PHENIKAA UNIVERSITY
Realizing your potential

CHƯƠNG TRÌNH DỊCH

Chương 4. Phân tích cú pháp
Thuật toán CYK

TS. Phạm Văn Cảnh
Khoa Công nghệ thông tin

Email: canh.phamvan@phenikaa-uni.edu.vn

Nội dung

1. Khắc phục hạn chế của các phương pháp thử-sai
2. Các phương pháp phân tích cú pháp vạn năng
3. Áp dụng quy hoạch động vào phân tích cú pháp
4. Thuật toán Cocke – Younger – Kasami (CYK)
   - ❑ Dạng chuẩn Chomsky (CNF)
   - ❑ Ý tưởng
   - ❑ Mã minh họa
   - ❑ Đánh giá thuật toán
5. Bài tập

1. Khắc phục hạn chế của các PP thử-sai
Hạn chế của PP thử-sai

PHENIKAA UNIVERSITY
Realizing your potential

- Hai thuật toán thử-sai cơ bản top-down và bottom-up đều có những hạn chế về văn phạm đầu vào
  - Top-down: văn phạm không có đệ quy trái
  - Bottom-up: văn phạm không có suy dẫn rỗng và không có kí hiệu đệ quy ($A \Rightarrow^{+} A$)
- Các thuật toán thử-sai có hạn chế về mặt tốc độ
  - Tốc độ chấp nhận được với một số văn phạm đơn giản và đơn nghĩa, đầu vào ngắn
  - Trường hợp xấu có độ phức tạp tính toán hàm mũ
- Không có cơ chế hiệu quả loại bỏ sự trùng lặp về kết quả (chẳng hạn như nhiều suy dẫn tương đương)

1. Khắc phục hạn chế của các PP thử-sai
Hạn chế của PP thử-sai

- Nguyên nhân của những hạn chế này
  - Hạn chế do bản thân cơ chế hoạt động của thử-sai
  - Không có cơ chế loại bỏ các phương án chắc-chắn-sai
- Ví dụ: quá trình suy dẫn S thành w = abcdefg

$$S \Rightarrow \ldots \Rightarrow abcAx \Rightarrow \ldots \Rightarrow abcdefg$$

- Ta nhận thấy phương án có chuỗi trung gian abcAx hoàn toàn không thể đạt được chuỗi w mong muốn
  - Vì x là kí hiệu không kết thúc, nó luôn luôn tồn tại trong các suy dẫn tiếp theo, trong khi chuỗi w không chứa x
- **Câu hỏi:** thuật toán thử sai tốt ~ cắt nhánh sớm?

2. Các phương pháp PTCP vạn năng

- Như vậy các thuật toán thử-sai có 2 điểm yếu
  - Hệ luật văn phạm bị hạn chế
  - Yêu cầu nhiều thời gian tính toán
- Vì vậy chúng ta cũng có 2 mục tiêu
  - Tạo ra thuật toán phân tích vạn năng (không bị hạn chế bởi luật văn phạm)
  - Tạo ra thuật toán phân tích tốc độ cao
- Tất nhiên nếu có thuật toán đạt được cả 2 mục tiêu trên thì quá tốt
- Trong phần này ta nhắm tới mục tiêu thứ nhất

2. Các phương pháp PTCP vạn năng

PHENIKAA UNIVERSITY
Realizing your potential

- Có 2 chiến lược:
  - Biến đổi văn phạm G thành văn phạm G' tương đương nhưng không có những hạn chế của thuật toán
  - Thay đổi cơ chế của thuật toán, nói cách khác là không sử dụng cơ chế thử-sai hiện có
- Chiến lược thứ nhất không có lời giải trọn vẹn
  - Thuật toán khử đệ quy trái có thể thay đổi ý nghĩa của văn phạm, kết quả là văn phạm G' thực chất không hoàn toàn tương đương G
  - Khử suy dẫn rỗng hoặc kí hiệu đệ quy làm cho văn phạm khó hiểu, các kí hiệu trung gian mất ý nghĩa ban đầu của nó

3. Áp dụng quy hoạch động vào PTCP

PHENIKAA UNIVERSITY
Realizing your potential

- Quy hoạch động gồm hai ý tưởng cơ bản.
  - Chia bài toán lớn thành các bài toán con độc lập.
  - Sử dụng bộ nhớ để lưu trữ lại các lời giải của các bài toán con (để tránh việc phải giải nhiều lần một bài toán).
- Áp dụng vào bài toán phân tích văn phạm.
  - Cây phân tích $S \Rightarrow^* w$ thực chất gồm các cây con, mỗi cây con phân tích một chuỗi con liên tiếp trong w.
  - Sử dụng bộ nhớ để lưu trữ lại các kết quả suy dẫn ra các chuỗi con của w (có nhiều chiến lược, chẳng hạn như lưu trữ các chuỗi từ $w_i w_{i+1} \dots w_j$ hoặc chuỗi $w_0 w_1 \dots w_k$, tùy vào mục tiêu cần lưu trữ).

