CHƯƠNG TRÌNH DỊCH

Bài 6: Phân tích cú pháp

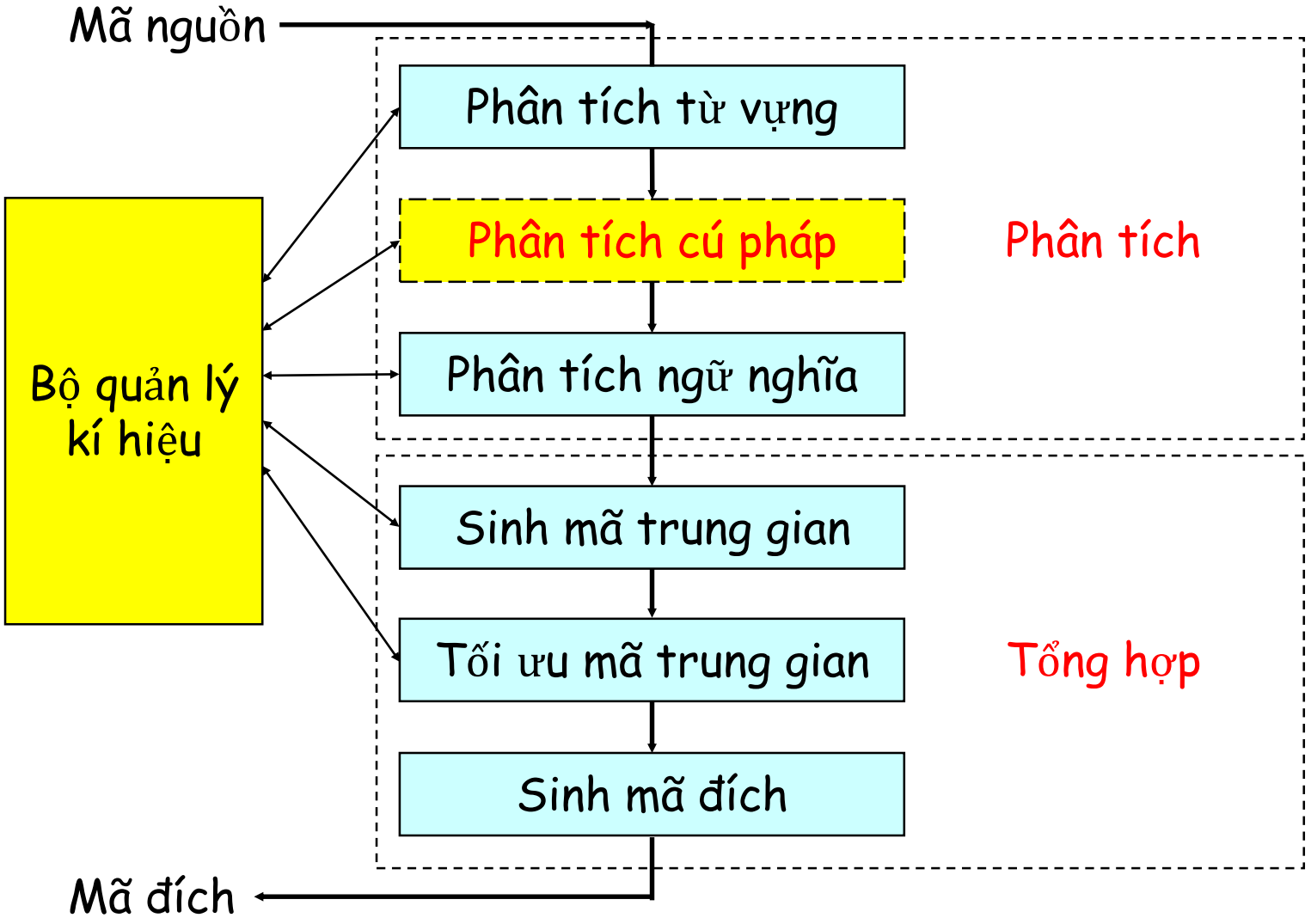
Nội dung

1. Vai trò của bộ phân tích cú pháp (PTCP)
2. Nhiệm vụ và các mục tiêu của PTCP
3. Đầu vào và đầu ra của PTCP
4. Các bước xây dựng bộ PTCP
5. Suy dẫn và biểu diễn suy dẫn bằng cấu trúc cây
6. Văn phạm có nhập nhằng
7. Các chiến lược phân tích cú pháp
8. Thảo luận và bài tập

Phần 1

Vai trò của bộ của phân tích cú pháp (PTCP)

