

Análise Sintática Descendente (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

- Os analisadores descendentes são ajudados por duas funções First e Follow associadas à gramática.
- Elas permitem escolher qual produção aplicar com base no próximo símbolo de entrada
- permite ainda a recuperação de erro (follow).

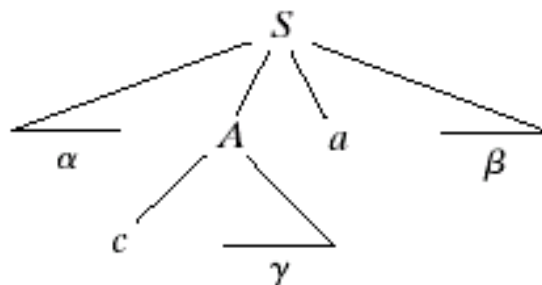


FIGURA 4.15 O símbolo terminal c está em $FIRST(A)$ e o símbolo terminal a está em $FOLLOW(A)$.

Análise Sintática Descendente (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

First(X)

Conjunto de terminais que iniciam cadeias derivadas de X

Regras:

1 – Se X é terminal, $\text{First}(X) = \{X\}$

2 – Se X é não-terminal e $X \rightarrow Y_1Y_2... Y_k$ é uma produção ($k \geq 1$), então acrescente a a $\text{First}(X)$ se

- para algum i, a estiver em $\text{First}(Y_i)$ e ϵ estiver em todos os $\text{First}(Y_1...Y_{i-1})$... ou seja, os não-terminais anteriores levam a ϵ .
- Se ϵ está em $\text{First}(Y_j)$ para todo $j=1,2,...,k$ então adicione ϵ em $\text{First}(X)$.
- Se Y_1 não derivar ϵ , não acrescentar mais nada a $\text{First}(X)$
- Se $Y_1 \Rightarrow \epsilon$, então adicionar $\text{First}(Y_2)$, e assim por diante.

3- Se $X \rightarrow \epsilon$ é uma produção, então acrescente ϵ a $\text{First}(X)$.

Análise Sintática Descende (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Follow(X)

Conjunto de terminais que *a* que podem aparecer imediatamente à direita de X em alguma forma sentencial.

Regras

- 1 – Coloque \$ em Follow(S), onde S é o símbolo inicial da gramática e \$ é o marcador de final de entrada.
- 2 – Se houver uma produção $A \rightarrow \alpha B \beta$, então tudo em $\text{First}(\beta)$ exceto ϵ , está em follow(B).
- 3 – Se houver uma produção $A \rightarrow \alpha B$ ou uma produção $A \rightarrow \alpha B \beta$, onde o $\text{First}(\beta)$ com tem ϵ , então inclua o Follow(A) em Follow(B).

Análise Sintática Descende (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Exemplo:

$$E \rightarrow T E'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

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

$$F \rightarrow (E) \mid id$$

$$\text{First}(F) = \text{First}(T) = \text{First}(E) = \{ (, id \}$$

$$\text{First}(E') = \{ +, \varepsilon \}$$

$$\text{First}(T') = \{ *, \varepsilon \}$$

$$\text{Follow}(E) = \text{Follow}(E') = \{), \$ \}$$

$$\text{Follow}(T) = \text{Follow}(T') = \{ +,), \$ \}$$

$$\text{Follow}(F) = \{ *, +,), \$ \}$$

Análise Sintática Descendente (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Algoritmo para construção de uma tabela preditiva $M[A,a]$ onde

- ✓ M = matriz, A não-terminal e a é terminal ou $\$$ (final de cadeia)
- ✓ Entrada = gramática G
- ✓ Saída = tabela de análise M

- ✓ Para cada produção $A \rightarrow \alpha$ da gramática faça:
 - ✓ 1. Para cada terminal a em $\text{First}(\alpha)$, inclua $A \rightarrow \alpha$ em $M[A,a]$
 - ✓ 2. Se ϵ pertence a $\text{First}(\alpha)$, inclua $A \rightarrow \alpha$ em $M[A,b]$ para cada terminal b em $\text{Follow}(A)$.
- ✓ Se ϵ pertence a $\text{First}(\alpha)$ e $\$$ pertence a $\text{Follow}(A)$, acrescente também $A \rightarrow \alpha$ em $M[A,\$]$.

