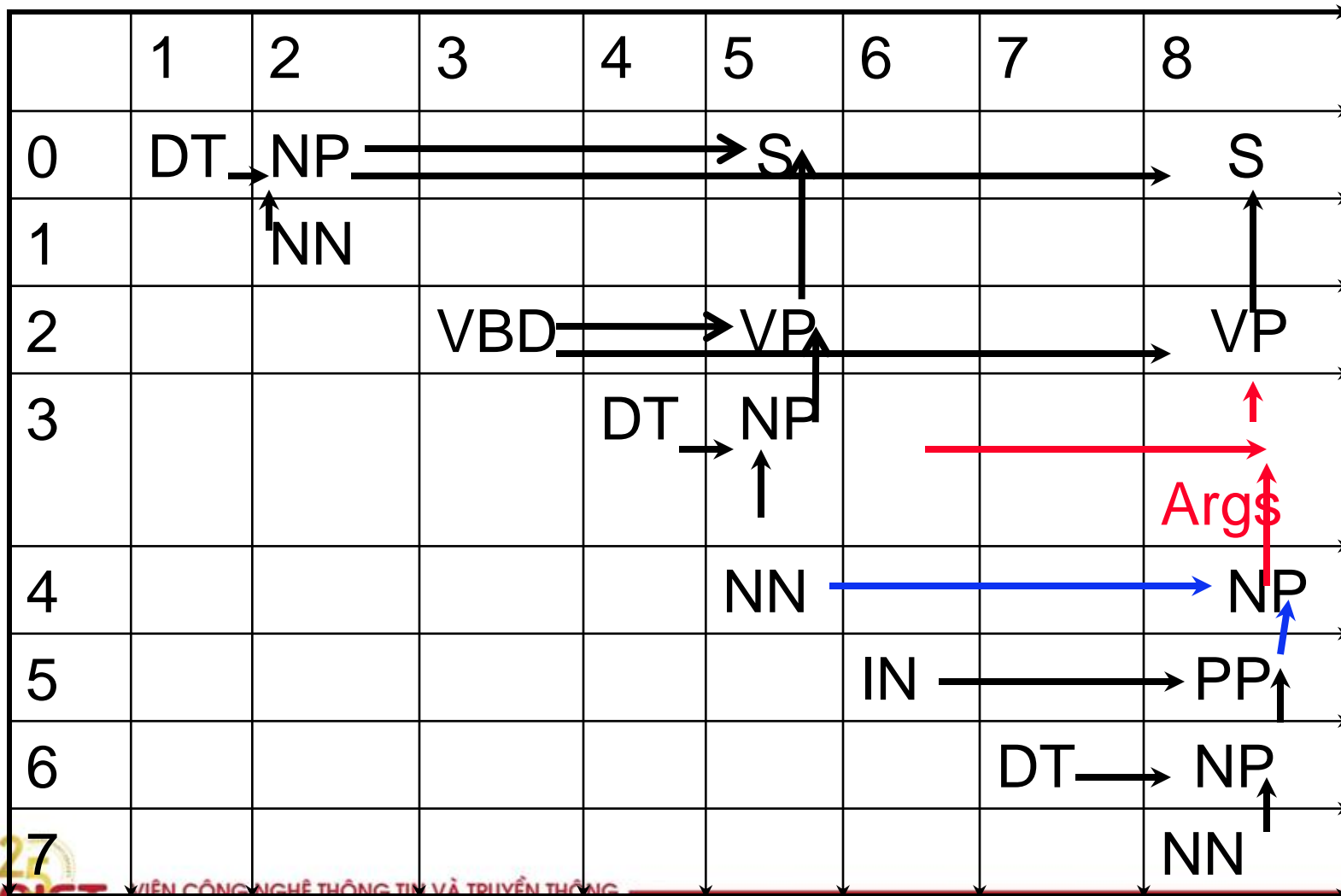
	1	2	3	4	5	6	7	8
0	DT→	NP						
1		NN						
2			VBD					
3				DT				
4					NN			
5						IN		
6							DT	
7								NN

# Nhập nhằng!

5. NP → NN PP

8.a. VP → V Arguments

8.b. Arguments → NP PP

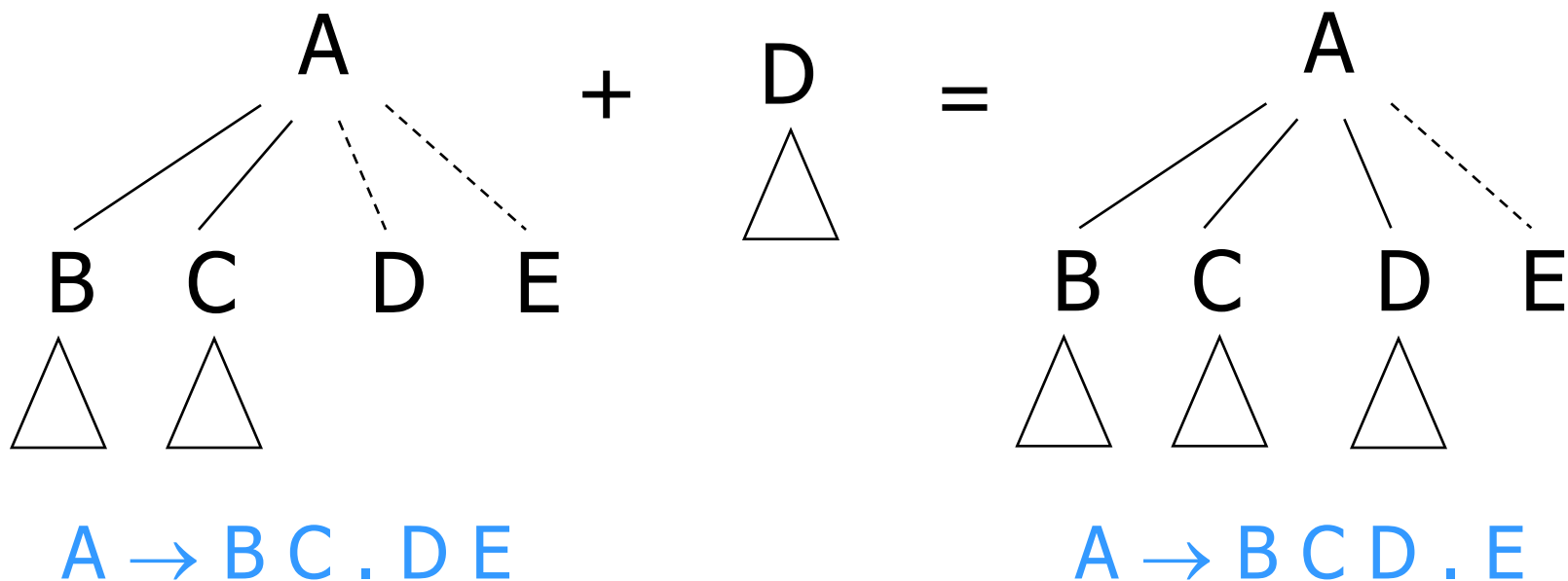


- $C \rightarrow DN \text{ ĐgN}$
  - $DN \rightarrow LT \text{ DT}$
  - $DN \rightarrow DN \text{ TT}$
  - $\text{ĐgN} \rightarrow \text{ĐgT DN}$
- 
- Phân tích câu sau sử dụng CKY
  - Con ngựa đá con ngựa đá
  - LT DT ĐgT LT DT TT

# Thuật toán Earley (top-down)

- Tìm các nhãn và các nhãn thiếu (partial constituents) từ đầu vào

- $A \rightarrow B C . D E$  là nhãn thiếu:



- Tiến hành dần từ trái sang phải

# Ví dụ

ROOT  $\rightarrow$  S

S  $\rightarrow$  NP VP

NP  $\rightarrow$  Det N

NP  $\rightarrow$  NP PP

VP  $\rightarrow$  VP PP

VP  $\rightarrow$  V NP

PP  $\rightarrow$  P NP

NP  $\rightarrow$  Papa

N  $\rightarrow$  caviar

N  $\rightarrow$  spoon

V  $\rightarrow$  ate

P  $\rightarrow$  with

Det  $\rightarrow$  the

Det  $\rightarrow$  a



# Recursive Descent (Đệ quy)

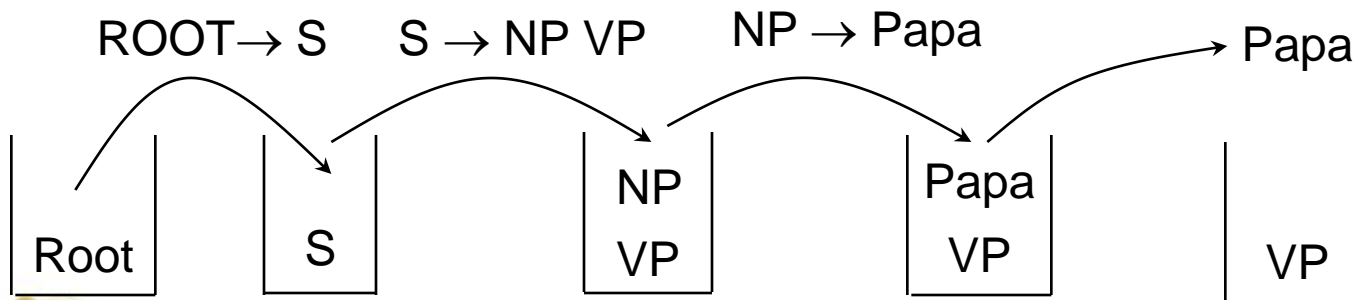
ROOT  $\rightarrow$  S  
S  $\rightarrow$  NP VP  
NP  $\rightarrow$  Det N  
NP  $\rightarrow$  NP PP

VP  $\rightarrow$  VP PP  
VP  $\rightarrow$  V NP  
PP  $\rightarrow$  P NP

NP  $\rightarrow$  Papa  
N  $\rightarrow$  caviar  
N  $\rightarrow$  spoon

V  $\rightarrow$  ate  
P  $\rightarrow$  with  
Det  $\rightarrow$  the  
Det  $\rightarrow$  a

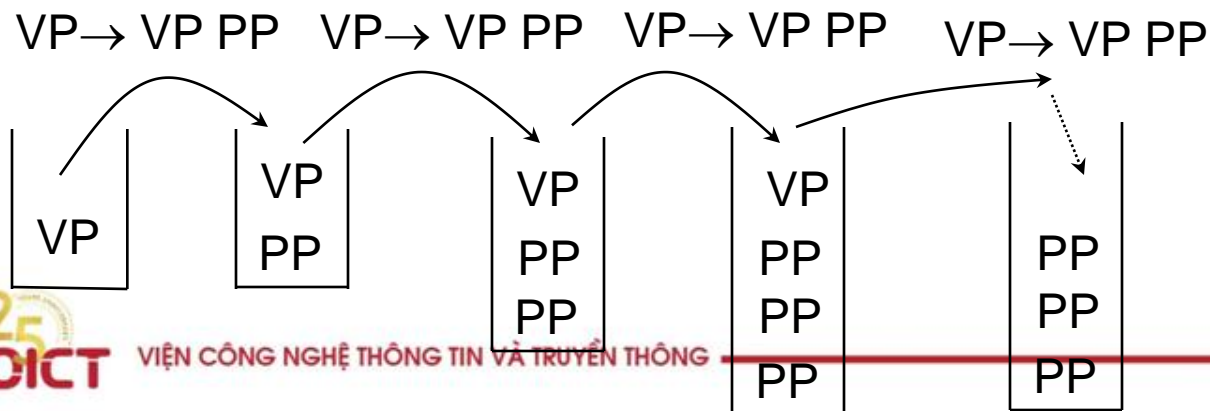
- 0 ROOT  $\rightarrow$  . S 0
  - 0 S  $\rightarrow$  . NP VP 0
    - 0 NP  $\rightarrow$  . Papa 0
    - 0 NP  $\rightarrow$  Papa . 1
  - 0 S  $\rightarrow$  NP . VP 1



# Recursive Descent

ROOT $\rightarrow$ S	VP $\rightarrow$ VP PP	NP $\rightarrow$ Papa	V $\rightarrow$ ate
S $\rightarrow$ NP VP	VP $\rightarrow$ V NP	N $\rightarrow$ caviar	P $\rightarrow$ with
NP $\rightarrow$ Det N	PP $\rightarrow$ P NP	N $\rightarrow$ spoon	Det $\rightarrow$ the
NP $\rightarrow$ NP PP			Det $\rightarrow$ a

- 0 S  $\rightarrow$  NP . VP 1
  - 1 VP  $\rightarrow$  . VP PP 1
    - 1 VP  $\rightarrow$  . VP PP 1
      - 1 VP  $\rightarrow$  . VP PP 1
        - 1 VP  $\rightarrow$  . VP PP 1 stack overflowed



# Recursive Descent

ROOT  $\rightarrow$  S  
 S  $\rightarrow$  NP VP  
 NP  $\rightarrow$  Det N  
 NP  $\rightarrow$  NP PP

VP  $\rightarrow$  V NP  
 VP  $\rightarrow$  VP PP  
 PP  $\rightarrow$  P NP

NP  $\rightarrow$  Papa  
 N  $\rightarrow$  caviar  
 N  $\rightarrow$  spoon  
 V  $\rightarrow$  ate  
 P  $\rightarrow$  with  
 Det  $\rightarrow$  the  
 Det  $\rightarrow$  a

0 ROOT  $\rightarrow$  . S 0

0 S  $\rightarrow$  . NP VP 0

0 NP  $\rightarrow$  . Papa 0

0 NP  $\rightarrow$  Papa . 1

0 S  $\rightarrow$  NP . VP 1

1 VP  $\rightarrow$  . V NP 1

1 V  $\rightarrow$  . ate 1

1 V  $\rightarrow$  ate . 2

1 VP  $\rightarrow$  V . NP 2

2 NP  $\rightarrow$  . ... 2

2 NP  $\rightarrow$  ... . 7

1 VP  $\rightarrow$  V NP . 7

0 S  $\rightarrow$  NP VP . 7

sau . = nonterminal, lặp đi lặp lại việc tìm ký hiệu này ("**predict**")

sau . = terminal, tìm nó ở đầu vào ("**scan**")

sau . = rỗng, đích con của cha nó đã hoàn chỉnh ("**attach**")

**predict** (đích con tiếp theo)

phân tích tiếp và cuối cùng ...

hoàn thành đích con NP của cha nó  $\rightarrow$  **attach**

**attach**

**attach**

# Recursive Descent

ROOT  $\rightarrow$  S  
S  $\rightarrow$  NP VP  
NP  $\rightarrow$  Det N  
NP  $\rightarrow$  NP PP

VP  $\rightarrow$  V NP  
VP  $\rightarrow$  VP PP  
PP  $\rightarrow$  P NP

NP  $\rightarrow$  Papa  
N  $\rightarrow$  caviar  
N  $\rightarrow$  spoon

V  $\rightarrow$  ate  
P  $\rightarrow$  with  
Det  $\rightarrow$  the  
Det  $\rightarrow$  a

- 0 ROOT  $\rightarrow$  . S 0
  - 0 S  $\rightarrow$  . NP VP 0
    - 0 NP  $\rightarrow$  . Papa 0
    - 0 NP  $\rightarrow$  Papa . 1
  - 0 S  $\rightarrow$  NP . VP 1
    - 1 VP  $\rightarrow$  . V NP 1
      - 1 V  $\rightarrow$  . ate 1
      - 1 V  $\rightarrow$  ate . 2
      - 1 VP  $\rightarrow$  V . NP 2
      - 2 NP  $\rightarrow$  . ... 2
      - 2 NP  $\rightarrow$  ... . 7
    - 1 VP  $\rightarrow$  V NP . 7
  - 0 S  $\rightarrow$  NP VP . 7

thực hiện bằng lời gọi hàm:

S() gọi NP() và VP(), VP được triển khai 1 cách đệ qui

cần quay lại để thử 1 luật VP khác

# Recursive Descent

ROOT $\rightarrow$ S	VP $\rightarrow$ V NP	NP $\rightarrow$ Papa	V $\rightarrow$ ate
S $\rightarrow$ NP VP	VP $\rightarrow$ VP PP	N $\rightarrow$ caviar	P $\rightarrow$ with
NP $\rightarrow$ Det N	PP $\rightarrow$ P NP	N $\rightarrow$ spoon	Det $\rightarrow$ the
NP $\rightarrow$ NP PP			Det $\rightarrow$ a

0 ROOT  $\rightarrow$  . S 0

0 S  $\rightarrow$  . NP VP 0

0 NP  $\rightarrow$  . Papa 0

0 NP  $\rightarrow$  Papa . 1

0 S  $\rightarrow$  NP . VP 1

1 VP  $\rightarrow$  . VP PP 1

• 1 V  $\rightarrow$  . ate 1

• 1 V  $\rightarrow$  ate . 2

• 1 VP  $\rightarrow$  V . NP 2

• 2 NP  $\rightarrow$  . ... 2 phân tích tiếp và cuối cùng...