Cấu trúc một chương trình dịch

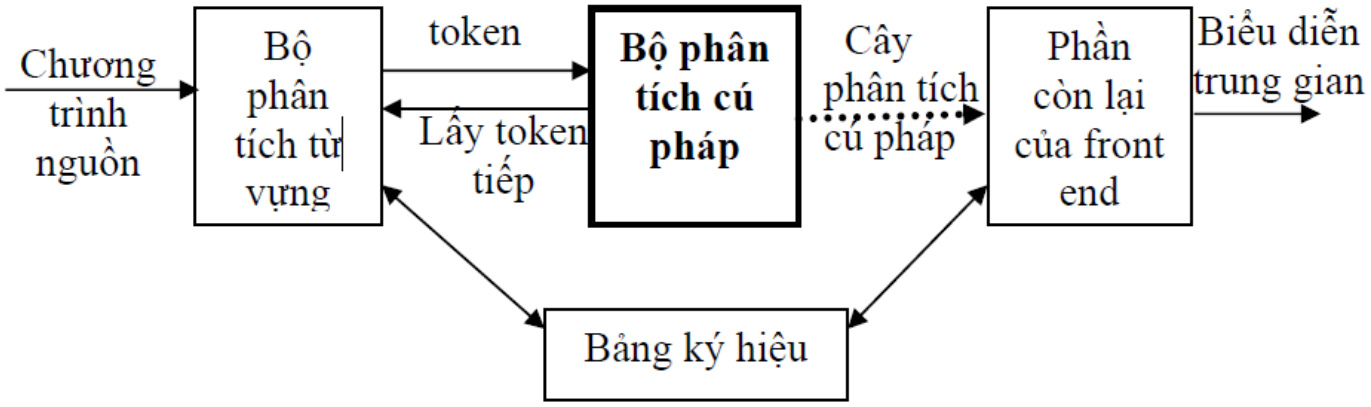


Vai trò của bộ phân tích cú pháp

- Phân tích cú pháp là bước thứ hai của trình dịch
- Bộ PTCP nhận dữ liệu đầu vào là dãy các từ tố (cùng với các thông tin kèm theo), dựa theo các luật văn phạm của ngôn ngữ, xây dựng cây cú pháp (syntax tree) của chuỗi vào
  - PTCP làm việc chặt chẽ với PTTV và thường có thể bắt đầu thực hiện công việc ngay khi PTTV mới có những kết quả ban đầu (không cần đợi PTTV kết thúc)
  - Đối với một số ngôn ngữ đơn giản, thiết kế trình dịch còn đi xa hơn bằng cách ghép PTTV và PTCP thành một module duy nhất (dịch trực tiếp văn bản)

Vai trò của bộ phân tích cú pháp

- PTCP cung cấp dữ liệu cho bộ phân tích ngữ nghĩa
  - Làm việc độc lập với bộ PTNN
  - Chỉ trả về kết quả cho phân tích ngữ nghĩa sau khi đã hoàn thành đầy đủ (hoặc tương đối đầy đủ) việc tạo cây cú pháp
- Ngoài ra PTCP cũng cung cấp dữ liệu về lỗi và gợi ý sửa lỗi cho bộ soạn thảo



Vai trò của bộ phân tích cú pháp

- Trái với bộ PTTV, thường được "đính kèm" nhiều nhiệm vụ khác, thiết kế bộ PTCP thường chỉ nhắm tới nhiệm vụ duy nhất là dựng cây cú pháp
  - Lý do chính là việc xây bộ PTCP hiệu quả khá phức tạp
- Tuy PTCP trả về kết quả cho PTNN, nhưng trong một số thiết kế, chính bộ PTCP sẽ quyết định khi nào thực hiện phân tích ngữ nghĩa dựa trên các điều kiện khởi động (trigger)
  - Chẳng hạn: sau khi kết thúc khai báo một class (trong ngôn ngữ C++) thì có thể tiến hành luôn việc phân tích ngữ nghĩa của class đó

Phần 2

Nhiệm vụ và các mục tiêu của phân tích cú pháp

Nhiệm vụ của phân tích cú pháp

- PTCP đảm nhiệm nhiệm vụ phức tạp nhất của trình dịch, đó là kiểm tra lỗi cú pháp của chuỗi vào (vốn có thể làm sai lệch hoàn toàn ý nghĩa của input)
- Các nhiệm vụ chính (nhất thiết phải có để đảm bảo hoạt động của chương trình dịch):
  - Xây dựng cây cú pháp cho chuỗi vào
  - Thực hiện một số thao tác ngữ nghĩa phục vụ cho việc phân tích tiếp theo
  - Phát hiện các lỗi về văn phạm và lựa chọn phương pháp xử lý phù hợp
    - Xử lý lỗi để tiếp tục thực hiện việc phân tích
    - Đưa ra các gợi ý sửa lỗi cho mã nguồn

