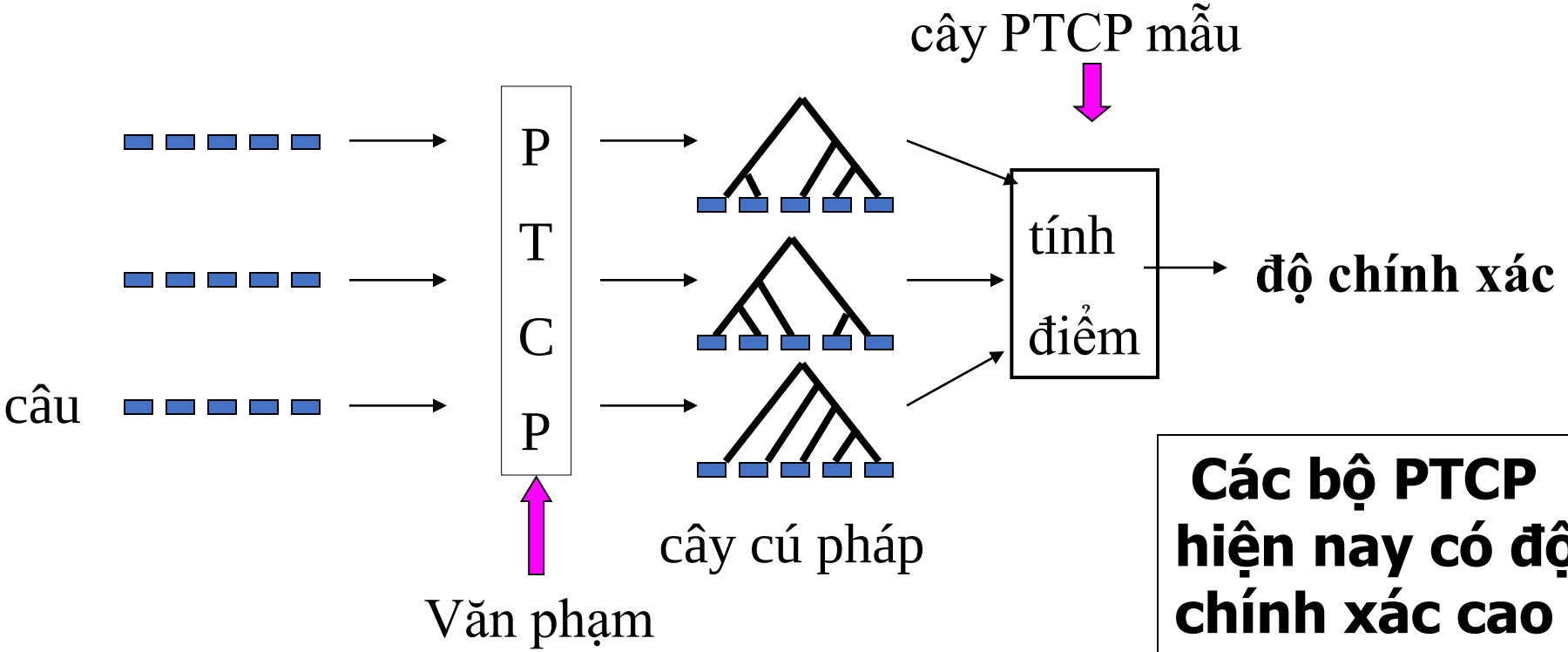
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Phân tích cú pháp