Após realizar os passos, as entradas vazias da matriz devem ser preenchidas com indicação de erro.

Análise Sintática Descendente (cont.)

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Gramáticas LL (1) – classe de gramáticas ao qual se pode implementar analisadores descendentes preditivos (recursivos) sem retrocesso.

✓ LL(1) – L = cadeia de entrada lida da esquerda para a direita (Left-to-right)

✓ L = derivação mais à esquerda (leftmost)

✓ 1 = uso de um símbolo a frente na entrada em cada passo para tomar decisões quanto à ação de análise.

✓ Uma gramática é LL(1) se e somente se, sempre que $A \rightarrow \alpha \mid \beta$ forem duas produções distintas de G , com as condições verdadeiras:

1 - $\text{First}(\alpha)$ e $\text{First}(\beta)$ são conjuntos distintos

✓ 2 – Se ϵ está em $\text{First}(\beta)$, então $\text{First}(\alpha)$ e $\text{Follow}(A)$ são conjuntos distintos. Idem se acontecer o com $\text{First}(\alpha)$.

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

$E \rightarrow T E'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid id$

NÃO TERMINAL	SÍMBOLO DE ENTRADA					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

FIGURA 4.17 Tabela M de análise para o Exemplo 4.32.

$First(F) = First(T) = First(E) = \{ (, id \}$
 $First(E') = \{ +, \epsilon \}$
 $First(T') = \{ *, \epsilon \}$

$Follow(E) = Follow(E') = \{), \$ \}$
 $Follow(T) = Follow(T') = \{ +,), \$ \}$
 $Follow(F) = \{ *, +,), \$ \}$

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

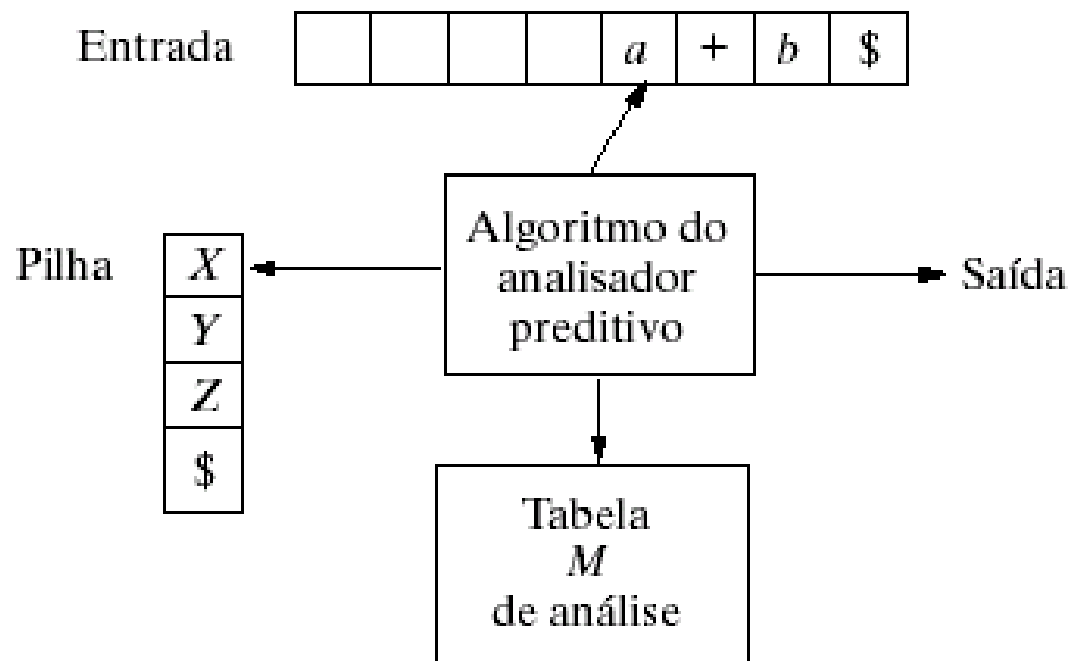


FIGURA 4.19 Modelo de um analisador sintático preditivo ditigido por tabela.