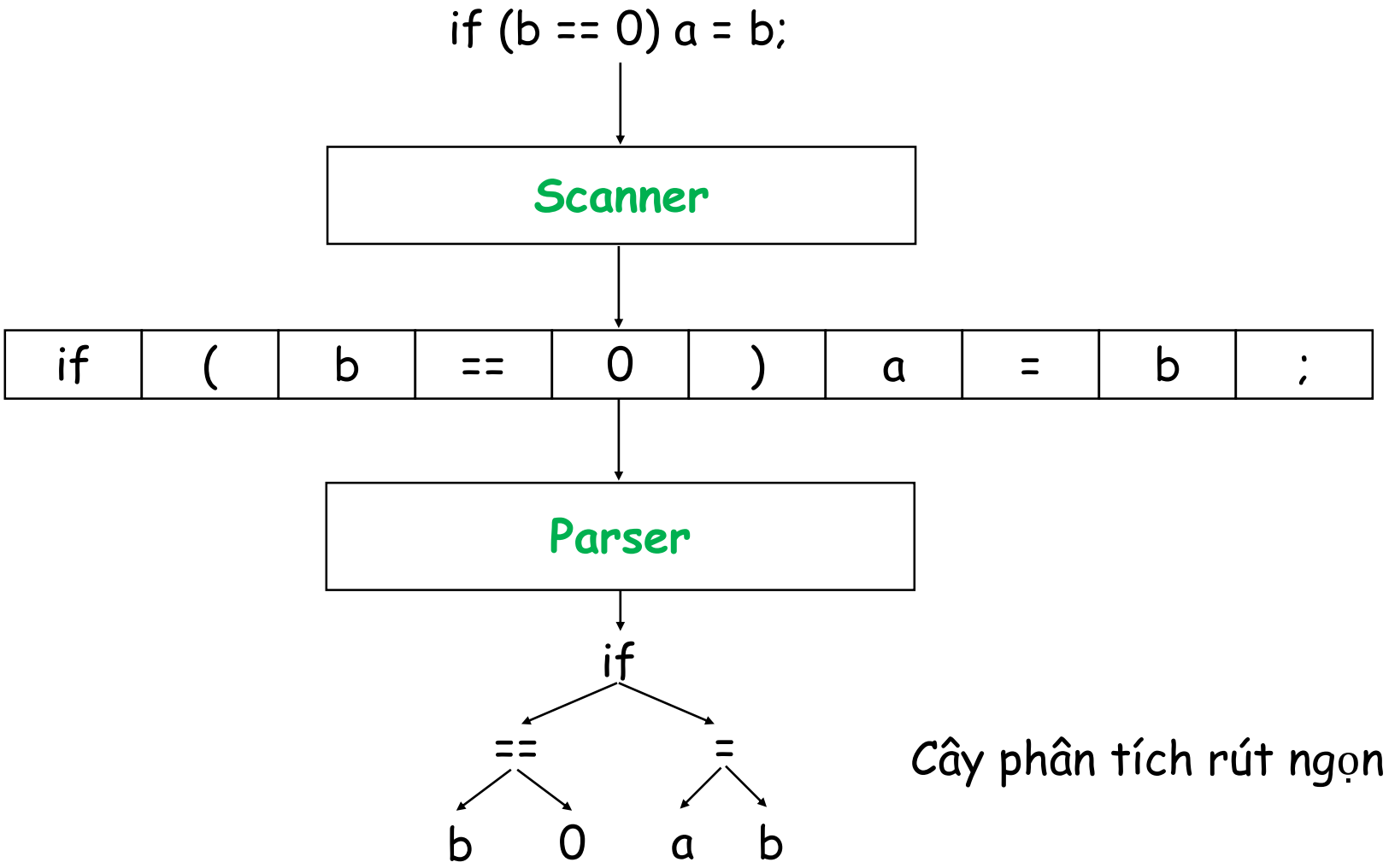
Các mục tiêu của PTCP

- **Chính xác**: đây là mục tiêu quan trọng nhất, kết quả phân tích cần trả về chính xác cây phân tích
- **Tốc độ**: khó xây dựng các bộ PTCP tuyến tính theo độ dài của chuỗi vào (ngoại trừ ngôn ngữ đầu vào có văn phạm quá đơn giản), nhưng bộ PTCP cần hoạt động đủ nhanh (nên cận tuyến tính)
- **Chịu lỗi**: bộ PTCP cần có khả năng chịu lỗi và có chiến lược khắc phục lỗi phù hợp
- **Hiệu quả về bộ nhớ**: bộ PTCP cần sử dụng bộ nhớ một cách hiệu quả (do việc phải lưu trữ toàn bộ cây phân tích cho mã nguồn)

Phần 3

Đầu vào và đầu ra của phân tích cú pháp

Ví dụ về phân tích cú pháp



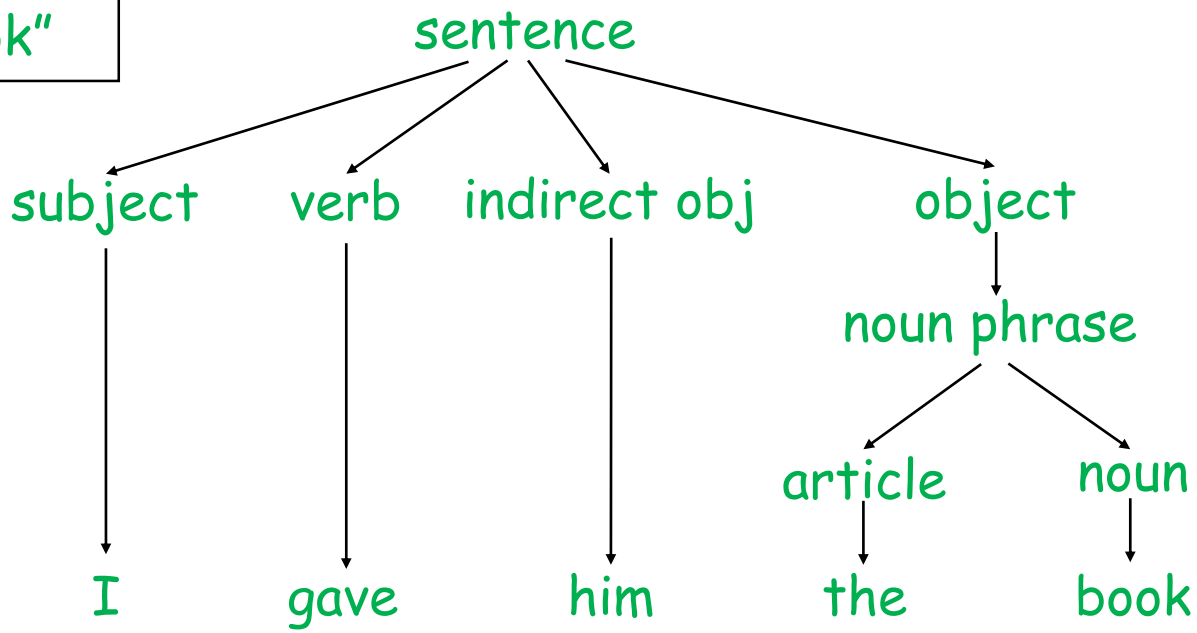
Cây phân tích rút ngọn

Ví dụ về phân tích cú pháp

PTCP có thể áp dụng với ngôn ngữ tự nhiên, nhưng:

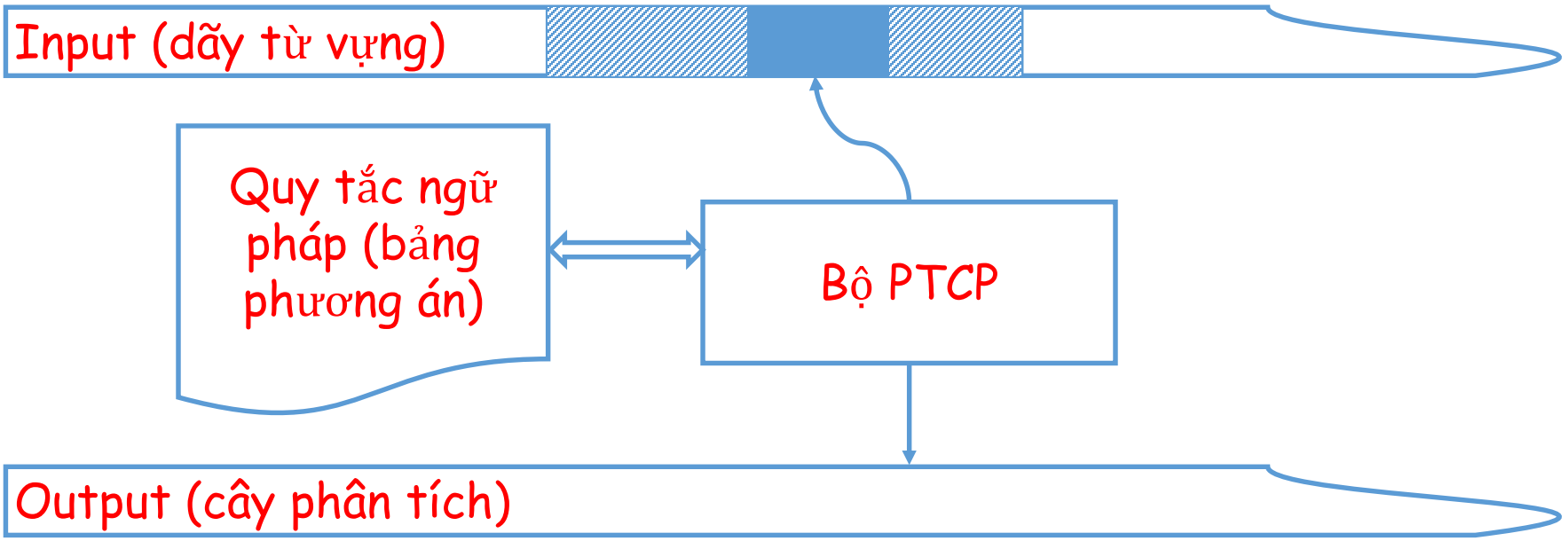
- Ngôn ngữ tự nhiên có luật ngữ pháp phức tạp
- Ngôn ngữ tự nhiên có yếu tố ngữ cảnh
- Ngôn ngữ tự nhiên có nhiều lỗi (trong thực tế sử dụng)

"I gave him the book"



Đầu vào của bộ PTCP

- Đầu vào của bộ PTCP là dãy các từ vựng đã được xác định chi tiết về từ loại
- Bộ PTCP thường cần quan sát nhiều hơn 1 kí hiệu đầu vào để ra quyết định dựng cây phân tích



Đầu ra của bộ PTCP

- Đầu ra của bộ PTCP là đầu vào của bộ PTNN, thường thì chỉ có thể hiểu đúng ngữ nghĩa khi đã xác định đầy đủ cấu trúc của câu, vì thế bộ PTCP thường trả về cây phân tích đầy đủ cho bộ PTNN
- Cây phân tích thường có các thành phần sau:
  - Cấu trúc cây (có nhiều lựa chọn kiểu cấu trúc dữ liệu)
  - Cấu trúc một nút cây:
    - Kí hiệu ở nút hiện tại
    - Từ vựng liên quan (nếu là nút lá)
    - Danh sách các nút con
    - Thông tin bổ sung, phục vụ cho việc phân tích tiếp theo

Phần 4

Các bước xây dựng bộ phân tích cú pháp

Các bước xây dựng bộ PTCP

- Mô tả các luật văn phạm của ngôn ngữ nguồn
  - Các mô tả này ban đầu có thể ở dạng ngôn ngữ tự nhiên
  - Đặc tả ý nghĩa các kí hiệu không kết thúc (non-terminal)
  - Chuyển thành các luật văn phạm ở dạng chặt chẽ
- Phân tích bộ văn phạm để lựa chọn phương pháp phân tích cú pháp phù hợp nhất
  - Văn phạm có nhập nhằng hay không?
  - Văn phạm có đệ quy trái hay không?
  - Văn phạm có sự mơ hồ hay không?
  - Văn phạm có độ phức tạp ở mức độ nào?

Các bước xây dựng bộ PTCP

- Lựa chọn phương pháp phân tích cú pháp phù hợp
  - Xây dựng bộ PTCP trực tiếp (dành cho các ngôn ngữ có độ phức tạp đơn giản)
  - Xây dựng bộ PTCP 2 bước
    - Dựa trên văn phạm đầu vào, xây dựng automat đoán nhận
    - Sử dụng automat để xử lý dãy từ tố từ PTTV
  - Xây dựng bộ PTCP vạn năng: trường hợp văn phạm quá phức tạp, có thể sử dụng các phương pháp phân tích vạn năng để xây dựng bộ PTCP
- Lựa chọn cách xử lý trong tình huống lỗi cú pháp, sinh các gợi ý sửa lỗi và các tình huống cần phải tổ hợp ngữ nghĩa

Phần 5

Suy dẫn

Suy dẫn

- Khái niệm: $\alpha A\beta \Rightarrow \alpha\gamma\beta$ (gọi là $\alpha A\beta$ suy dẫn ra $\alpha\gamma\beta$) nếu $A \rightarrow \gamma$ là một luật sinh, α và β là các chuỗi ký hiệu thuộc ngôn ngữ L nào đó
- Nếu $\alpha^1 \Rightarrow \alpha^2 \Rightarrow \ldots \Rightarrow \alpha^n$ ta nói α^1 suy dẫn ra α^n
- Hệ thống kí hiệu:
  - $\Rightarrow$ suy dẫn trực tiếp
  - $\Rightarrow^*$ suy dẫn ra qua 0 hoặc nhiều bước
  - $\Rightarrow^+$ suy dẫn ra qua 1 hoặc nhiều bước
- Một số tính chất:
  - $\alpha \Rightarrow^* \alpha$ với $\forall\alpha$
  - $\alpha \Rightarrow^* \beta$ và $\beta \Rightarrow^* \gamma$ thì $\alpha \Rightarrow^* \gamma$