Viện Công nghệ Thông tin và Truyền thông

Bài toán PTCP

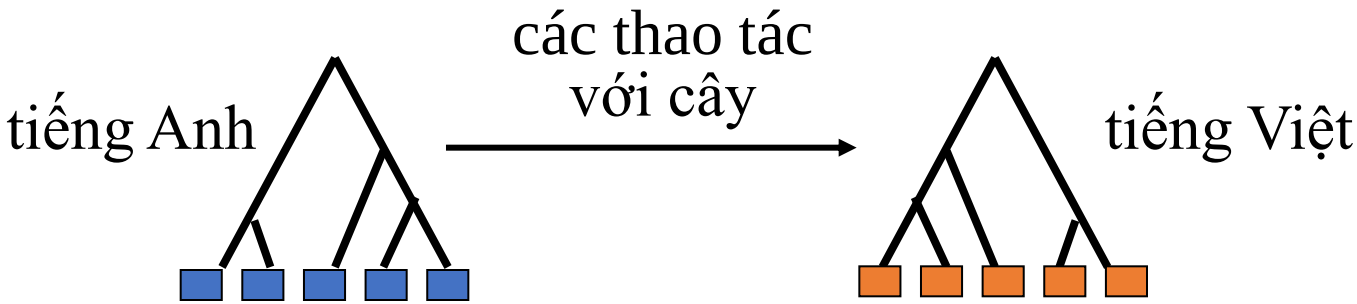


Các bộ PTCP hiện nay có độ chính xác cao

(Eisner, Collins, Charniak, etc.)

Các ứng dụng của PTCP

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



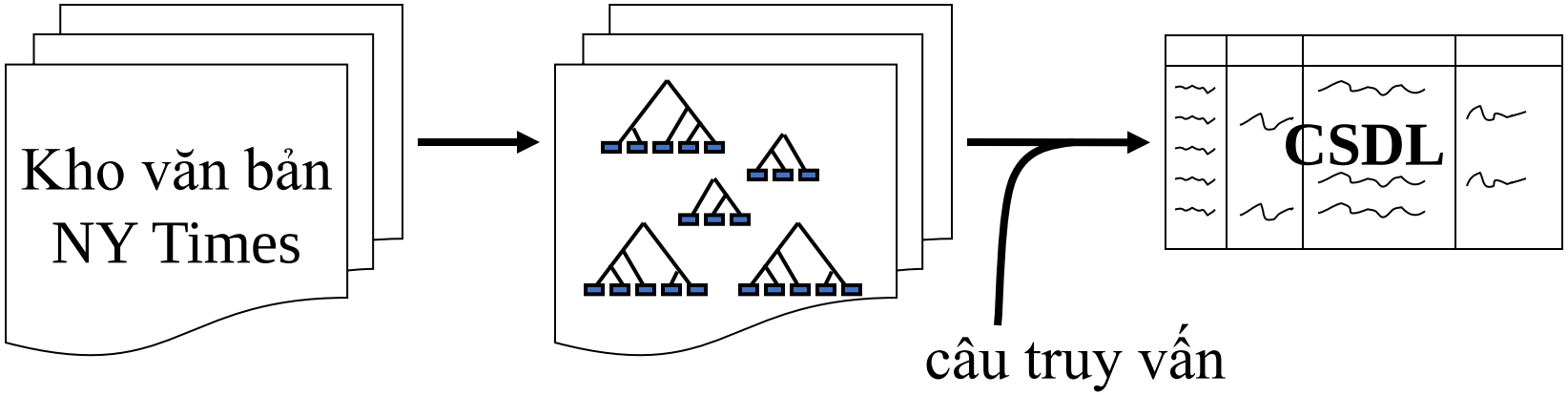
- Nhận dạng tiếng nói sử dụng PTCP (Chelba et al 1998)

  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:

  $S \rightarrow NP\ VP$

  $NP \rightarrow$ John, garbage

  $VP \rightarrow$ laughed, walks

G có thể sinh ra các câu sau:

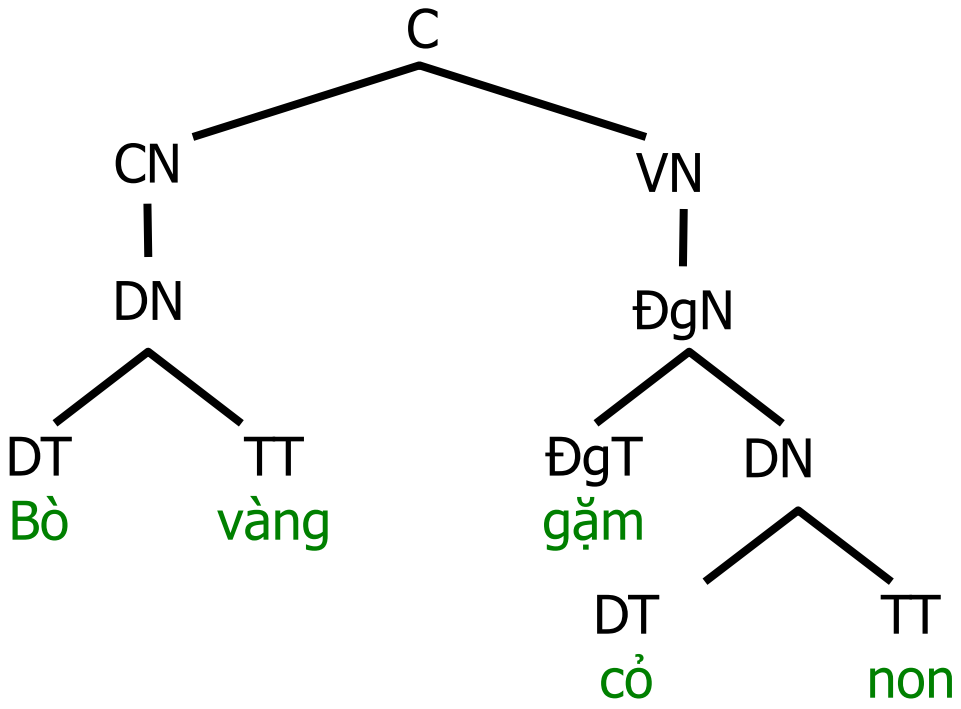
John laughed. John walks.