• 2 NP  $\rightarrow$  ... . 4 ... đoạn NP đúng là từ 2 đến 4

chỗ này cũng cần quay lại

# Recursive Descent

0 Papa 1 ate 2 the 3 caviar 4 with 5 a 6 spoon 7

ROOT $\rightarrow$ S	VP $\rightarrow$ V NP	NP $\rightarrow$ Papa	V $\rightarrow$ ate
S $\rightarrow$ NP VP	VP $\rightarrow$ VP PP	N $\rightarrow$ caviar	P $\rightarrow$ with
NP $\rightarrow$ Det N	PP $\rightarrow$ P NP	N $\rightarrow$ spoon	Det $\rightarrow$ the
NP $\rightarrow$ NP PP			Det $\rightarrow$ a

- 0 ROOT  $\rightarrow$  . S 0
  - 0 S  $\rightarrow$  . NP VP 0
    - 0 NP  $\rightarrow$  . Papa 0
    - 0 NP  $\rightarrow$  Papa . 1
  - 0 S  $\rightarrow$  NP . VP 1
    - 1 VP  $\rightarrow$  . VP PP 1
      - 1 VP  $\rightarrow$  . VP PP 1
        - 1 VP  $\rightarrow$  . VP PP 1
        - 1 VP  $\rightarrow$  . VP PP 1
        - 1 VP  $\rightarrow$  . VP PP 1

stack overflowed

không giải quyết được gì

– cần thay đổi tập luật để loại trừ đệ qui trái

# Thuật toán Earley

- Thuật toán Earley giống thuật toán đệ qui nói trên, nhưng giải quyết được vấn đề đệ qui trái.
- Sử dụng bảng phân tích giống thuật toán CKY, nhằm lưu lại các thông tin đã tìm thấy → lập trình động “**Dynamic programming.**”

## Các thao tác của thuật toán

- Xử lý phần đi sau dấu . theo kiểu đệ qui :
  - Nếu là từ, quét (**scan**) đầu vào để xem có phù hợp không
  - Nếu là ký hiệu không kết thúc, đoán (**predict**) các khả năng để khớp nó (giảm số phép tiên đoán bằng cách nhìn trước k ký hiệu từ đầu vào và chỉ sử dụng các luật phù hợp với k ký hiệu đó)
  - Nếu xong, ta đã hoàn thành một thành phần ngữ pháp, gắn (**attach**) nó vào những chỗ liên quan

0
0 ROOT . S

## khởi tạo

*tương đương với*  $(0, \text{ROOT} \rightarrow . S)$



0
0 ROOT . S
0 S . NP VP
 

**predict** luật có vế trái là  $S$

$$(0, S \rightarrow \cdot NP VP)$$

0

0 ROOT . S

0 S . NP VP

0 NP . Det N

0 NP . NP PP

0 NP . Papa

**predict** luật có VT = NP  
(có 3 luật phù hợp)





[illegible]

**scan:** từ phù hợp từ đầu vào

0	Papa	1
0 ROOT . S	0 NP Papa .	
0 S . NP VP		
0 NP . Det N		
0 NP . NP PP		
0 NP . Papa		

0 Det . the

0 Det . a

**scan:** không phù hợp

0	Papa	1
0 ROOT . S		0 NP Papa .
0 S . NP VP		
0 NP . Det N		
0 NP . NP PP		
0 NP . Papa		
0 Det . the		

0 Det. a	<b>scan:</b> không phù hợp
----------	----------------------------

[illegible]

**attach NP mới tạo** (bắt đầu từ 0) với các phần liên quan (các phần chưa hoàn thành kết thúc tại 0 và có NP sau dấu . )



# predict

# predict



0	Papa	1
0 ROOT . S	0 NP Papa .	
0 S . NP VP	0 S NP . VP	
0 NP . Det N	0 NP NP . PP	
0 NP . NP PP	1 VP . V NP	
0 NP . Papa	1 VP . VP PP	
0 Det . the	1 PP . P NP	
0 Det . a	1 V . ate	

predict



0	Papa	1
0 ROOT . S	0 NP Papa .	
0 S . NP VP	0 S NP . VP	
0 NP . Det N	0 NP NP . PP	
0 NP . NP PP	1 VP . V NP	
0 NP . Papa	1 VP . VP PP	
0 Det . the	1 PP . P NP	
0 Det . a	1 V . ate	
	1 P . with	

predict

0	Papa	1	ate	2
0 ROOT . S	0 NP Papa .	1 V ate .		
0 S . NP VP	0 S NP . VP			
0 NP . Det N	0 NP NP . PP			
0 NP . NP PP	1 VP . V NP			
0 NP . Papa	1 VP . VP PP			
0 Det . the	1 PP . P NP	scan: thanh		
0 Det . a	1 V . ate			
	1 P . with			
  <b>SOICT</b>	<b>VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG</b>			

**scan:** thành công!

0	Papa	1	ate	2
0 ROOT . S	0 NP Papa .	1 V ate .		
0 S . NP VP	0 S NP . VP			
0 NP . Det N	0 NP NP . PP			
0 NP . NP PP	1 VP . V NP			
0 NP . Papa	1 VP . VP PP			
0 Det . the	1 PP . P NP			
0 Det . a	1 V . ate			
	1 P . with			

**scan:** không hợp





[illegible]

0	Papa	1	ate	2
0 ROOT . S	0 NP Papa .	1 V ate .		
0 S . NP VP	0 S NP . VP	1 VP V . NP		
0 NP . Det N	0 NP NP . PP	2 NP . Det N		
0 NP . NP PP	1 VP . V NP	2 NP . NP PP		
0 NP . Papa	1 VP . VP PP	2 NP . Papa		
0 Det . the	1 PP . P NP	2 Det . the		
0 Det . a	1 V . ate	2 Det . a		
	1 P . with			

**predict** (các bước sau tương tự)

0	Papa	1	ate	2
0 ROOT . S	0 NP Papa .	1 V ate .		
0 S . NP VP	0 S NP . VP	1 VP V . NP		
0 NP . Det N	0 NP NP . PP	2 NP . Det N		
0 NP . NP PP	1 VP . V NP	2 NP . NP PP		
0 NP . Papa	1 VP . VP PP	2 NP . Papa		
0 Det . the	1 PP . P NP	2 Det . the		
0 Det . a	1 V . ate	2 Det . a		
	1 P . with			

predict

[illegible]

0	Papa	1	ate	2	the	3
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .			
0 S . NP VP	0 S NP . VP	1 VP V . NP				
0 NP . Det N	0 NP NP . PP	2 NP . Det N				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa				
0 Det . the	1 PP . P NP	2 Det . the				
0 Det . a	1 V . ate	2 Det . a				
	1 P . with					

**scan:** thành công!



[illegible]









## attach

## attach

[illegible]

[illegible]

0	Papa	1	ate	2	the	3	caviar	4
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP				
	1 P . with			0 ROOT S .				

attach





0	Papa	1	ate	2	the	3	caviar	4
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP				
	1 P . with			0 ROOT S .				
				4 P . with				

0	Papa	1	ate	2	the	3	caviar	4
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP				
	1 P . with			0 ROOT S .				
				4 P . with				

[illegible]

[illegible]

[illegible]

0	Papa	1	ate	2	the	3	caviar	4	with	5
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	4 P with .					
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP					
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N					
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP					
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .	5 NP . Papa					
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP	5 Det . the					
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP	5 Det . a					
	1 P . with			0 ROOT S .						
				4 P . with						

0	Papa	1	ate	2	the	3	caviar	4	with	5
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	4 P with .					
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP					
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N					
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP					
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .	5 NP . Papa					
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP	5 Det . the					
0 Det a	1 V ate	2 Det a		4 PP P NP	5 Det a					

