



CHƯƠNG TRÌNH DỊCH

Bài 7: Các chiến lược phân tích cú pháp



Nội dung

1. Suy dẫn
2. Biểu diễn suy dẫn bằng cấu trúc cây
3. Văn phạm có nhập nhằng
4. Các chiến lược phân tích cú pháp
 - Chiến lược thử-sai (quay lui): top-down, bottom-up
 - Chiến lược quy hoạch động: CYK, Earley,...
 - Chiến lược tất định (deterministic): LL, LR,...



Phần 1

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 $\alpha^1 \Rightarrow \alpha^2 \Rightarrow \dots \Rightarrow \alpha^n$ 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:

$$S \rightarrow E + S \mid E$$

$$E \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid (S)$$

- 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:

$$\begin{aligned} S &\Rightarrow E + S \Rightarrow E + E \Rightarrow E + 5 \Rightarrow (S) + 5 \Rightarrow (E + S) + 5 \\ &\Rightarrow (E + E + S) + 5 \Rightarrow (E + E + E) + 5 \\ &\Rightarrow (E + E + (S)) + 5 \Rightarrow (E + E + (E + S)) + 5 \\ &\Rightarrow (E + E + (E + E)) + 5 \Rightarrow (E + E + (E + 4)) + 5 \\ &\Rightarrow (E + E + (3 + 4)) + 5 \Rightarrow (E + 2 + (3 + 4)) + 5 \\ &\Rightarrow (1 + 2 + (3 + 4)) + 5 \end{aligned}$$

- *Câu hỏi:* qua các ví dụ về quá trình biến đổi từ S thành W , 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?



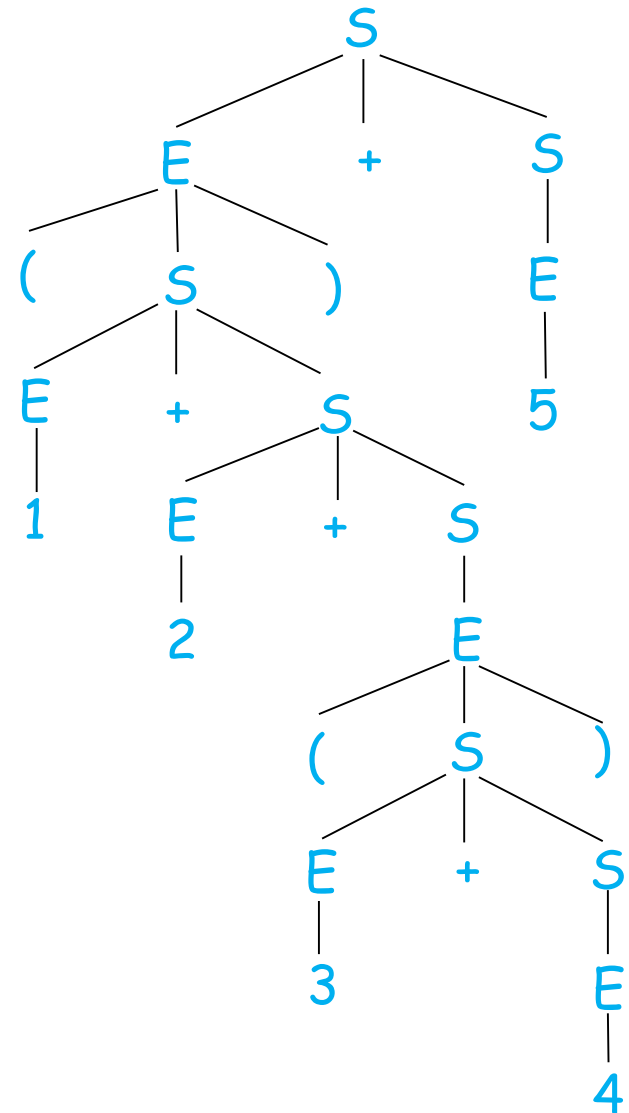
Phần 2

Biểu diễn suy diễn bằng cấu trúc cây



Cây phân tích (parse tree)