Garbage laughed. Garbage walks.

Ví dụ về văn phạm

Phân tích câu “Bò vàng gặm cỏ non”

- Cây cú pháp:
- Tập luật
  - C → CN VN
  - CN → DN
  - VN → ĐgN
  - ĐgN → ĐgT DN
  - DN → 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 = { $\alpha \rightarrow \beta$ | $\alpha, \beta \in (T \cup N)^*$ }

    $\alpha \rightarrow \beta$ gọi là luật sản xuất

Ví dụ

- G1 = ({a,b}, {X}, X, {X→ε, X→aXb})
  Xác định L(G1)

- G2 = ({a,b}, {X}, X, {X→ε, X→aXb, X→XX})
  Xác định L(G2)

Dạng chuẩn Chomsky

- Mọi NNPNC không chứa ε đều có thể sinh từ một văn phạm trong đó mọi sản xuất đều có dạng A → BC hoặc A → 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 → bA|aB
  - A →bAA|aS|a
  - B → aBB|bS|b

Văn phạm phi ngữ cảnh (Context-Free Grammar)

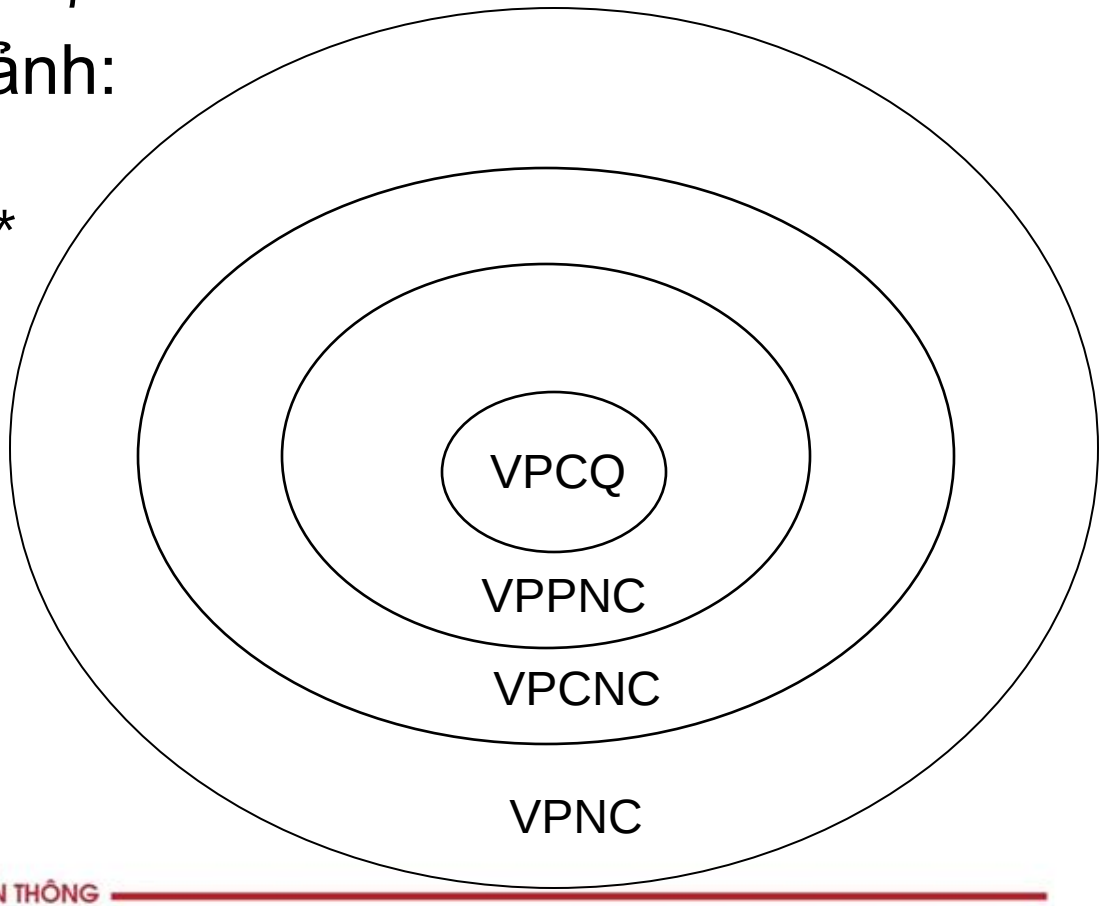
… còn gọi là văn phạm cấu trúc đoạn

- G = <T,N,P,S,R>
  - 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; γ 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: $\alpha A \gamma \Rightarrow \alpha \beta \gamma$

- 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 $A, B \in N$, $a \in T$.