0Bò1 vàng2 gặm3 cỏ4 non5 trên6 cánh đồng7

C → CN VN

CN → DN

CN → ĐgN

VN → ĐgN

DN → DT TT

DN → DN TT

ĐgN → ĐgT DN

ĐgN → ĐgN GN

GN → GT DT

DT → bò, cỏ, cánh đồng

ĐgT → gặm

TT → vàng, non

GT → trên

GN = giới ngữ

GT = giới từ

[illegible]



[illegible]

Bò vàng gặm cỏ non

$C \rightarrow CN\ VN$

$CN \rightarrow DN$

$CN \rightarrow ĐgN$

$VN \rightarrow ĐgN$

$DN \rightarrow DT\ TT$

$DN \rightarrow DN\ TT$

$ĐgN \rightarrow ĐgT\ DN$

$ĐgN \rightarrow ĐgN\ GN$

$GN \rightarrow GT\ DN$

$DT \rightarrow \text{bò, cỏ}$

$ĐgT \rightarrow \text{gặm}$

$TT \rightarrow \text{vàng, non}$

$GN = \text{giới ngữ}$

$GT = \text{giới từ}$

	ate	2	the	3	caviar	4	with	5	a	6
.	1 V ate .	2 Det the .	3 N caviar .	4 P with .	5 Det a .					
P	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP	5 NP Det . N					
PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N						
P	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP						
PP	2 NP . Papa		0 S NP VP .	5 NP . Papa						
P	2 Det . the		1 VP VP . PP	5 Det . the						
	2 Det . a		4 PP . P NP	5 Det . a						
			0 ROOT S .							
			4 P . with							

	ate	2	the	3	caviar	4	with	5	a	6
.	1 V ate .	2 Det the .	3 N caviar .	4 P with .	5 Det a .					
P	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP	5 NP Det . N					
PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N	6 N . caviar					
P	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP	6 N . spoon					
PP	2 NP . Papa		0 S NP VP .	5 NP . Papa						
P	2 Det . the		1 VP VP . PP	5 Det . the						
	2 Det . a		4 PP . P NP	5 Det . a						
			0 ROOT S .							
			4 P . with							

	ate	2	the	3	caviar	4	with	5	a	6
.	1 V ate .	2 Det the .	3 N caviar .	4 P with .	5 Det a .					
P	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP	5 NP Det . N					
PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N	6 N . caviar					
P	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP	6 N . spoon					
PP	2 NP . Papa		0 S NP VP .	5 NP . Papa						
P	2 Det . the		1 VP VP . PP	5 Det . the						
	2 Det . a		4 PP . P NP	5 Det . a						
			0 ROOT S .							
			4 P . with							

ate      2      the      3      caviar      4      with      5      a      6      spoon      7

.	1 V ate .	2 Det the .	3 N caviar .	4 P with .	5 Det a .	6 N spoon .
P	1 VP V . NP	2 NP Det . N	2 NP Det N .	4 PP P . NP	5 NP Det . N	
PP	2 NP . Det N	3 N . caviar	1 VP V NP .	5 NP . Det N	6 N . caviar	
P	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP . NP PP	6 N . spoon	
PP	2 NP . Papa		0 S NP VP .	5 NP . Papa		
P	2 Det . the		1 VP VP . PP	5 Det . the		
	2 Det . a		4 PP . P NP	5 Det . a		
			0 ROOT S .			
			4 P . with			

[illegible]






[illegible]

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP				
	1 P . with			0 ROOT S .						
				4 P . with						
	</									

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP				
	1 P . with			0 ROOT S .		1 VP V NP .				
				4 P . with		2 NP NP . PP				

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP				
	1 P . with			0 ROOT S .		1 VP V NP .				
				4 P . with		2 NP NP . PP				
						0 S NP VP .				
						1 VP VP . PP				



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG


92

0      Papa      1      ate      2      the      3      caviar      4      with a spoon      7					
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	... 6 N spoon .
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .	5 NP Det N .
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .	4 PP P NP .
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP NP . PP
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .	2 NP NP PP .
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP	1 VP VP PP .
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP	7 PP . P NP
	1 P . with			0 ROOT S .	1 VP V NP .
				4 P . with	2 NP NP . PP
					0 S NP VP .
					1 VP VP . PP
					7 P . with

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .	5 NP Det N .					
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .	4 PP P NP .					
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP NP . PP					
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .	2 NP NP PP .					
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP	1 VP VP PP .					
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP	7 PP . P NP					
	1 P . with			0 ROOT S .	1 VP V NP .					
				4 P . with	2 NP NP . PP					
					0 S NP VP .					
					1 VP VP . PP					
					7 P . with					


94

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .	5 NP Det N .					
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .	4 PP P NP .					
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP	5 NP NP . PP					
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .	2 NP NP PP .					
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP	1 VP VP PP .					
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP	7 PP . P NP					
	1 P . with			0 ROOT S .	1 VP V NP .					
				4 P . with	2 NP NP . PP					
					0 S NP VP .					
					1 VP VP . PP					
					7 P . with					





VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

95

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP				
	1 P . with			0 ROOT S .		1 VP V NP .				
				4 P . with		2 NP NP . PP				
						0 S NP VP .				
						1 VP VP . PP				
						7 P . with				
						0 ROOT S .				
 VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG										
						96				



0      Papa      1      ate      2      the      3      caviar      4      with a spoon      7						
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP
	1 P . with			0 ROOT S .		1 VP V NP .
				4 P . with		2 NP NP . PP
						0 S NP VP .
						1 VP VP . PP
						7 P . with
						0 ROOT S .
 VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG						
						97

0      Papa      1      ate      2      the      3      caviar      4      with a spoon      7						
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP
	1 P . with			0 ROOT S .		1 VP V NP .
				4 P . with		2 NP NP . PP
						0 S NP VP .
						1 VP VP . PP
						7 P . with
						0 ROOT S .
 VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG						
						98

0 Papa 1 ate 2 the 3 caviar 4 with a spoon 7						
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP
	1 P . with			0 ROOT S .		1 VP V NP .
				4 P . with		2 NP NP . PP
						0 S NP VP .
						1 VP VP . PP
						7 P . with
						0 ROOT S .

0	Papa	1	ate	2	the	3	caviar	4	with a spoon	7
0 ROOT . S	0 NP Papa .	1 V ate .	2 Det the .	3 N caviar .	...	6 N spoon .				
0 S . NP VP	0 S NP . VP	1 VP V . NP	2 NP Det . N	2 NP Det N .		5 NP Det N .				
0 NP . Det N	0 NP NP . PP	2 NP . Det N	3 N . caviar	1 VP V NP .		4 PP P NP .				
0 NP . NP PP	1 VP . V NP	2 NP . NP PP	3 N . spoon	2 NP NP . PP		5 NP NP . PP				
0 NP . Papa	1 VP . VP PP	2 NP . Papa		0 S NP VP .		2 NP NP PP .				
0 Det . the	1 PP . P NP	2 Det . the		1 VP VP . PP		1 VP VP PP .				
0 Det . a	1 V . ate	2 Det . a		4 PP . P NP		7 PP . P NP				
	1 P . with			0 ROOT S .		1 VP V NP .				
				4 P . with		2 NP NP . PP				
						0 S NP VP .				
						1 VP VP . PP				
						7 P . with				
						0 ROOT S .				