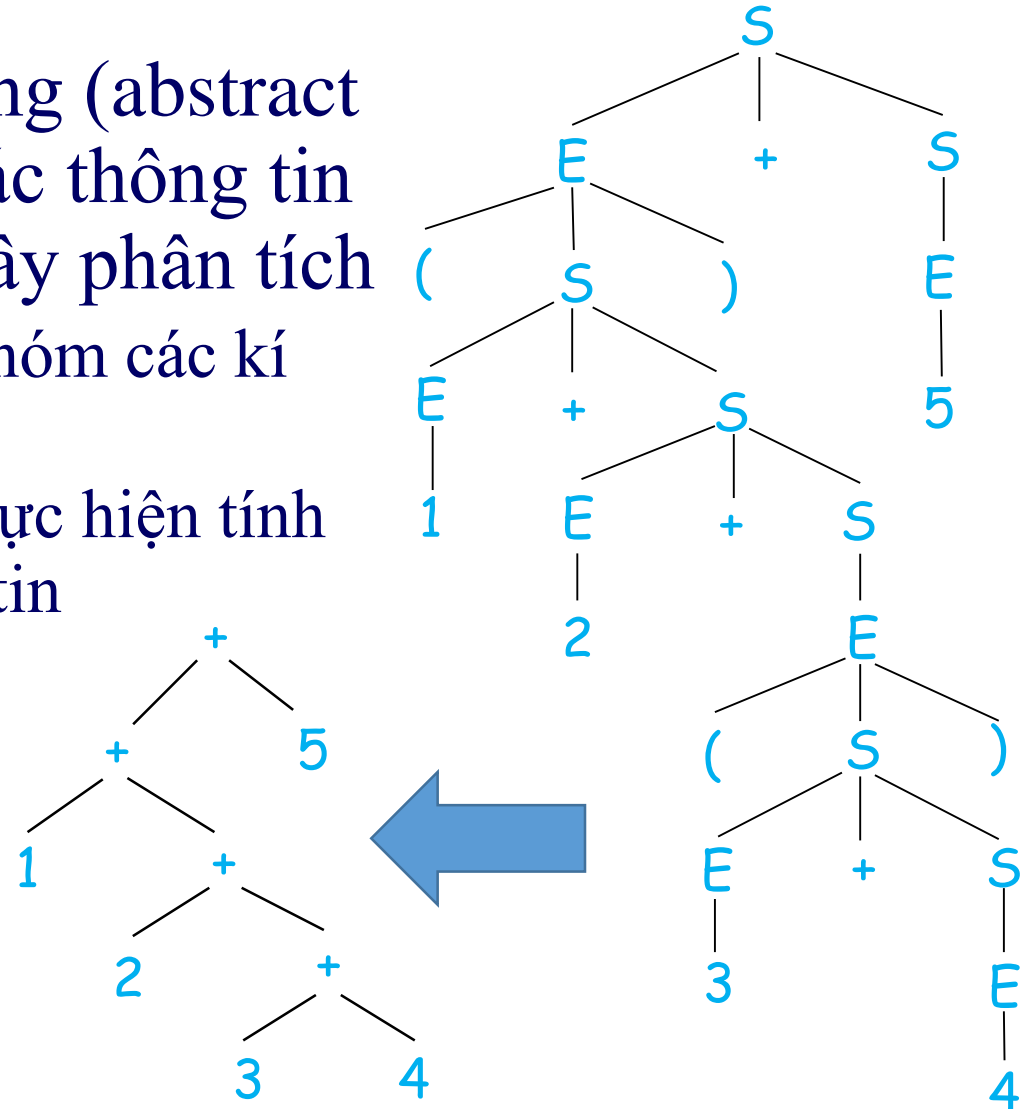
- Cây phân tích thể hiện cấu trúc của một suy dẫn
 - Nút gốc là kí hiệu bắt đầu
 - Các nút lá luôn là kí hiệu kết thúc
 - Các nút trong luôn là các kí hiệu trung gian
 - Cây không thể hiện thứ tự thực hiện các suy dẫn trực tiếp
 - Việc duyệt cây sẽ tạo thành thứ tự thực hiện suy dẫn
 - Suy dẫn trái tương 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 3

Văn phạm có nhập nhằng

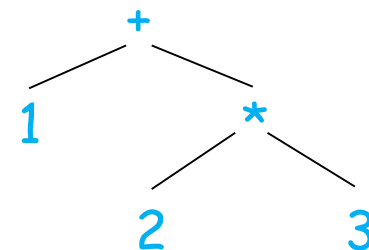


Văn phạm có nhập nhằng

- 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: $S \rightarrow S + S \mid S * S \mid \text{number}$
- Phân tích xâu vào: $1 + 2 * 3$

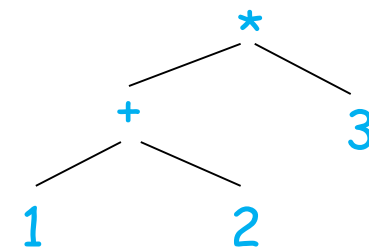
- Kết quả 1:

$$\begin{aligned} S &\Rightarrow S + S \Rightarrow 1 + S \Rightarrow 1 + S * S \\ &\Rightarrow 1 + 2 * S \Rightarrow 1 + 2 * 3 \end{aligned}$$



- Kết quả 2:

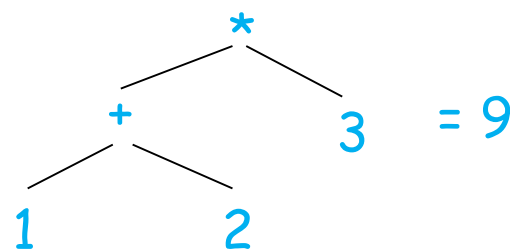
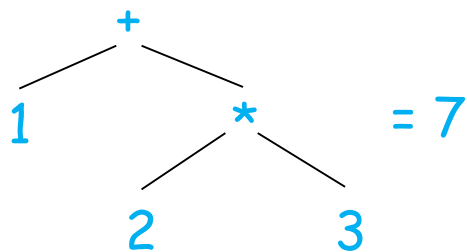
$$\begin{aligned} S &\Rightarrow S * S \Rightarrow S + S * S \Rightarrow 1 + S * S \\ &\Rightarrow 1 + 2 * S \Rightarrow 1 + 2 * 3 \end{aligned}$$





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: $S \rightarrow S + S \mid S * S \mid \text{number}$
- Khử nhập nhằng bằng cách thêm biến trung gian:
 $S \rightarrow S + T \mid T$
 $T \rightarrow T * \text{number} \mid \text{number}$



Phần 4

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 kỹ 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 vạm 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: vạm vỡ 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)