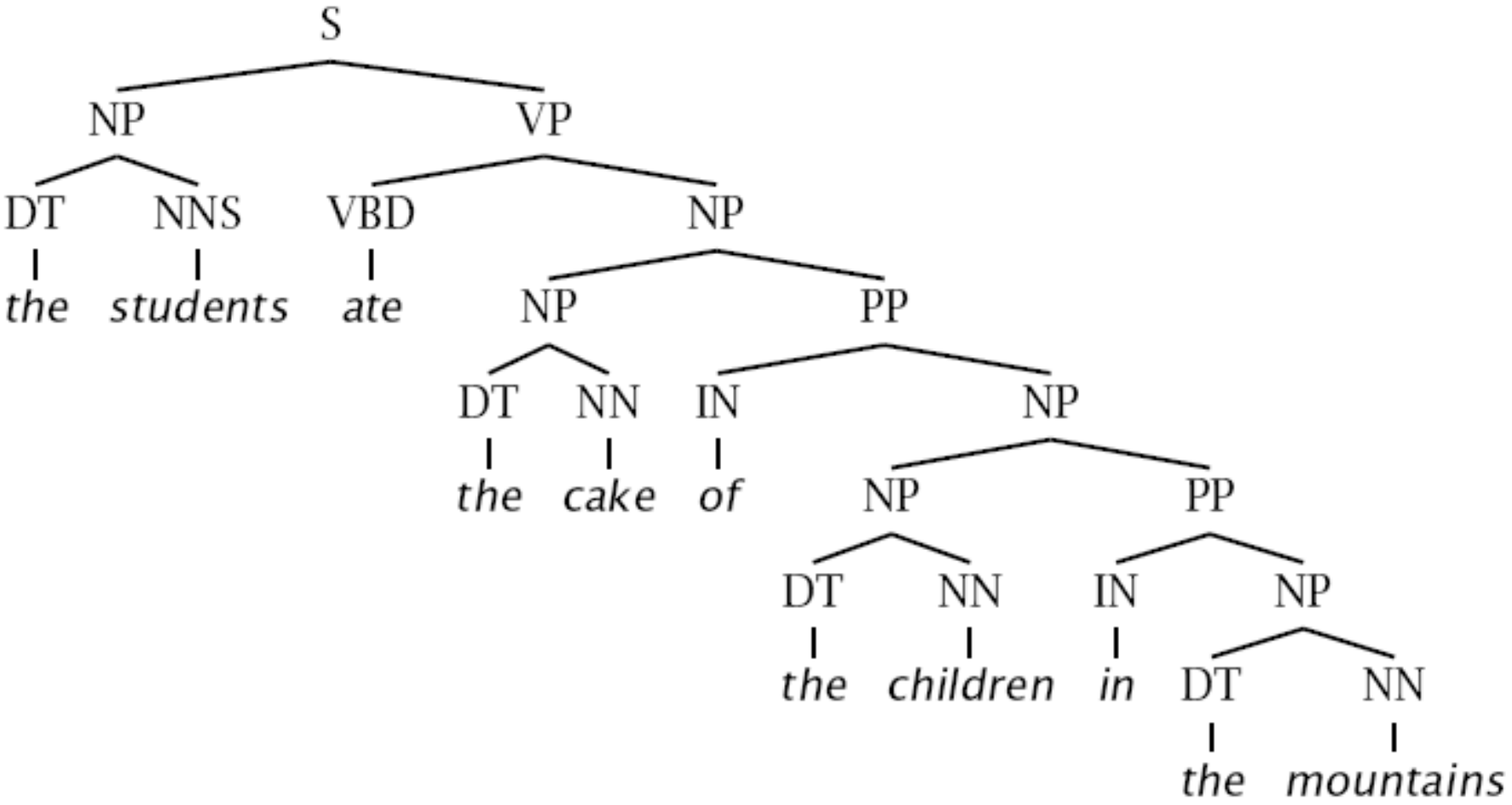
Văn phạm phi ngữ cảnh

$$
\begin{array}{lcl}
S & \rightarrow & NP\ VP \\
NP & \rightarrow & \{DT\ NNS,\ DT\ NN,\ NP\ PP\} \\
VP & \rightarrow & \{VP\ PP,\ VBD,\ VBD\ NP\} \\
PP & \rightarrow & IN\ NP
\end{array}
\qquad
\begin{array}{lcl}
DT & \rightarrow & \{the\} \\
NNS & \rightarrow & \{children,\ students,\ mountains\} \\
