

– INF01147 –
Compiladores

Análise Sintática
Conjuntos Primeiro e Sequência

Prof. Lucas M. Schnorr
– Universidade Federal do Rio Grande do Sul –



Revisão

- ▶ Os três tipos de analisadores descendentes
 - ▶ Recursivo com retrocesso
 - ▶ Recursivo preditivo
 - ▶ Tabular preditivo
 - ▶ **Qual a diferença de funcionamento entre eles?**
- ▶ Qual a vantagem do recursivo com retrocesso?

Plano da Aula de Hoje

- ▶ Conjuntos *Primeiro* e *Sequência*
- ▶ Estratégia **Descendente** Recursiva Preditiva
 - ▶ Algumas questões de implementação

Conjunto Primeiro

- ▶ Definição do conjunto Primeiro
- ▶ Seja α qualquer sequência de símbolos (terminais ou não)
- ▶ Primeiro (α)
 - ▶ Conjunto de todos os **terminais** que **começam** qualquer sequência derivável de α
 - ▶ Definição formal
 - ▶ Se $\alpha \Rightarrow^* \epsilon$, então ϵ é um elemento de Primeiro(α)
 - ▶ Se $\alpha \Rightarrow^* a\beta$, então **a** é um elemento de Primeiro(α) sendo **a** um terminal, e β uma forma sentencial ou vazio
- ▶ Exemplo
 - $A \rightarrow B \mid C \mid D$
 - $B \rightarrow b$
 - $C \rightarrow c$
 - $D \rightarrow d$
- ▶ **Em geral**, calcula-se somente o Primeiro dos não-terminais

Conjunto Primeiro

- ▶ Condição para usar $\text{Primeiro}(\alpha)$ em um preditivo
 - ▶ Supondo as regras de produção para o não-terminal A
 $A \rightarrow \beta_1 | \beta_2 | \beta_3 | \dots | \beta_n$
 - ▶ Os conjuntos Primeiro de cada uma dessas produções devem obrigatoriamente ser **disjuntos**, ou seja
 $\text{Primeiro}(\beta_1) \cap \text{Primeiro}(\beta_2) \cap \dots \cap \text{Primeiro}(\beta_n) = \emptyset$
- ▶ Pergunta
 - ▶ O que acontece se eles não forem disjuntos?
Exemplo: $B \rightarrow cbB \mid ca \mid bX$
 - ▶ Como resolver esse problema?

Algoritmo para Primeiro(X)

- ▶ Regras para calcular o Primeiro(X) de todos os símbolos X
→ X é qualquer terminal ou não-terminal
- ▶ Se X é terminal, então $\text{Primeiro}(X) = \{ X \}$
- ▶ Se X é não-terminal e $X \rightarrow Y_1 Y_2 \dots Y_k$ com $k \geq 1$
 - ▶ Acrescente a em Primeiro(X)
 - ▶ Se, para algum i , a estiver em Primeiro(Y_i) e
 - ▶ Se ϵ estiver em todos os Primeiro(Y_1) ... Primeiro(Y_{i-1})
 - ▶ Acrescente ϵ em Primeiro(X)
 - ▶ Somente se ϵ está em Primeiro(Y_j) para todo $j = 1, 2, \dots, k$
 - ▶ Se existe uma regra $X \rightarrow \epsilon$
 - ▶ Acrescente ϵ em Primeiro(X)
- ▶ Aplicar estas regras iterativamente
→ Até não existir mais terminais ou ϵ para adicionar

Exercício Primeiro.1

- Calcule o conjunto Primeiro de cada não-terminal

exp	→	exp soma termo
exp	→	termo
soma	→	+
soma	→	-
termo	→	termo mult fator
termo	→	fator
mult	→	*
fator	→	(exp)
fator	→	número

- Resposta

exp	(número
termo	(número
fator	(número
soma	+ -
mult	*

Exercício Primeiro.2

- Calcule o conjunto Primeiro de cada não-terminal

declaração	→	if-decl
declaração	→	outra
if-decl	→	if (exp) declaração else-parte
else-parte	→	else declaração
else-parte	→	ε
exp	→	0
exp	→	1

- Resposta

declaração	if outra
if-decl	if
else-parte	else ε
exp	0 1

Exercício Primeiro.3

- Calcule o conjunto Primeiro de cada não-terminal

decl-sequência	→	decl decl-seq'
decl-seq'	→	; decl-sequência
decl-seq'	→	ε
decl	→	s

- Resposta

decl-sequência	s
decl	s
decl-seq'	; ε

Usando Primeiro(X) (e motivação)

Para análise descendente recursiva preditiva

- Considerando a seguinte gramática

$S \rightarrow ACE$

$A \rightarrow a \mid b \mid \epsilon$

$C \rightarrow c \mid d \mid \epsilon$

$E \rightarrow e$

- Implementação possível para o não-terminal $A()$

```
A() {  
    switch {  
        case 'a': Consume('a'); break;  
        case 'b': Consume('b'); break;  
        case '???': ???  
        default: abortar ("Erro Sintático");  
    }  
}
```

- Pergunta

- O que fazer para detectar o ϵ ?
- Precisamos saber o que pode vir depois do não-terminal $A \rightarrow$
Precisamos calcular o conjunto **Sequência(A)**

Conjunto Sequência

Conjunto Sequência

- ▶ Definição do conjunto Sequência
- ▶ Seja A um não-terminal pertencente a gramática
- ▶ Sequência (A)
 - ▶ Conjunto de **terminais** que aparecem imediatamente **após** o não-terminal A em alguma forma sentencial válida
 - ▶ Definição formal
 - ▶ Conjunto de terminais **a**, tal que exista uma derivação $S \Rightarrow^* \alpha A a \beta$, para qualquer forma sentencial α e β
- ▶ Exemplo
$$S \rightarrow a \mid [L]$$
$$L \rightarrow S ; L \mid S$$