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

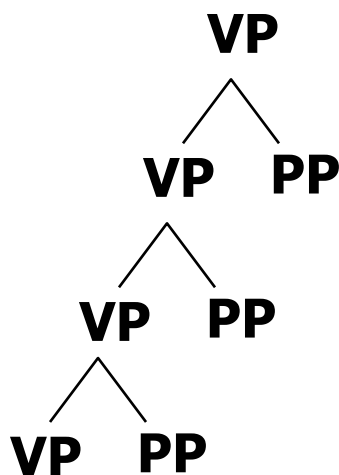
100

0 Ông\_già 1 đi 2 nhanh 3 quá 4  
DT ĐgT TT PT

- C → CN VN
- CN → DT
- CN → DN
- VN → ĐgN
- VN → ĐgT
- DN → DT TT
- ĐgN → ĐgT TT
- ĐgN → ĐgT TN
- TN → TT PT

- DT → ông\_già, ngựa
- ĐgT → đi, đá
- TT → nhanh, chậm
- PT → quá, thể

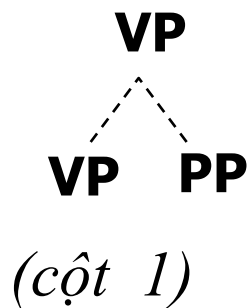
# Vấn đề với PTCP trên xuống: đệ quy trái



gắn liên tục các luật mới  
vào cây trước khi thấy  
PPs

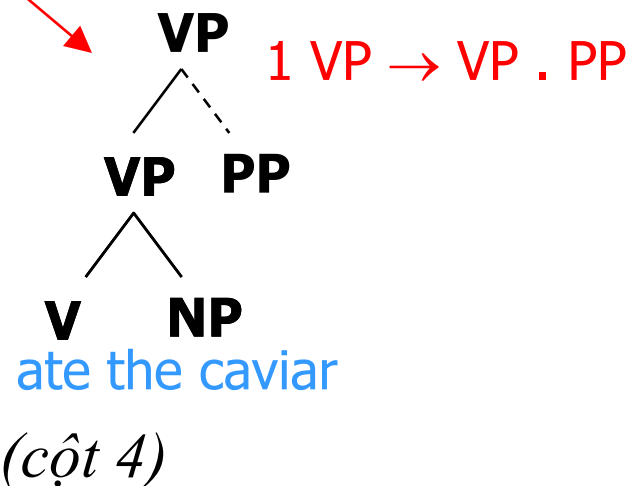
→ cần đoán trước số PP  
cần ở đâu vào

# ... nhưng thuật toán Earley Ok!



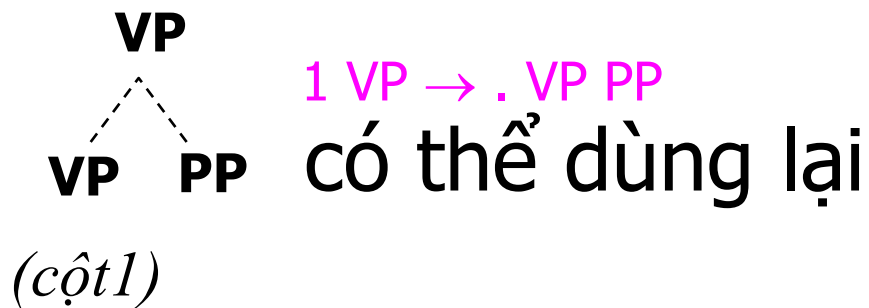
1 VP → . VP PP

**attach**

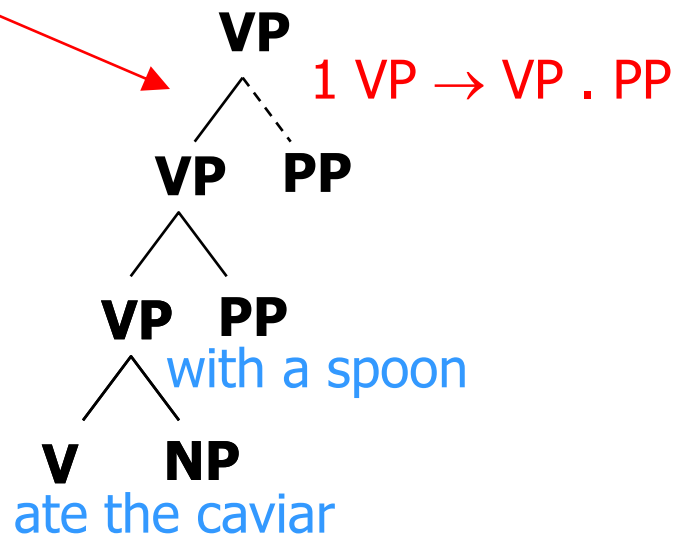


1 VP → VP . PP

# ... nhưng thuật toán Earley Ok!



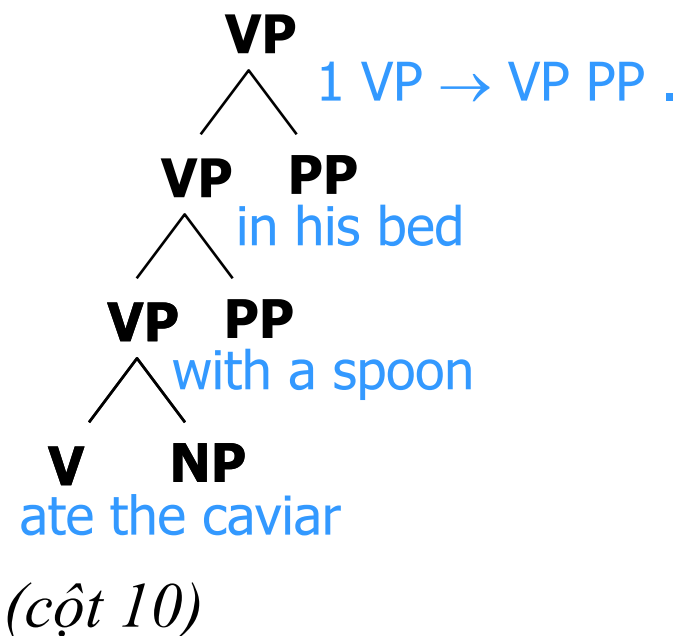
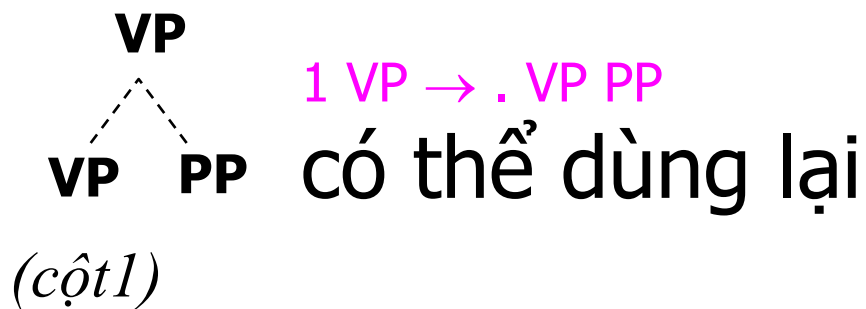
**attach**



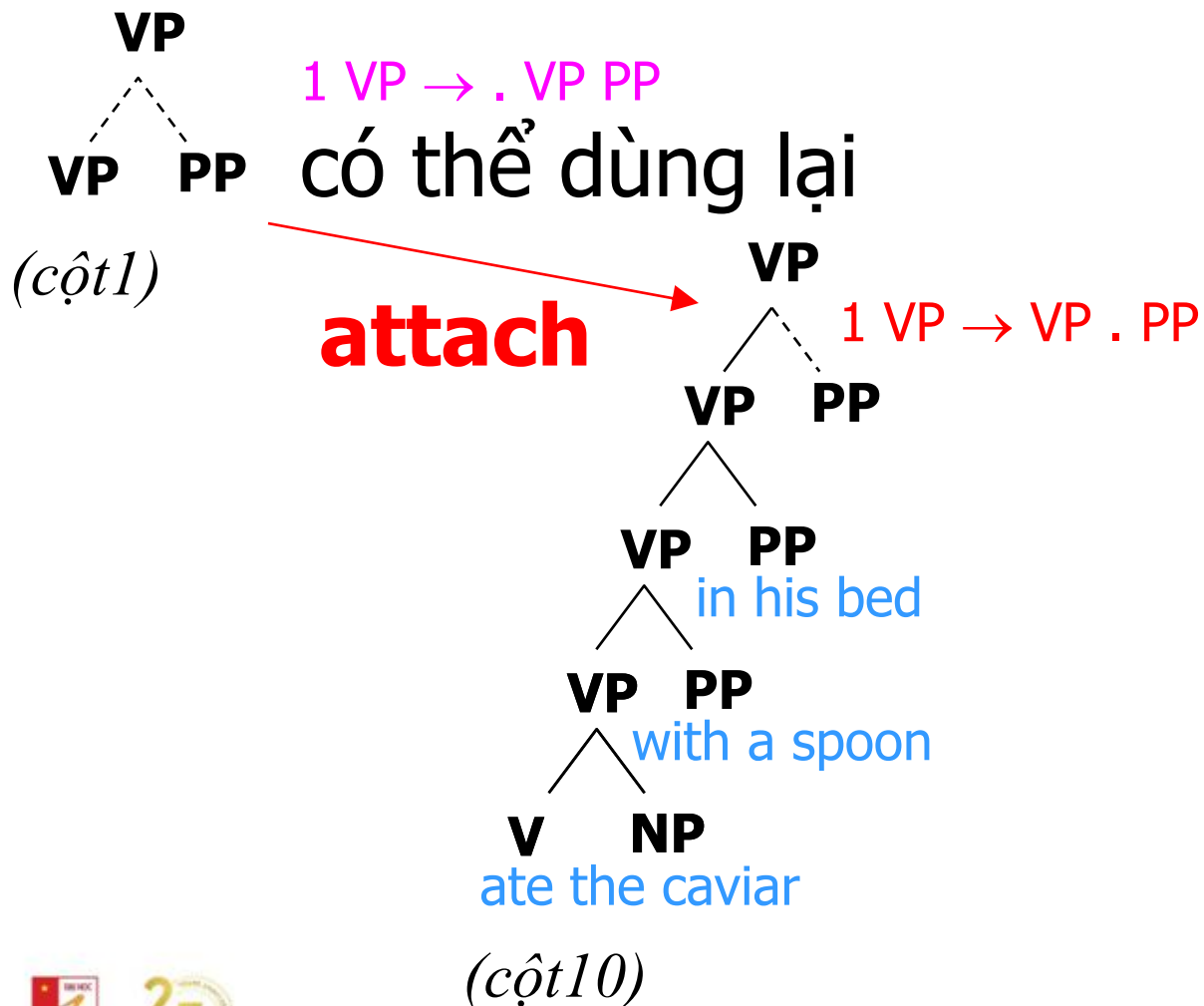
(cột 7)