VBD & \rightarrow & \{slept,\ ate,\ saw\} \\
IN & \rightarrow & \{in,\ of\} \\
NN & \rightarrow & \{cake\}
\end{array}
$$

Áp dụng tập luật ngữ pháp

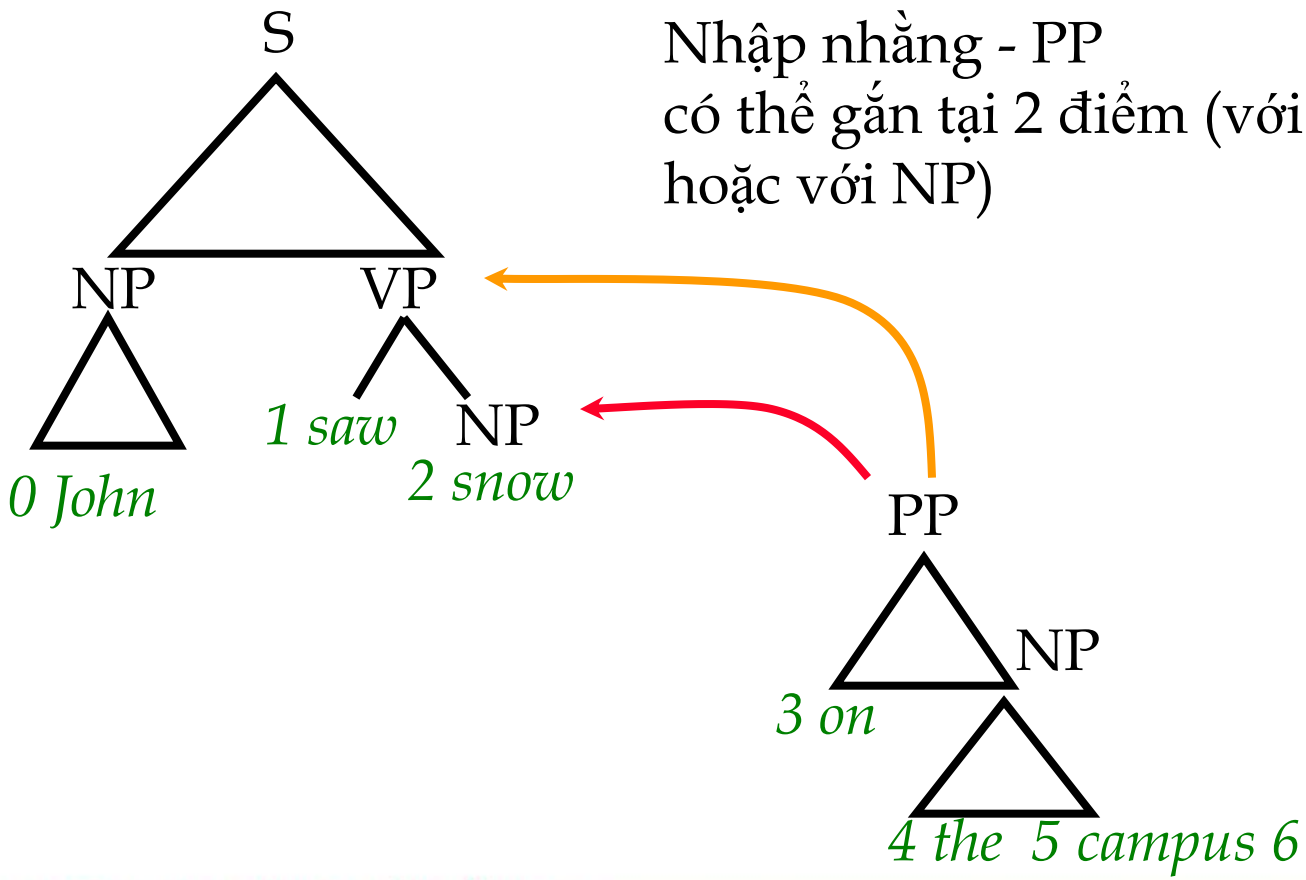
- S
  - → NP VP
  - → DT NNS VBD
  - → *The children slept*
- S
  - → NP VP
  - → DT NNS VBD NP
  - → DT NNS VBD DT NN
  - → *The children ate the cake*