Suy dẫn trái và suy dẫn phải

- Bài toán phân tích cú pháp thực chất là bài toán tìm chuỗi suy dẫn $S \Rightarrow^* \alpha \Rightarrow^* \beta$, trong đó:
  - S là kí hiệu gốc
  - α là chuỗi có chứa kí hiệu trung gian
  - β là chuỗi chỉ gồm các kí hiệu kết thúc
- Dễ nhận thấy trong quá trình suy dẫn trên:
  - Có nhiều phương án suy dẫn từ S thành β
  - Một kí hiệu trung gian thuộc α thì trước sau gì nó cũng phải bị biến đổi bởi một luật sinh nào đó
  - Nếu kí hiệu trung gian được chọn để biến đổi luôn là trái nhất của α thì ta gọi phương án này là suy dẫn trái
  - Định nghĩa tương tự cho suy dẫn phải

Suy dẫn trái và suy dẫn phải

- Cho văn phạm G với các luật sinh:

  $S \rightarrow E + S \mid E \qquad E \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid (S)$

- Xâu vào: W = (1 + 2 + (3 + 4)) + 5
- Suy dẫn trái từ S thành W như sau:

$$S \Rightarrow E + S \Rightarrow (S) + S \Rightarrow (E + S) + S \Rightarrow (1 + S) + S$$
$$\Rightarrow (1 + E + S) + S \Rightarrow (1 + 2 + S) + S$$
$$\Rightarrow (1 + 2 + E) + S \Rightarrow (1 + 2 + (S)) + S$$
$$\Rightarrow (1 + 2 + (E + S)) + S \Rightarrow (1 + 2 + (3 + S)) + S$$
$$\Rightarrow (1 + 2 + (3 + E)) + S \Rightarrow (1 + 2 + (3 + 4)) + S$$
$$\Rightarrow (1 + 2 + (3 + 4)) + E \Rightarrow (1 + 2 + (3 + 4)) + 5$$

Suy dẫn trái và suy dẫn phải

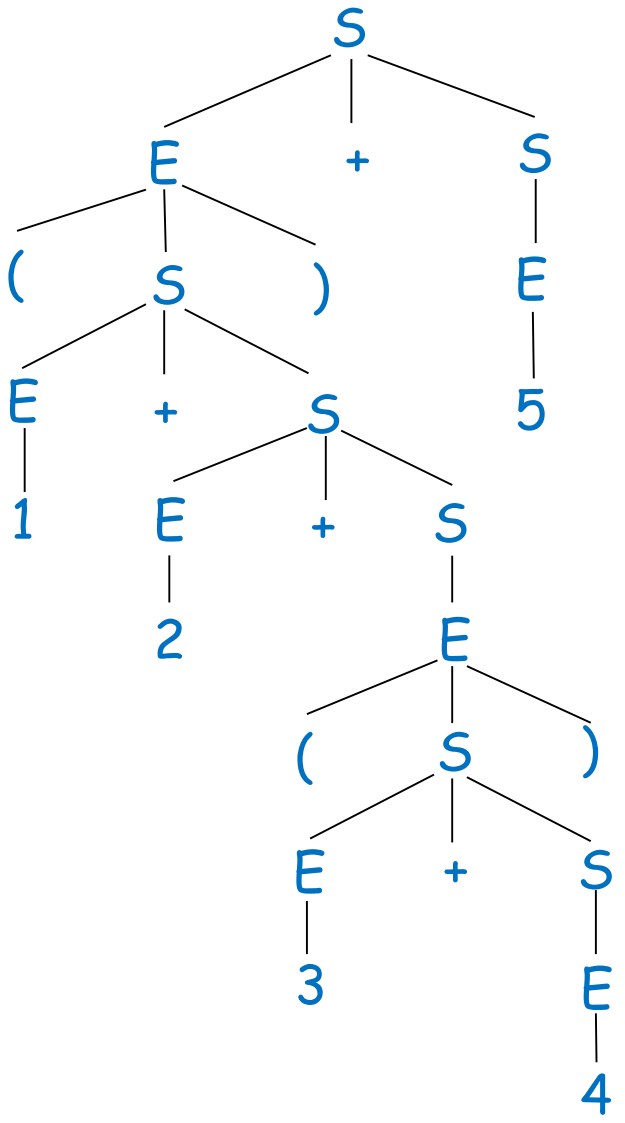
- Suy dẫn phải từ S thành W như sau:

$$S \Rightarrow E + S \Rightarrow E + E \Rightarrow E + 5 \Rightarrow (S) + 5 \Rightarrow (E + S) + 5$$
$$\Rightarrow (E + E + S) + 5 \Rightarrow (E + E + E) + 5$$
$$\Rightarrow (E + E + (S)) + 5 \Rightarrow (E + E + (E + S)) + 5$$
$$\Rightarrow (E + E + (E + E)) + 5 \Rightarrow (E + E + (E + 4)) + 5$$
$$\Rightarrow (E + E + (3 + 4)) + 5 \Rightarrow (E + 2 + (3 + 4)) + 5$$
$$\Rightarrow (1 + 2 + (3 + 4)) + 5$$

- *Câu hỏi: chúng ta nên sử dụng cách mã hóa như thế nào để lưu trữ quá trình suy dẫn và sử dụng các thông tin đó để in ra quá trình suy dẫn như thế nào?*

