CHƯƠNG TRÌNH DỊCH

Bài 13: Phân tích cú pháp bằng thuật toán LL

Nội dung

1. Giới thiệu về bộ phân tích cú pháp tất định
2. Tiếp cận top-down
3. Phân tích LL(1)
   - FIRST
   - FOLLOW
   - Bảng phân tích LL(1)
   - Ví dụ
4. Bài tập

Phần 1

Giới thiệu về bộ phân tích cú pháp tất định

Ràng buộc về thời gian tính toán

- Các thuật toán phân tích vạn năng (CYK, Earley)
  - Phân tích mọi văn phạm phi ngữ cảnh
  - Tốc độ chấp nhận được: $O(n^3)$ với n là độ dài chuỗi vào
- Đối với những mã nguồn các ngôn ngữ lập trình, giá trị của n có thể lên tới vài triệu, bài toán phân tích văn phạm trở nên rất đặc biệt
  - Tốc độ chấp nhận được nếu là gần tuyến tính O(n)
  - Văn phạm đơn giản, chặt chẽ, đơn nghĩa
- Hệ quả là nảy sinh nhu cầu xây dựng các bộ phân tích văn phạm tất định (deterministic)

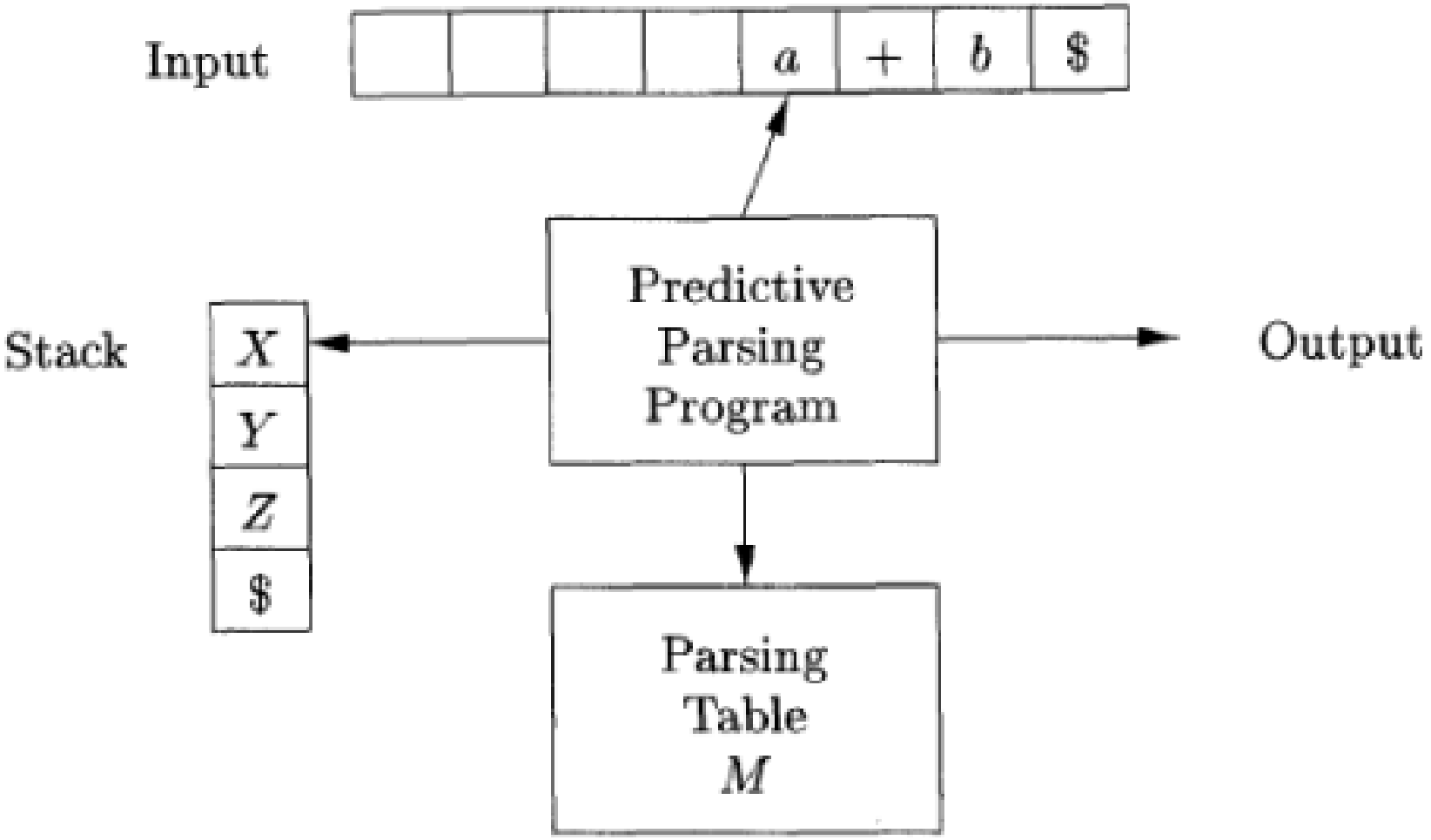
Chiến lược tất định

- Thế nào là "tất định" – do ràng buộc độ phức tạp tính toán là O(n), hệ quả là:
  - Khi nhận một kí hiệu đầu vào, bộ phân tích văn phạm cần ngay lập tức quyết định sẽ sử dụng luật sinh nào cho trường hợp này
  - Quyết định chọn luật sinh nào cần phải đủ tốt để không phải thử lại phương án khác
  - Tính chất "tất định" ~ không có quay lui
- Cái giá phải trả cho sự "tất định":
  - Văn phạm sẽ không còn vạn năng nữa
  - Nhưng văn phạm đủ tốt để dùng trong thực tế

Kiến trúc chung: bảng phương án

- Việc lựa chọn ngay lập tức phương án suy dẫn dẫn tới yêu cầu cần nghiên cứu trước bộ luật văn phạm và có các phương án phù hợp trong các tình huống có thể xảy ra