# ... nhưng thuật toán Earley Ok!

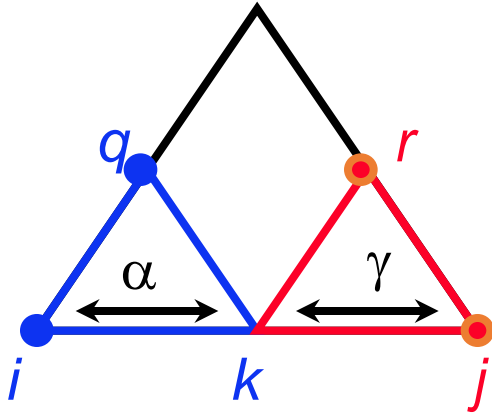


# ... nhưng thuật toán Earley Ok!



# Phục hồi cây cú pháp

$[s, i]$  trong tập trạng thái  $j$  Sử dụng thuật toán dùng queue đơn giản, dựa trên các thành phần có ích



- 1 thành phần ở trạng thái kết thúc là *có ích*
- If  $s = [A \rightarrow \alpha \bullet B, i]$  trong tập đích  $k$  & *có ích*
- then  $q = [A \rightarrow \alpha \bullet, k]$  & item  $r = [B \rightarrow \gamma \bullet, j]$  là *có ích*

$[s, i]$  : một thành phần với luật  $s$  & trả về con trỏ  $i$ .

Đánh dấu tất cả các thành phần trong tập trạng thái  $S_n$  ở dạng

$Root \rightarrow \alpha S \bullet, 0 \square$

for  $j = n$  downto 0 do

for  $i = 0$  to  $j$  do

for mọi bộ đã đánh dấu  $[s, i]$  trong tập trạng thái  $j$  do

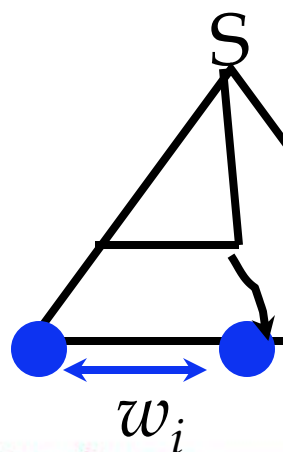
for  $k = i$  to  $j$  do

if  $[q, i] \in S_k$  &  $[r, k] \in S_j$  &  $s = q \otimes r$  then

đánh dấu  $[q, i]$  và  $[r, k]$

# Ưu điểm

- Thuật toán Earley thực hiện một vài phép lọc *top-down*: bất cứ thành phần nào (state, or triple) được đưa vào tập trạng thái cần tương thích với phần đã được sinh ra ở bên trái. Ví dụ:  $S \Rightarrow w_i$  \* trong đó  $w_i$  là phần của câu đã được duyệt qua



# Nhược điểm

- Biểu diễn luật: Cần cách biểu diễn luật tường minh: mất thời gian xây dựng
- Thực hiện phép lọc bên trái nhưng không lọc bên phải

Phép lọc nhìn trước cho ký hiệu không kết thúc  $A$ :

$$FIRST(A) = \{x | A \Rightarrow x\delta\}, x = 1 \text{ token}$$

v.d.,  $FIRST(S) = \text{who, did, the, etc.}$

# Các phương pháp khác

- Các phương pháp khác ứng với các cách khác nhau để tìm các đoạn
- Đoạn  $X[i, j]$  là đoạn có nhãn  $X$  phủ đầu vào từ  $i$  đến  $j$

Example:

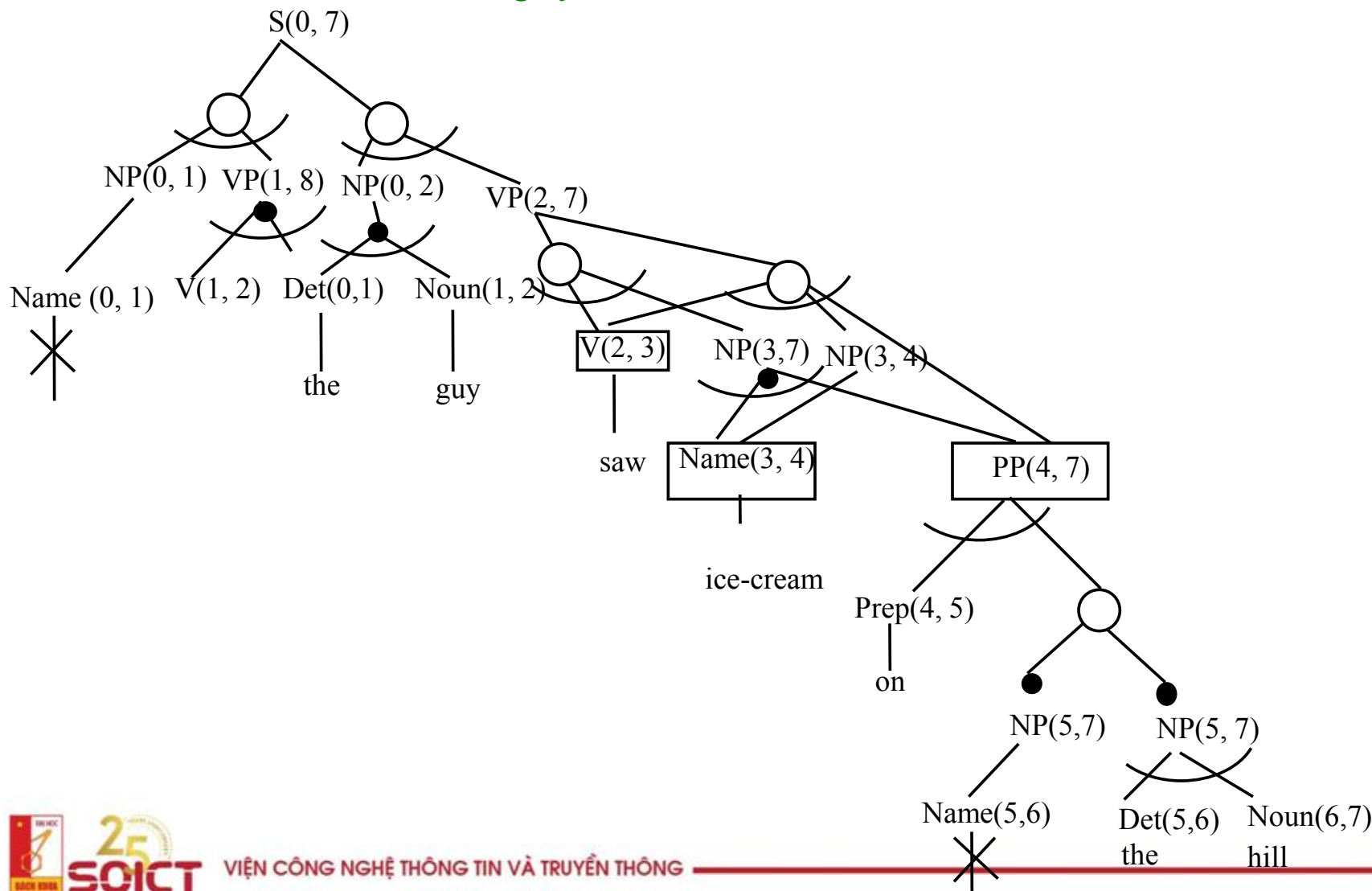
<sub>0</sub> *John* <sub>1</sub> *ate* <sub>2</sub> *ice-cream* <sub>3</sub> *on* <sub>4</sub> *the* <sub>5</sub> *table* <sub>6</sub>

PP[3,6]; S[0,6]; ...