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

```
define ip para que aponte para o primeiro símbolo de w;  
define X para ser o símbolo no topo da pilha;  
while (  $X \neq \$$  ){ /* pilha não está vazia */  
    if ( X é a ) desempilha e avança ip;  
    else if ( X é um terminal ) error();  
    else if ( M[X, a] é uma entrada de erro ) error();  
    else if (  $M[X, a] = X \rightarrow Y_1 Y_2 \dots Y_k$  ) {  
        imprime a produção  $X \rightarrow Y_1 Y_2 \dots Y_k$ ;  
        desempilha X;  
        empilha  $Y_k, Y_{k-1}, \dots, Y_1$  na pilha, com  $Y_1$  no topo;  
    }  
    define X como o símbolo no topo da pilha;  
}
```

FIGURA 4.20 Algoritmo do analisador preditivo.

NÃO TERMINAL	SÍMBOLO DE ENTRADA					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow * FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

FIGURA 4.17 Tabela M de análise para o Exemplo 4.32.

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid id$

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	imprime $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	imprime $T \rightarrow FT'$
	$id T'E'\$$	$id + id * id\$$	imprime $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	casa id
id	$E'\$$	$+ id * id\$$	imprime $T' \rightarrow \epsilon$
id	$+ TE'\$$	$+ id * id\$$	imprime $E' \rightarrow + TE$
$id +$	$TE'\$$	$id * id\$$	casa $+$
$id +$	$FT'E'\$$	$id * id\$$	imprime $T \rightarrow FT'$
$id +$	$id T'E'\$$	$id * id\$$	imprime $F \rightarrow id$
$id+id$	$T'E'\$$	$* id\$$	casa id
$id+id$	$* FT'E'\$$	$* id\$$	imprime $T' \rightarrow * FT'$
$id+id *$	$FT'E'\$$	$id\$$	casa $*$
$id+id *$	$id T'E'\$$	$id\$$	imprime $F \rightarrow id$
$id+id * id$	$T'E'\$$	$\$$	casa id
$id+id * id$	$E'\$$	$\$$	imprime $T' \rightarrow \epsilon$
$id+id * id$	$\$$	$\$$	imprime $E' \rightarrow \epsilon$

FIGURA 4.21 Movimentos efetuados por um analisador sintático preditivo para $id + id * id$.

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Exercícios

1. Faça a análise seguindo a tabela do slide anterior para a cadeia $(id + id) * id$
2. Calcule as relações First e Follow para cada não terminal da gramáticas abaixo

G1:

$$\begin{aligned} S &\rightarrow a b A \mid \epsilon \\ A &\rightarrow a S a a \mid b \end{aligned}$$

G2:

$$\begin{aligned} S &\rightarrow A S \mid B A \\ A &\rightarrow a B \mid C \\ B &\rightarrow b a \mid d \\ C &\rightarrow c \end{aligned}$$

Compiladores

Princípios, técnicas e ferramentas

2ª Edição

Exercícios

3. Calcule as relações First e Follow para cada não terminal da gramática e construa a tabela de análise preditiva para ela. Proponha uma cadeia e analise-a segundo o método descendente preditivo.

$\langle \text{Cmdos} \rangle ::= \langle \text{Cmdo} \rangle ; \langle \text{Cmdos} \rangle \mid \langle \text{Cmdo} \rangle$

$\langle \text{Cmdo} \rangle ::=$ if exp then $\langle \text{Cmdo} \rangle$ $\langle \text{Pelse} \rangle$
 | for id := exp to exp do $\langle \text{Cmdo} \rangle$
 | while exp do $\langle \text{Cmdo} \rangle$
 | id := exp
 | begin $\langle \text{Cmdos} \rangle$ end

$\langle \text{Pelse} \rangle ::=$ else $\langle \text{Cmdo} \rangle \mid \varepsilon$