- Các thuật toán phân tích tất định đều sử dụng kĩ thuật xây dựng trước bảng phương án
- Có nhiều kĩ thuật xây dựng bảng phương án khác nhau ứng với các phương pháp tiếp cận khác nhau
- Với các loại bảng phương án, thuật toán phân tích cũng có sự khác biệt khi thực hiện đoán nhận

Kiến trúc chung: bảng phương án



Phần 2

Tiếp cận top-down

Tiếp cận top-down

- Hãy quan sát quá trình thực hiện phân tích top-down chuỗi w = ( ) ( ) của văn phạm:

  S → (S) S | ε

  - Cần tìm quá trình suy dẫn S ⇒* w = ( ) ( )
  - Ở đây chúng ta chỉ có 1 non-terminal duy nhất S
  - Có 2 terminal "(" và ")"
- Bước suy dẫn đầu tiên, S ⇒ (S) S ⇒* ( ) ( )
- Vậy ở bước 2, cần tìm quá trình S) S ⇒*) ( )
- Rõ ràng trong tình huống này, ta không thể áp dụng luật sinh S → (S) S mà phải sử dụng S → ε

Tiếp cận top-down

Quan sát quá trình suy dẫn từ $\alpha \Rightarrow^* w$, dễ thấy:

- **Nếu α bắt đầu bởi terminal**, thì terminal đó nhất thiết phải trùng với kí hiệu bắt đầu của w, trong tình huống này ta gạt bỏ kí hiệu này ở cả 2 chuỗi
- **Nếu α bắt đầu bởi non-terminal A**, thì A nhất thiết phải suy dẫn (trực tiếp hoặc gián tiếp) ra kí hiệu bắt đầu của w (w_1) hoặc ra ε
- Ta có thể dựa trên văn phạm G để tính được A có suy ra w_1 được hay không?
- Lập một bảng phương án 2 chiều, 1 chiều gồm các non-terminal, 1 chiều gồm các terminal, ta đưa ra các tình huống áp dụng luật sinh cho mỗi cặp (A, w_1)

Phần 3

Phân tích LL(1)

Phân tích LL(1)