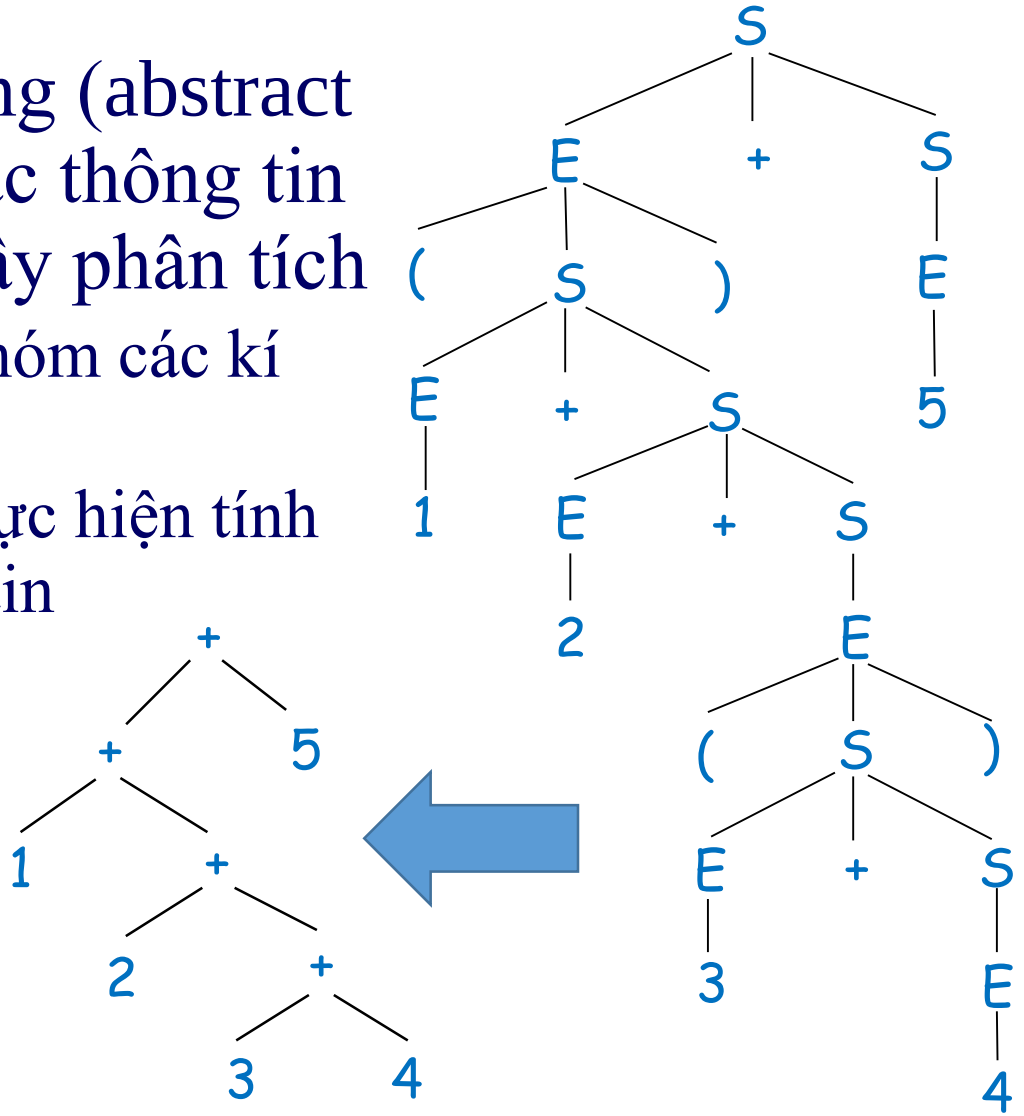
Cây phân tích (parse tree)

- Cây phân tích thể hiện cấu trúc của một suy dẫn
  - Nút gốc là kí hiệu bắt đầu
  - Các nút lá luôn là kí hiệu kết thúc
  - Các nút trong luôn là các kí hiệu trung gian
  - Cây không thể hiện thứ tự thực hiện các suy dẫn trực tiếp
    - Việc duyệt cây sẽ tạo thành thứ tự thực hiện suy dẫn
    - Suy dẫn trái tương đương với quá trình duyệt cây theo thứ tự giữa-trái-phải



Cây cú pháp trừu tượng

- Cây cú pháp trừu tượng (abstract syntax tree) loại bỏ các thông tin không cần thiết của cây phân tích
  - Minh họa quá trình nhóm các kí hiệu với nhau
  - Thích hợp với việc thực hiện tính toán và tổ hợp thông tin



Suy dẫn vs các cấu trúc cây

- Suy dẫn là cách biểu diễn thông tin 1 chiều
- Cấu trúc cây là cách biểu diễn thông tin 2 chiều
- Cấu trúc cây minh họa tương quan giữa các thành phần trong một cấu trúc không gian
- Cây phân tích mô tả đầy đủ nhất việc biến đổi từ kí hiệu gốc thành chuỗi cần phân tích, phù hợp nhất cho mọi mục đích sử dụng
- Cây cú pháp gạt bỏ các thành phần trung gian, tập trung mô tả tương quan giữa các kí hiệu kết thúc, cấu trúc này phù hợp với việc tổ hợp thông tin

