

Análise Sintática

Analísadores TopDown

Profa. Dra. Andréa Aparecida Konzen

Analísadores Descendentes (Top-down)

- a árvore de derivação correspondente é construída de *cima para baixo*, ou seja, da raiz (símbolo inicial S) para as folhas.
- é necessário decidir qual a regra $A \rightarrow \beta$ deve ser aplicada a um nó rotulado por um não-terminal A . A *expansão* de A é feita criando nós filhos rotulados com os símbolos de β .

Tipos de analisadores

- **Com Retrocesso (1)**

- Análise Recursiva com Retrocesso**

- Mais poderosos, mais lentos, não indicados, testa diferentes possibilidades retrocedendo se falhar

- **Preditivos (2)**

- Análise Recursiva Preditiva (descida recursiva)**

- Análise Preditiva Tabular (LL dirigida por tabela)**

- Tentam prever as construções “seguintes” na cadeia com base em um ou mais marcas de verificação à frente

Algoritmos -> descendentes recursivos (1) / LL(1) não necessita de retrocesso (2)

Gramáticas LL(1)

IMPORTANTE: A tabela sintática não possui entradas múltiplas, não é ambígua e não possui recursividade à esquerda

LL(1) indica que:

- a cadeia de entrada é **examinada da esquerda para a direita**

L = *left-to-right*

- o analisador procura **construir uma derivação esquerda**

L = *leftmost*

- exatamente **1 símbolo** do resto da entrada **é examinado**

Análise Preditiva Tabular - Algoritmo

Construção da tabela sintática preditiva:

Para cada produção $X \rightarrow \alpha$ da gramática execute:

Para cada terminal $a \in \text{first}(\alpha)$

Adicione $X \rightarrow \alpha$ na linha X , coluna a .

Se α pode ser nulo

Para cada terminal $a \in \text{follow}(X)$

Adicione $X \rightarrow \alpha$ na linha X , coluna a

Análise Preditiva Tabular - Algoritmo

Construção da tabela sintática preditiva (cont.):

para cada $a \in \text{first}(\alpha)$

insira $A \rightarrow \alpha$ na posição $[A,a]$

Se $\lambda \in \text{first}(\alpha)$

insira $A \rightarrow \alpha$ na posição $[A,b]$

onde $b \in \text{follow}(A)$

Obs.: Pontos de sincronismo são o conjuntos follow das regras que não tem produção nula. Espaços em branco na tabela correspondem a rotinas de erros.

Exemplo:

Gramática

$E \rightarrow TE'$

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

$T \rightarrow FT'$

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

$F \rightarrow (E) \mid a$

	first	follow
E	(,a	\$,)
E'	+, ε	\$,)
T	(,a	+,), \$
T'	*, ε	+,), \$
F	(,a	*, +,), \$

Exemplo (cont.) – Tabela sintática

LE/LD	a	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	<i>sinc</i>	<i>sinc</i>
E'		$E' \rightarrow + TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$	<i>sinc</i>		$T \rightarrow FT'$	<i>sinc</i>	<i>sinc</i>
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT_1$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow a$	<i>sinc</i>	<i>sinc</i>	$F \rightarrow (E)$	<i>sinc</i>	<i>sinc</i>

Exemplo (cont.) – reconhecimento de entrada

Pilha	Entrada	Produção
\$ E	a+a*a\$	
\$ E' T	a+a*a\$	$E \rightarrow TE'$
\$ E' T' F	a+a*a\$	$T \rightarrow FT'$
\$ E' T' a •	• a+a*a\$	$F \rightarrow a$
\$ E' T'	+a*a\$	
\$ E'	+a*a\$	$T' \rightarrow \varepsilon$
\$ E' T+ •	• +a*a\$	$E' \rightarrow +TE'$
\$ E' T	a*a\$	
\$ E' T' F	a*a\$	$F \rightarrow FT'$
\$ E' T' a •	• a*a\$	$F \rightarrow a$
\$ E' T'	*a\$	
\$ E' T' F* •	• *a\$	$T' \rightarrow *FT'$
\$ E' T' F	a\$	
\$ E' T' a •	• a\$	$F \rightarrow a$
\$ E' T'	\$	$T' \rightarrow \varepsilon$
\$ E'	\$	$E' \rightarrow \varepsilon$