Bước	Chuỗi nguồn	Chuỗi đích	Hành động
1	S$	()()$	S → (S) S
2	(S)S$	()()$	gạt bỏ
3	S)S$	)()$	S → ε
4	)S$	)()$	gạt bỏ
5	S$	()$	S → (S) S
6	(S)S$	()$	gạt bỏ
7	S)S$	)$	S → ε
8	)S$	)$	gạt bỏ
9	S$	$	S → ε

M[N, T]	(	)	$
S	$S \rightarrow (S)S$	$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

Phân tích LL(1)

- Như vậy bộ phân tích LL(1) hoạt động tương tự như phân tích top-down, nhưng không có bước quay lui (vì không có sự lựa chọn thử-sai)
- Vấn đề lớn nhất: làm sao xây dựng được bảng phương án?
- LL(1) nghĩa là gì? Viết tắt của "Left-to-right parse, Leftmost-derivation, 1-symbol lockahead"
- Kí hiệu k trong LL(k) nghĩa là bộ phân tích sẽ nhìn trước k kí hiệu khi ra quyết định

FIRST(X)

- Nếu X là kí hiệu kết thúc thì FIRST(X) là {X}
- Nếu $X \rightarrow \varepsilon$ là một luật sinh thì thêm ε vào FIRST(X)
- Nếu $X \rightarrow Y_1Y_2Y_3...Y_k$ là một luật sinh thì:
  - Thêm tất cả các ký hiệu kết thúc khác ε của FIRST(Y_1) vào FIRST(X)
  - Nếu $\varepsilon \in$ FIRST(Y_1) thì tiếp tục thêm vào FIRST(X) tất cả các ký hiệu kết thúc khác ε của FIRST(Y_2)
  - Nếu $\varepsilon \in$ FIRST(Y_1) $\cap$ FIRST(Y_2) thì thêm tất cả các ký hiệu kết thúc khác $\varepsilon \in$ FIRST(Y_3)
  - Tiếp tục như vậy cho tới Y_k
  - Thêm ε vào FIRST(X) nếu $\varepsilon \in \cap^{i=1 \rightarrow k}$ FIRST(Y_i)

FIRST(α)

Định nghĩa FIRST(α): giả sử α là một chuỗi các ký hiệu văn phạm, FIRST(α) là tập hợp các ký hiệu kết thúc mà nó bắt đầu một chuỗi dẫn xuất từ α

- Giả sử $\alpha = X_1X_2\ldots X_n$
- Thêm vào FIRST(α): FIRST(X_1)-{ε}
- Với mọi i=2,3,…,n; nếu FIRST(X_k) chứa ε với mọi k=1,2…,i-1 thì thêm vào FIRST(α): FIRST(X_i)-{ε}
- Nếu với mọi i=1,2,…n; nếu FIRST(X_i) chứa ε thì thêm ε vào FIRST(α)

Tính FIRST: ví dụ

Xét văn phạm G:

$E \rightarrow T\ E'$

$E' \rightarrow +\ T\ E' \mid \varepsilon$

$T \rightarrow F\ T'$

$T' \rightarrow *\ F\ T' \mid \varepsilon$

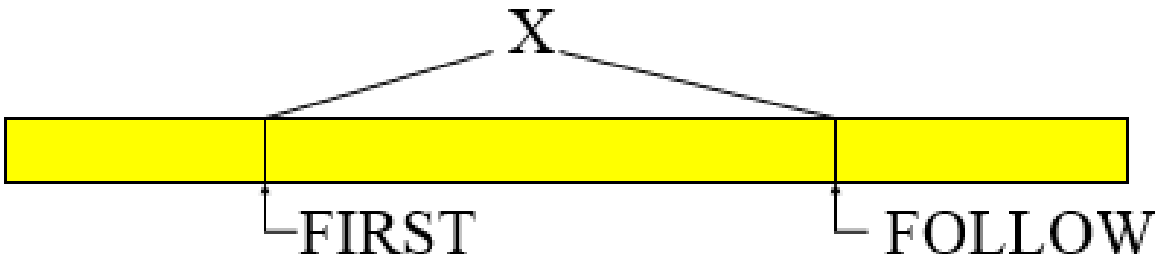
$F \rightarrow (E) \mid id$

- FIRST(E) = FIRST(T) = FIRST(F) = { (, id }
- FIRST(E') = { +, ε }
- FIRST(T') = { *, ε }

FOLLOW

Định nghĩa FOLLOW(A): tập hợp các ký hiệu kết thúc a mà nó xuất hiện ngay sau A (bên phải của A) trong một dạng câu nào đó