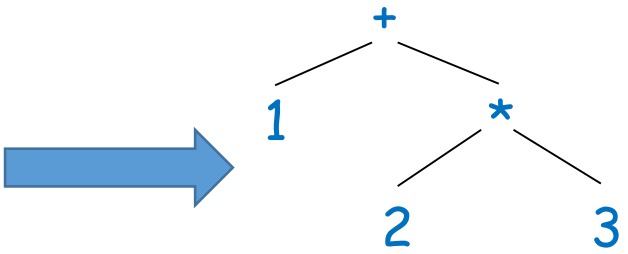
Phần 6

Văn phạm có nhập nhằng

Văn phạm có nhập nhằng

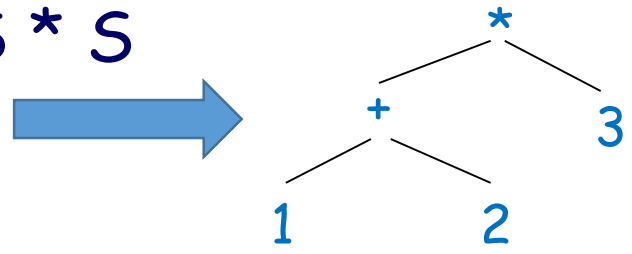
- Một văn phạm thiếu chặt chẽ dẫn tới việc có nhiều cây phân tích khác nhau với một chuỗi đầu vào
- Ví dụ văn phạm: S → S + S | S * S | number
- Phân tích xâu vào: 1 + 2 * 3
- Kết quả 1:

  S ⇒ S + S ⇒ 1 + S ⇒ 1 + S * S
  ⇒ 1 + 2 * S ⇒ 1 + 2 * 3



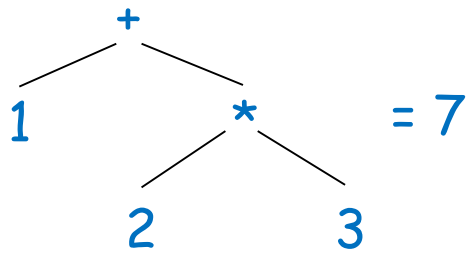
- Kết quả 2:

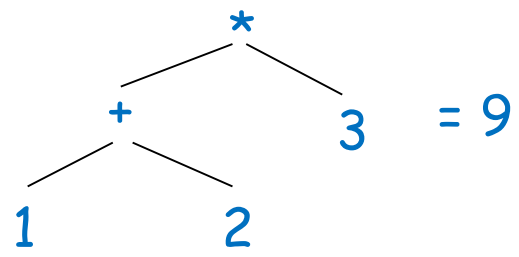
  S ⇒ S * S ⇒ S + S * S ⇒ 1 + S * S
  ⇒ 1 + 2 * S ⇒ 1 + 2 * 3



Văn phạm có nhập nhằng

- Văn phạm tồn tại ít nhất một chuỗi w có từ 2 cây phân tích tương ứng trở lên gọi là văn phạm có nhập nhằng
- Vấn đề lớn nhất của văn phạm có nhập nhằng là tính đa nghĩa của chuỗi w (có nhiều cách hiểu), hệ quả là không thể tính chính xác ngữ nghĩa của w





Khử nhập nhằng

- Việc khử nhập nhằng thực ra tạo một văn phạm mới dựa trên văn phạm cũ nhưng hai văn phạm không hoàn toàn tương đương
- Có nhiều chiến lược khử nhập nhằng
  - Thêm vào các biến trung gian
  - Đưa ra các ràng buộc ngoài văn phạm (ví dụ như quy định mức độ ưu tiên của các phép toán)
- Ví dụ văn phạm: S → S + S | S * S | number
- Khử nhập nhằng bằng cách thêm biến trung gian:

  S → S + T | T

  T → T * number | number

Phần 7

Các chiến lược phân tích cú pháp

Các chiến lược phân tích cú pháp

- Chiến lược phân tích cú pháp chia thành 3 nhóm:
  - Chiến lược thử-sai (quay lui): top-down, bottom-up
  - Chiến lược quy hoạch động: CYK, Earley,…
  - Chiến lược tất định (deterministic): LL, LR,…
- Ngoài ra còn có một số phương pháp khác dựa trên đặc điểm của ngôn ngữ để áp dụng các kĩ thuật hiệu quả (ví dụ như phân tích theo thứ bậc toán tử)
- Một số phương pháp tổng quát (như Earley chẳng hạn), nhưng đa số các phương pháp chỉ làm việc với những văn phạm có đặc thù riêng

Chiến lược thử-sai

- Chiến lược thử-sai hay là quay lui được nghĩ tới đầu tiên khi giải quyết bài toán phân tích văn phạm
- Các chiến lược này đơn giản là thử áp dụng các luật suy dẫn cho tới khi đạt được chuỗi suy dẫn mục tiêu
- Chia thành 2 cách tiếp cận ngược nhau:
  - Phương pháp top-down:
    - Nhìn từ cây phân tích thì đi từ trên xuống
    - Cố gắng từ S biến đổi dần ra w
  - Phương pháp bottom-up:
    - Nhìn từ cây phân tích thì đi từ dưới lên
    - Cố gắng thu gọn từ w về S

Chiến lược thử-sai

- Cả hai phương pháp này đều có hạn chế về văn phạm đầu vào:
  - Top-down yêu cầu văn phạm đầu vào không đệ quy trái
  - Bottom-up yêu cầu văn phạm đầu vào không có sản xuất rỗng ($A \rightarrow \varepsilon$) và không có suy dẫn dạng $A \Rightarrow^+ A$
- Các chiến lược này chỉ có ý nghĩa về mặt lý thuyết vì chậm và hạn chế văn phạm, tuy nhiên quá trình thử-sai đem lại nhiều gợi ý cho các thuật toán khác
  - Loại bỏ hạn chế văn phạm: các phương pháp vạn năng
  - Loại bỏ sự quay lui: các phương pháp tất định