4. Thuật toán Cocke – Younger –Kasami (CYK)
Dạng chuẩn Chomsky (CNF)

- Văn phạm phi ngữ cảnh ở dạng chuẩn Chomsky nếu mọi luật sinh đều có dạng A → BC hoặc A → a
- Dễ thấy mọi văn phạm phi ngữ cảnh không chứa suy dẫn rỗng (A → ϵ) đều có thể chuyển về dạng chuẩn Chomsky bằng thuật toán đơn giản sau
  - Nếu luật sinh sẵn ở dạng chuẩn Chomsky thì giữ nguyên
  - Nếu luật sinh không ở dạng chuẩn Chomsky thì sẽ có dạng

$$A \rightarrow B_1 B_2 \dots B_n, \text{ với n} > 2$$

    - Ta bổ sung các kí hiệu trung gian mới $C_1, C_2, \dots, C_{n-2}$
    - Thay thế luật trên bằng các luật mới

$$A \rightarrow C_1 B_n, C_1 \rightarrow C_2 B_{n-1}, \dots C_{n-2} \rightarrow B_1 B_2,$$

    các luật mới này thỏa mãn chuẩn Chomsky

Thuật toán CYK

- CYK không phải là thuật toán vạn năng vì không chấp nhận văn phạm có suy dẫn rỗng.
- CYK minh họa một cách đơn giản ý tưởng quy hoạch động:
  - Giả thiết chuỗi $w = w_1 w_2 \dots w_n$.
  - Ta định nghĩa tập X_{ij} là tập tất cả các kí hiệu có thể suy dẫn ra chuỗi con $w_i w_{i+1} \dots w_{i+j-1}$ (chuỗi con bắt đầu từ w_i và có độ dài j).
  - Bài toán đoán nhận $S \Rightarrow^* w$ tương đương với việc trả lời S có thuộc tập X_{1n} hay không?
  - Vấn đề bây giờ là tính X_{ij} như thế nào?

Thuật toán CYK - Chương trình

```
// tính X của các chuỗi độ dài 1
for (int i = 1; i <= n; i++)
    X[i,1] = { A | A → w_i }

// tính X của các chuỗi độ dài 2,3,…,n
for (int j = 2; j <= n; j++)
    for (int i = 1; i <= n-j+1; i++) {
        X[i,j] = {}
        for (int k = 1; k <= j-1; k++)
            X[i,j] += { A | nếu A → BC
                        | và B thuộc X[i,k]
                        | và C thuộc X[i+k,j-k] }
    }
```

Thuật toán CYK – Ví dụ

Văn phạm:

$S \rightarrow AB \mid BC$

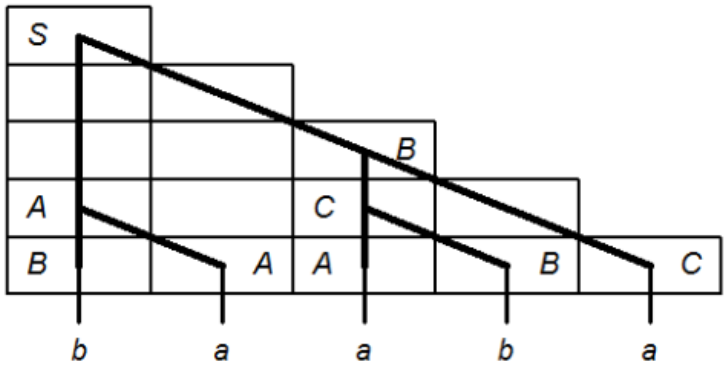
$A \rightarrow BA \mid a$

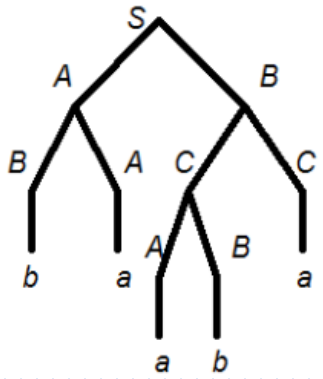
$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

Chuỗi w = baaba

j \ i	1	2	3	4	5
5	S, A, C				
4	∅	S, A, C			
3	∅	B	B		
2	S, A	B	S, C	S, A	
1	B	A, C	A, C	B	A, C
	b	a	a	b	a





Thuật toán CYK

- Hạn chế:
  - Thuật toán không làm việc với suy dẫn rỗng
  - Số lượng kí hiệu trung gian (non-terminal) nhiều, do việc chuyển đổi từ CFG sang chuẩn Chomsky
- Độ phức tạp tính toán (xấu nhất) là $O(n^3|G|)$
  - Số n là độ dài của chuỗi w
  - |G| là kích thước của văn phạm dạng CNF
- Bản chất là ý tưởng bottom-up nhưng áp dụng các kĩ thuật quy hoạch động
- Dễ dàng liệt kê mọi cây phân tích khác nhau và loại bỏ các suy dẫn trùng lặp

Bài tập

1. Cho văn phạm G:

S → AB | XB

T → AB | XB

X → AT

A → a

B → b

Chỉ ra quá trình thực hiện thuật toán CYK với w = aaabbb

2. Cho văn phạm G:

S→AA | AS | b

A→SA | AS | a

Chỉ ra quá trình thực hiện thuật toán CYK với w = abaab

Bài tập

3. Sử dụng thuật toán CYK để chỉ ra cây phân tích cho chuỗi **(5+7)*3** thuộc văn phạm G

E ⟶ E + T | T

T ⟶ T * F | F

F ⟶ (E) | số

4. Chỉ ra cây phân tích của chuỗi **true and not false** sinh bởi thuật toán CYK với tập luật văn phạm G

E ⟶ E and T | T

T ⟶ T or F | F

F ⟶ not F | (E) | true | false