- Biểu diễn không gian tìm kiếm như cây and-or
  - Disjuncts (or) = các đường phân tích khác nhau
  - Conjuncts (and) = vế phải của luật, ví dụ vế phải của  $S$  là NP VP

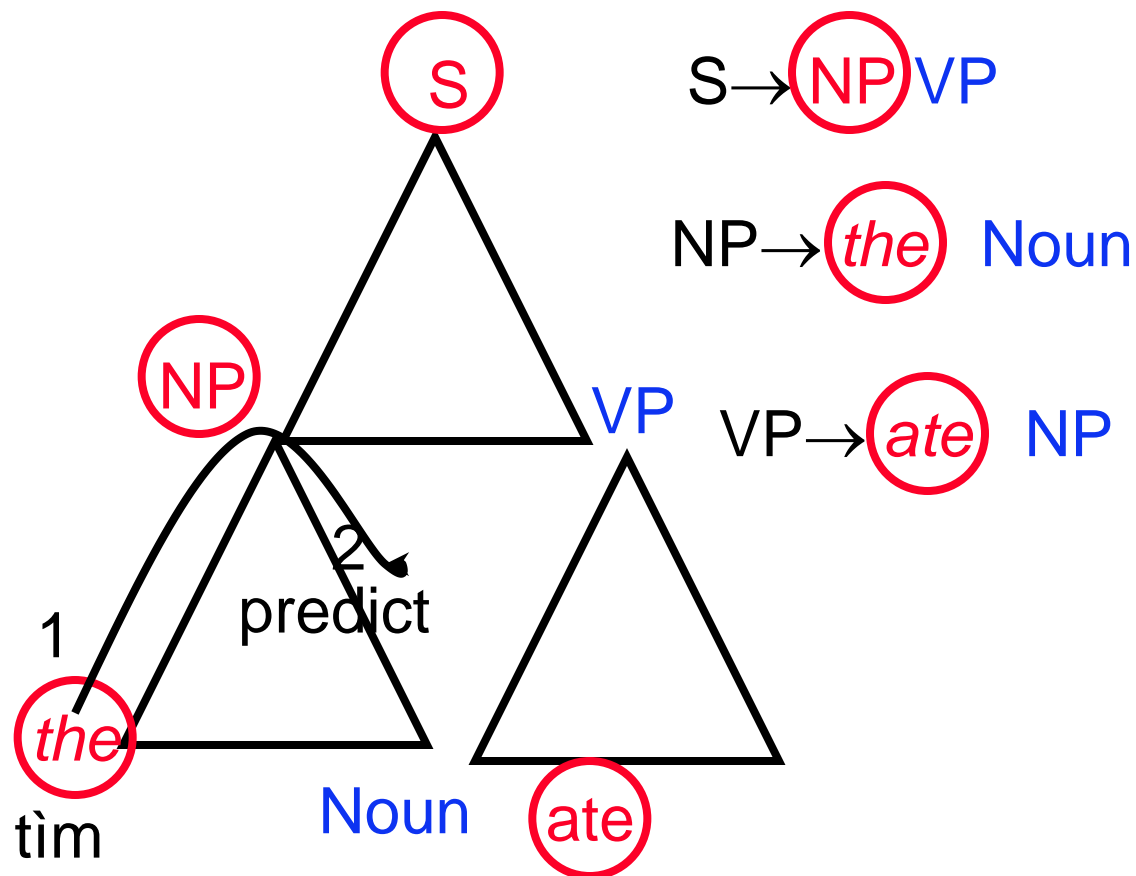
# PTCP là việc tìm kiếm

0 the 1 guy 2 saw 3 ice-cream 4 on 5 the 6 hill 7



# PTCP góc trái (Left-corner parsing)

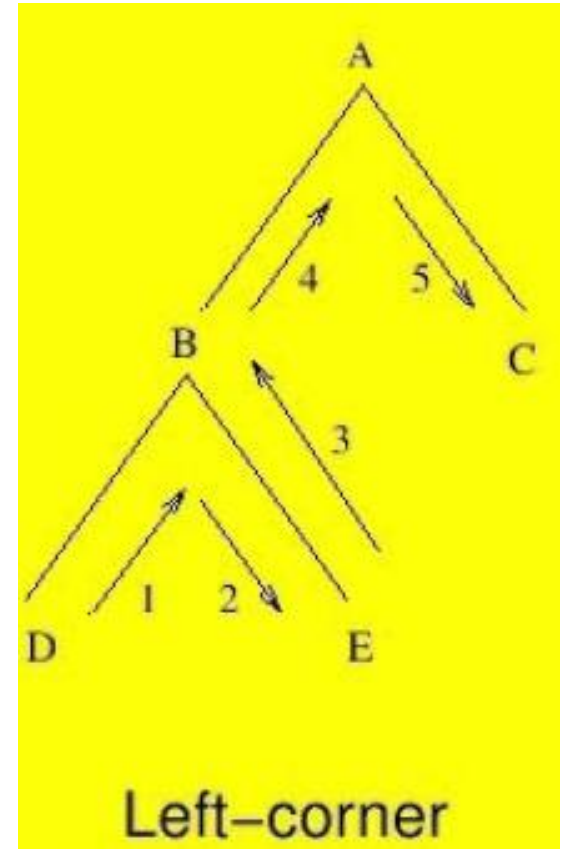
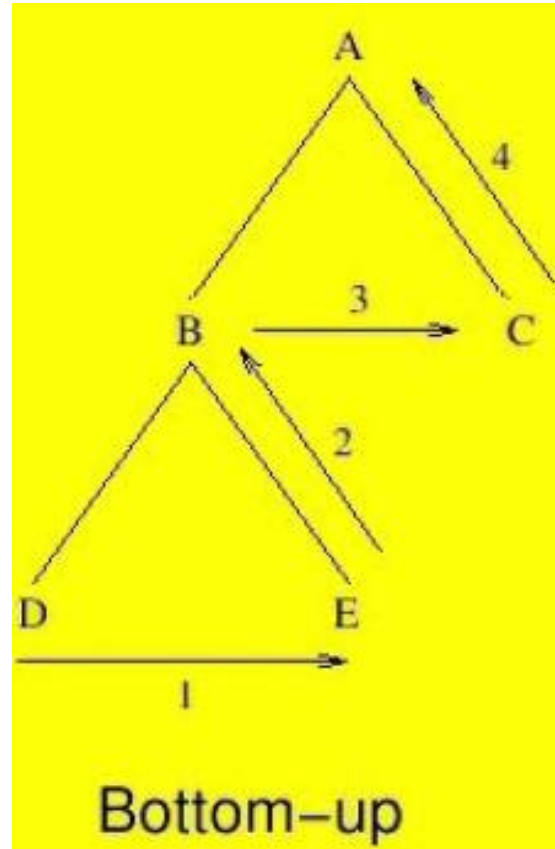
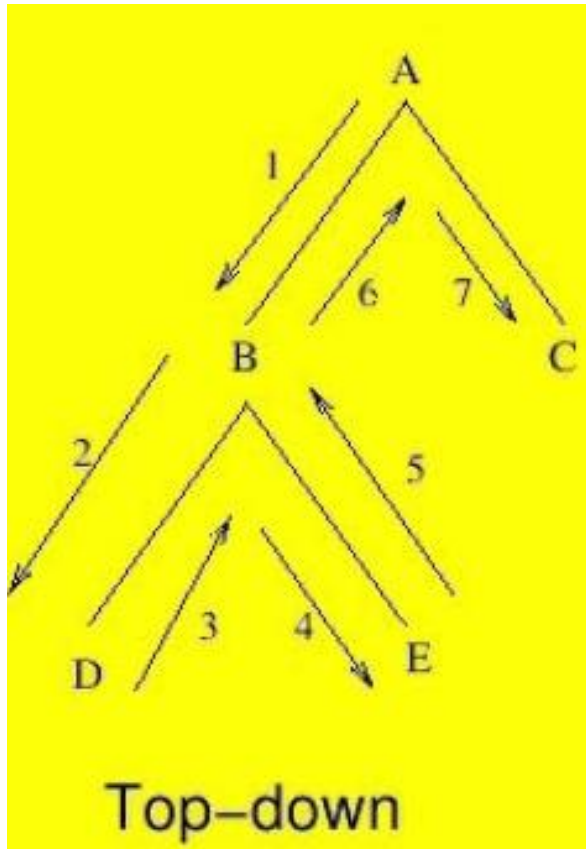
- Nhìn từ dưới lên để tìm ký hiệu đầu tiên (left-corner) của đoạn, sau đó phân tích phần còn lại theo kiểu trên xuống
- Tìm cách kết hợp các đặc trưng tốt nhất của phân tích trên xuống và dưới lên