- Tức là tập hợp các ký hiệu kết thúc a, sao cho tồn tại một dẫn xuất dạng $S \Rightarrow^* \alpha A a \beta$
- Chú ý rằng nếu A là ký hiệu phải nhất trong một dạng câu nào đó thì $\$ \in FOLLOW(A)$ ($ là ký hiệu kết thúc chuỗi nhập)



Tính FOLLOW

Tính FOLLOW (A): áp dụng các quy tắc sau cho đến khi không thể thêm gì vào mọi tập FOLLOW được nữa

- Đặt $ vào follow(S), trong đó S là ký hiệu bắt đầu của văn phạm và $ là ký hiệu kết thúc chuỗi nhập
- Nếu có một luật sinh A→ αBβ thì thêm mọi phần tử khác ε của FIRST(β)vào trong FOLLOW(B)
- Nếu có luật sinh A→ αB hoặc A→ αBβ mà ε ∈ FIRST(β) thì thêm tất cả các phần tử trong FOLLOW(A) vào FOLLOW(B)

Tính FOLLOW: ví dụ

Xét văn phạm G:

E → T E'

E' → + T E' | ε

T → F T'

T' → * F T' | ε

F → (E) | id

- FOLLOW(E) = FOLLOW(E') = { $,) }
- FOLLOW(T) = FOLLOW(T') = { +,), $ }
- FOLLOW(F) = {*,+,), $ }

Bảng phân tích LL(1)

1. Với mỗi luật sinh A→ α của văn phạm, thực hiện:
   1. Với mỗi ký hiệu kết thúc a ∈ FIRST(α), thêm A→ α vào M[A,a]
   2. Nếu ε ∈ FIRST(α) thì đưa luật sinh A→ α vào M[A,b] với mỗi ký hiệu kết thúc b ∈ FOLLOW(A)
   3. Nếu ε ∈ FIRST(α) và $ ∈ FOLLOW(A) thì đưa luật sinh A→ α vào M[A,$]
2. Các ô trống trong bảng tương ứng với lỗi (error)

Chú ý: một ô trong bảng có thể chứa nhiều suy dẫn, tình huống này gọi là bảng có nhập nhằng

Ví dụ

Xét văn phạm G:

$E \rightarrow T\,E'$ $\quad E' \rightarrow +\,T\,E' \mid \varepsilon$

$T \rightarrow F\,T'$ $\quad T' \rightarrow *\,F\,T' \mid \varepsilon$ $\quad F \rightarrow (E) \mid id$

Ký hiệu chưa kết thúc	Ký hiệu nhập					
	id	**+**	*****	**(**	**)**	**$**
E	E → TE'			E → TE'		
E'		E → +TE'			E→ ε	E'→ ε
T	T → FT'			T → FT'		
T'		T'→ ε	T'→ *FT'		T'→ ε	T'→ ε
F	F → id			F → (E)		

Phần 4

Bài tập

Bài tập

1. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

   $S \rightarrow Ac \mid BBc$ $\quad\quad$ $A \rightarrow BC$

   $C \rightarrow b \mid bCd$ $\quad\quad$ $B \rightarrow dBb \mid dDb \mid \varepsilon$

   $D \rightarrow bd \mid bDd$

2. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

   $S \rightarrow AD \mid abc$ $\quad\quad$ $A \rightarrow Bc$

   $B \rightarrow dBc \mid CC$ $\quad\quad$ $D \rightarrow Dd \mid \varepsilon$

   $C \rightarrow DCb \mid CDb \mid \varepsilon$

Bài tập

3. Tạo bảng phân tích LL(1) và chỉ ra quá trình phân tích chuỗi w = (a+a) với văn phạm G sau đây:

   S → F S → (S+F) F → a

4. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

   S → A A → T | A+T T → b | (A)

5. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

   E → E or T | T

   T → T and F | F

   F → not F | (E) | x