Algoritmo para Sequência(A)

- ▶ Regras para calcular o Sequência(A)
 - A é qualquer não-terminal
 - α e β a seguir são formas sentenciais válidas
- ▶ Se A é o símbolo inicial
 - ▶ Acrescente \$ ao conjunto Sequência(A)
- ▶ Se houver $X \rightarrow \alpha A \beta$
 - ▶ Acrescente Primeiro(β) exceto o ϵ ao conjunto Sequência(A)
(note que ϵ nunca fará parte de conjuntos Sequência)
- ▶ Se houver $X \rightarrow \alpha A$
 - ▶ Acrescente Sequência(X) ao conjunto Sequência(A)
- ▶ Se houver $X \rightarrow \alpha A \beta$ e Primeiro(β) contém ϵ
 - ▶ Acrescente Sequência(X) ao conjunto Sequência(A)
- ▶ Aplicar estas regras iterativamente
 - Até que nenhum conjunto Sequência aumente de tamanho

Exercício Sequência.1

- Calcule os conjuntos Primeiro e Sequência

exp \rightarrow exp soma termo
exp \rightarrow termo
soma \rightarrow +
soma \rightarrow -
termo \rightarrow termo mult fator
termo \rightarrow fator
mult \rightarrow *
fator \rightarrow (exp)
fator \rightarrow número

- Resposta

	Primeiro	Sequência
exp	(número	\$ + -)
termo	(número	\$ + - *)
fator	(número	\$ + - *)
soma	+ -	(número
mult	*	(número

Exercício Sequência.2

- Calcule os conjuntos Primeiro e Sequência

declaração \rightarrow if-decl
declaração \rightarrow **outra**
if-decl \rightarrow **if (exp)** declaração else-parte
else-parte \rightarrow **else** declaração
else-parte \rightarrow ϵ
exp \rightarrow **0**
exp \rightarrow **1**

- Resposta

	Primeiro	Sequência
declaração	if outra	\$ else
if-decl	if	\$ else
else-parte	else ϵ	\$ else
exp	0 1)

Exercício Sequência.3

- Calcule os conjuntos Primeiro e Sequência

decl-sequência	→	decl decl-seq'
decl-seq'	→	; decl-sequência
decl-seq'	→	€
decl	→	s

- Resposta

	Primeiro	Sequência
decl-sequência	s	\$
decl	s	;
decl-seq'	; €	\$

Aplicando o algoritmo para Sequência(A)

- Considerando a seguinte gramática

$S \rightarrow ACE$

$A \rightarrow a \mid b \mid \epsilon$

$C \rightarrow c \mid d \mid \epsilon$

$E \rightarrow e$

- Implementação possível para o não-terminal A()

```
A() {  
    switch {  
        //Tratando o Primeiro(A)  
        case 'a': Consome('a'); break;  
        case 'b': Consome('b'); break;  
  
        //Tratando o Sequência(A)  
        case 'c':  
        case 'd':  
        case 'e':  
            break;  
        default: abortar ("Erro Sintático");  
    }  
}
```

Exercício 1

- Calcule os conjuntos Primeiro e Sequência (apenas as letras minúsculas são terminais)

$$S \rightarrow AB$$

$$A \rightarrow c \mid \epsilon$$

$$B \rightarrow cbB \mid ca$$

- Respostas

$$\text{Primeiro}(A) = \{ c, \epsilon \}$$

$$\text{Primeiro}(B) = \{ c \}$$

$$\text{Primeiro}(S) = \{ c \}$$

$$\text{Sequência}(A) = \{ c \}$$

$$\text{Sequência}(B) = \{ \$ \}$$

$$\text{Sequência}(S) = \{ \$ \}$$

Exercício 2

- Calcule os conjuntos Primeiro e Sequência (apenas as letras minúsculas são terminais)

$$S \rightarrow XYZ$$

$$X \rightarrow aXb \mid \epsilon$$

$$Y \rightarrow cYZcX \mid d$$

$$Z \rightarrow eZYe \mid f$$

- Respostas

$$\text{Primeiro}(X) = \{ a, \epsilon \}$$

$$\text{Primeiro}(Y) = \{ c, d \}$$

$$\text{Primeiro}(Z) = \{ e, f \}$$

$$\text{Primeiro}(S) = \{ a, c, d \}$$

$$\text{Sequência}(X) = \{ c, d, b, e, f \}$$

$$\text{Sequência}(Y) = \{ e, f \}$$

$$\text{Sequência}(Z) = \{ \$, c, d \}$$

$$\text{Sequência}(S) = \{ \$ \}$$

Exercício 3

- Calcule os conjuntos Primeiro e Sequência (apenas as letras minúsculas são terminais)

$$\begin{aligned} T &\rightarrow TS \mid \wedge \text{idf} \mid \text{array} [TS] \text{ of } T \\ TS &\rightarrow \text{integer} \mid \text{char} \mid \text{num pp num} \end{aligned}$$

- Respostas
TBD

Conclusão

- ▶ Leituras Recomendadas
 - ▶ Livro do Dragão
 - ▶ Seções 2.4.2 e 4.4.2
 - ▶ Série Didática
 - ▶ Seção 3.2.2 (Primeiro) e parte da 3.2.3 (Sequência)
 - ▶ Dick Grune et. al.
 - ▶ Seção 2.2.4.1 e 2.2.4.2
- ▶ **Material Complementar**
 - ▶ Vídeos do Prof. Nicolas (sobre Primeiro e Sequência)
- ▶ Próxima Aula
 - ▶ Gramáticas LL(1) e Análise LL(1) com tabela preditiva