Phương pháp này làm việc tốt với ngôn ngữ với thành phần quan trọng đặt ở đầu như tiếng Anh. Các tiếng Đức, Hà Lan, Nhật là ngôn ngữ có phần quan trọng đặt cuối.



# PTCP góc trái



Top down thường không quan tâm đến sâu đầu vào  
Bottom up không cần biết đang xây cái gì

# PTCP góc trái

- Rules:
  - $S \rightarrow NP VP$
  - $NP \rightarrow NN \mid DT NN \mid DT NNS \mid NNP$
  - $VP \rightarrow V \mid V NP$
  - $V \rightarrow VBZ \mid VBP \mid VBD$
- DT  $\rightarrow$  the | a
- NNS  $\rightarrow$  children
- NN  $\rightarrow$  table | song | plant
- NNP  $\rightarrow$  Kate
- VBZ  $\rightarrow$  sings
- VBP  $\rightarrow$  sing | plant
- VBD  $\rightarrow$  died
- Input:
  - Kate sings
  - Kate sings a song
  - The children sing a song
  - The plant died

# Vấn đề với PTCP góc trái

- Có vài luật cùng “góc trái”
- Nhập nhằng: có trường hợp có 2 khả năng:
  - Cấu trúc mới tạo có thể dùng để hoàn thành 1 cấu trúc đang xét
  - Cấu trúc mới tạo có thể là phần đầu của 1 cấu trúc mới khác

# Văn phạm phi ngữ cảnh

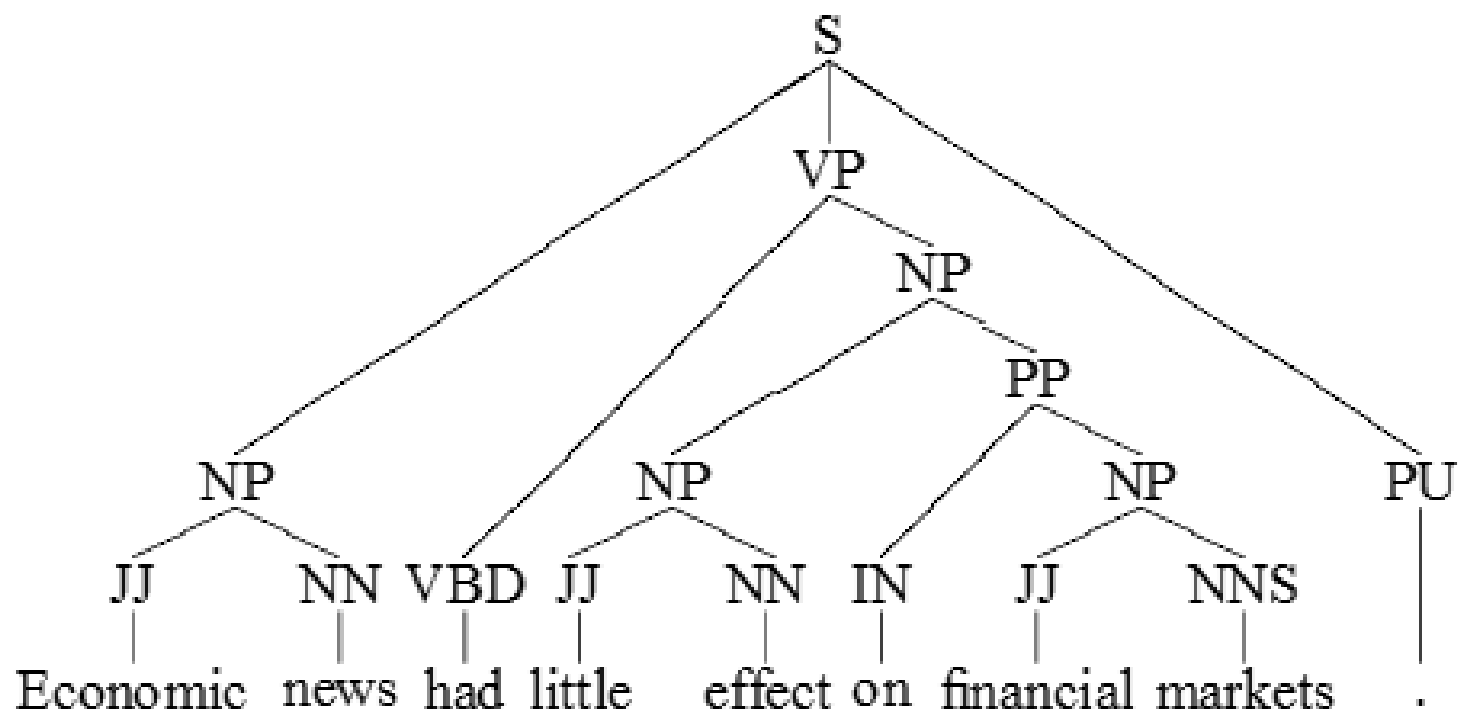


Figure 1: Constituent structure for English sentence from the Penn Treebank

# Văn phạm phụ thuộc

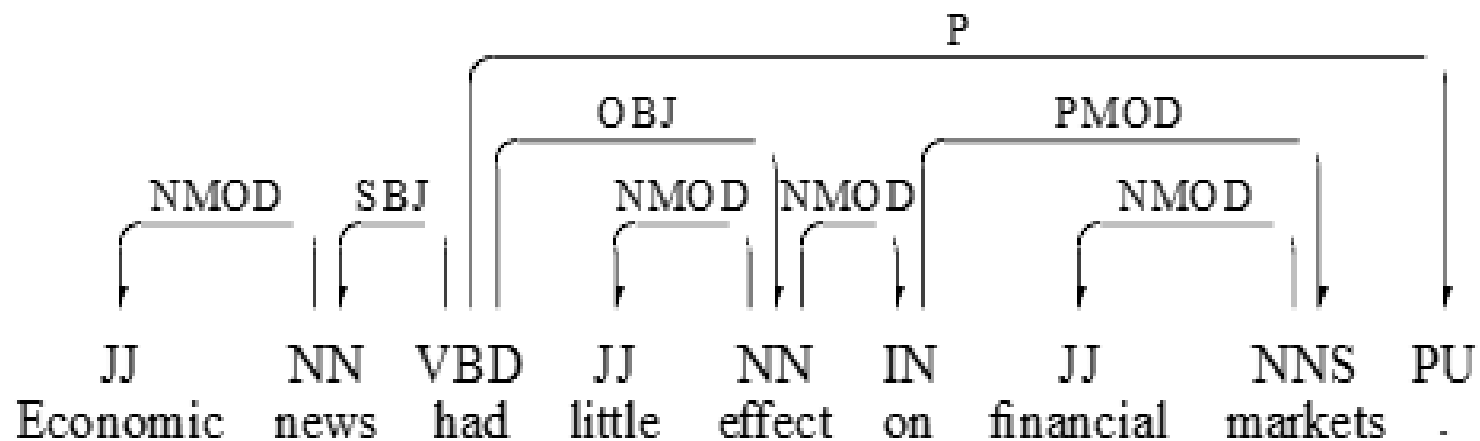


Figure 2: Dependency structure for English sentence from the Penn Treebank