Cấu trúc đoạn đệ qui



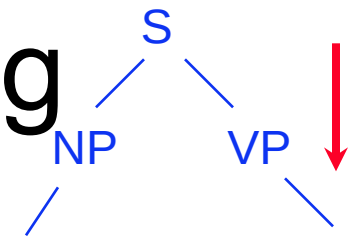
Văn phạm cho ngôn ngữ tự nhiên có nhập nhằng

John saw snow on the campus



Nhập nhằng - PP
có thể gắn tại 2 điểm (với VP hoặc với NP)

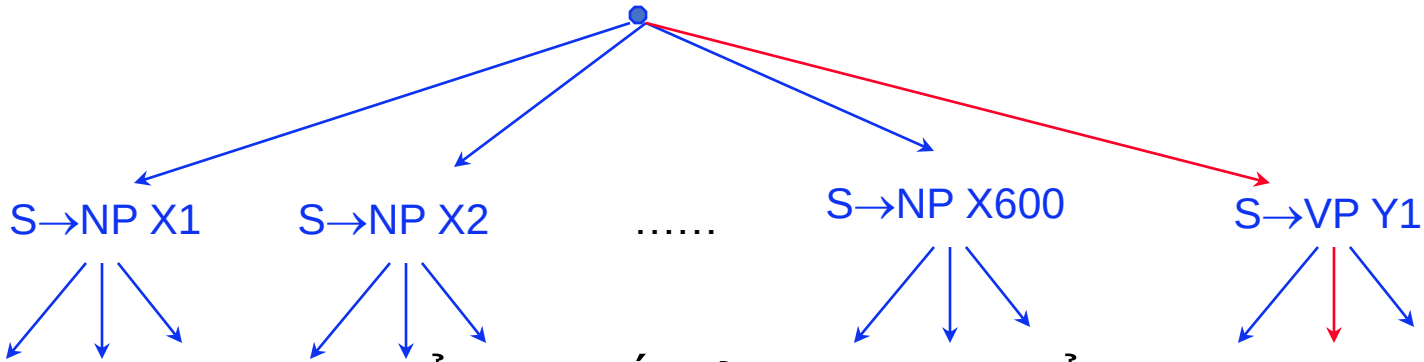
PTCP kiểu trên xuống



- 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
  - triểu 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)

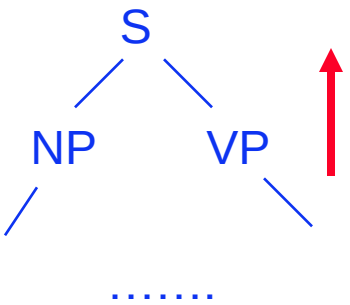
Khó khăn với PTCP trên xuống

- Các luật đệ qui 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 $\rightarrow$ 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.