Chiến lược quy hoạch động

- Ý tưởng quy hoạch động nhắm tới mục tiêu:
  - Xây dựng các phương pháp không có hạn chế về văn phạm đầu vào
  - Lưu trữ lại các chuỗi con đã phân tích để tránh phải quay lui
- Thuật toán CYK:
  - Cần biến đổi văn phạm về dạng chuẩn Chomsky
  - Văn phạm không có suy dẫn rỗng
  - Không quay lui
- Thuật toán Earley: vạn năng hơn, không có ràng buộc về văn phạm, không quay lui

Chiến lược tất định

- Ý tưởng tất định đi theo lựa chọn khác: hi sinh sự phong phú của văn phạm để đổi lấy tốc độ
- Đặc điểm chung:
  - Các văn phạm có sự ràng buộc nhất định
  - Dựa trên phân tích trước văn phạm để tiên đoán các tình huống có thể xảy ra
  - Xây dựng các bảng phương án, trong đó chỉ ra việc cần thực hiện khi gặp các tình huống cụ thể
- Đây là chiến lược mà tất cả các chương trình dịch đều sử dụng do ưu thế về tốc độ (không quay lui)

Phần 8

Thảo luận và bài tập

Thảo luận

1. Có thể sử dụng biểu thức chính quy và automat hữu hạn để thực hiện việc mô tả ngôn ngữ và dựng cây phân tích được hay không?
   - Chỉ ra một vấn đề đơn giản nhất thường gặp trong các ngôn ngữ lập trình và không thể giải quyết bằng automat hữu hạn
2. Lựa chọn cấu trúc dữ liệu phù hợp cho:
   - Kết quả trả về của bộ PTCP (cây phân tích)
   - Lưu trữ các luật văn phạm
   - Cấu trúc của nút lá
3. Thiết kế prototype cho máy PTCP
   - Viết mã minh họa hoạt động của prototype trên

Bài tập

1. Xác định ngôn ngữ được sinh bởi văn phạm:
   1. $S \rightarrow S\ (\ S\ )\ S \mid \varepsilon$
   2. $S \rightarrow a\ S\ b \mid b\ S\ a \mid \varepsilon$
   3. $S \rightarrow +\ S\ S \mid *\ S\ S \mid a$
   4. $S \rightarrow 0\ S\ 1 \mid \varepsilon$
2. Xây dựng văn phạm sản sinh ra ngôn ngữ:
   1. Số nhị phân lẻ
   2. Số nguyên không dấu
   3. Số nguyên có dấu
   4. Số thực, số nguyên không và có dấu
   5. Các từ đơn tiếng Việt

Bài tập

3. Khử nhập nhằng của văn phạm mô tả biểu thức số học dưới đây để phép nhân và chia được ưu tiên hơn phép cộng và trừ

   S → S A S | (S) | - S | số

   A → + | - | * | /

4. Xét văn phạm: S → a S b S | b S a S | ε

   a) Chứng minh văn phạm này nhập nhằng bằng cách xây dựng hai suy dẫn trái cho chuỗi w = abab

   b) Xây dựng các suy dẫn phải tương ứng cho abab

   c) Xây dựng các cây phân tích tương ứng cho abab

   d) *Văn phạm này sinh ra ngôn ngữ nào?

Bài tập

5. Cho văn phạm G: $S \rightarrow S\ a\ S \mid b$
   a) Ngôn ngữ được sinh bởi văn phạm G có đặc điểm gì?
   b) Hãy chỉ ra mọi phương án suy dẫn từ $S \Rightarrow^*$ babab
   c) *Hãy chỉ ra công thức tổng quát tính số lượng suy dẫn từ S thành chuỗi (ba)*b

6. Cho văn phạm G sau mô tả các biểu thức logic:

   $B \rightarrow false \mid true \mid \neg B \mid (B) \mid B \wedge B \mid B \vee B$

   a) Hãy viết lại văn phạm G để tránh nhập nhằng và thỏa mãn: mức độ ưu tiên cao nhất là phép Đảo (¬), tiếp theo là phép Và (∧), cuối cùng là phép Hoặc (∨)
   b) Hãy tạo văn phạm X là con của G chỉ gồm các biểu thức có giá trị true

Bài tập

7. Xét văn phạm:

S → (L) | a

L → L , S | S

a) Tìm các ký hiệu kết thúc, không kết thúc

b) Tìm các cây phân tích cú pháp cho các chuỗi sau

   i. (a, a)

   ii. (a, (a, a))

   iii. (a, ((a, a), (a, a)))

c) Xây dựng dẫn xuất trái cho mỗi câu trong b)

d) Xây dựng một dẫn xuất phải cho mỗi câu trong b)

e) *Văn phạm này sinh ra ngôn ngữ nào?

Bài tập

8. Cho văn phạm G sau:

Exp → Exp + Exp
Exp → Exp / Exp
Exp → số
Exp → (Exp)

Hãy chỉ ra cây phân tích cú pháp của các biểu thức:

a) 3 / (2 + 1)
b) (4 + 5) / (2 + 3)
c) 3 / 2 + 1
d) 1 + 2 + 3
e) 1 + 2 / 3

Bài tập

9. Cho biểu thức số học gồm các số nguyên, phép toán và các cặp ngoặc. Ngôn ngữ L được xây dựng bằng cách xóa các thành phần của biểu thức chỉ giữ lại các ngoặc (đóng ngoặc và mở ngoặc)
   - Hãy xây dựng văn phạm mô tả ngôn ngữ L
   - Ngôn ngữ L có những đặc điểm gì?
   - *Có bao nhiều chuỗi w có độ dài 20 thuộc L?
10. Viết bộ luật văn phạm câu tiếng Việt mô tả đồ vật
   - Vẽ cây cú pháp của câu “quyển vở đó màu xanh nhạt”
   - Vẽ cây cú pháp của câu “quyển sách to đó màu xanh”
   - Vẽ cây cú pháp của câu “quyển sách đó màu xanh to”