

Analísadores Ascendentes

São analisadores sintáticos que montam a árvore de análise sintática das folhas até a raiz, usando a derivação mais à direita de forma reversa, ou seja, as produções da gramática serão mostradas em ordem inversa.

Este modelo de analisador, também chamado de redutivo, essencialmente é guiado por uma tabela de decisão e baseado em uma pilha de análise, que armazena o estado corrente e os símbolos terminais e não terminais, tendo no topo o não terminal da última regra. Quando toda a cadeia tiver sido lida, no topo deverá estar S permitindo o reconhecimento da sentença.

Exemplo de gramática para gerar listas

$$S \rightarrow [L] \mid a$$
$$L \rightarrow L; S \mid S$$

Cadeia: [a;a]

$$S \Rightarrow [L] \Rightarrow [L;S] \Rightarrow [L;a] \Rightarrow [S;a] \Rightarrow [a;a]$$

fases	Pilha	Entrada	Ação
1	\$	[a;a]	empilha [
2	\$[a;a]\$	empilha a
3	\$[a	;a]\$	reduz $S \rightarrow a$
4	\$[S	;a]\$	reduz $L \rightarrow S$
5	\$[L	;a]\$	Empilha ;
6	\$[L;	a]\$	Empilha a
7	\$[L;a]\$	Reduz $S \rightarrow a$
8	\$[L;S]\$	Reduz $L \rightarrow L;S$
9	\$[L]\$	Empilha]
10	\$[L]	\$	Reduz $S \rightarrow [L]$
11	\$S	\$	Aceita

Ou seja, através de movimentos de empilhamento e de redução o processo termina quando toda a cadeia foi lida e não terminal inicial está no topo.

Redução

O processo de redução é feito por meio dos “Handles” que são basicamente as regras cujo lado direito aparecem em ordem reversa na pilha . A substituição do lado direito lado esquerdo da regra é chamado de redução.

Analísadores LR(k)

São analisadores ascendentes baseados na derivação mais à direita lendo a cadeia da esquerda para a direita (Left to right usando a Rightmost derivation). São Tabulares e usam a pilha para armazenar os símbolos da cadeia e da gramática, como também para armazenar os estados.

Analísador inicia o processamento na configuração

$\langle \$ E_0; a_0 a_1 \dots a_i \dots a_n \$ \rangle$

Em um momento do processo tem – se $\langle \$ E_0 X_1 E_1 \dots X_m E_m; a_i a_{i+1} \dots a_n \$ \rangle$

Tabela de decisão é definida:

Terminais	Não Terminais
<i>Ação</i>	<i>Transição</i>
empilha	próximo estado
reduz	
aceita	
erro	

Movimentos do analisador

SE $TAB[E_m; a_i] = \text{aceita}$ ENTÃO aceita

SE $TAB[E_m; a_i] = \text{erro}$ ENTÃO erro

SE $TAB[E_m; a_i] = \text{empilha X}$ ENTÃO

Configuração = $\langle \$ E_0 X_1 \dots E_m a_i X; a_{i+1} \dots a_n \$ \rangle$

SE $TAB[E_m; a_i] = \text{reduz n}$ ENTÃO (onde $n = \text{numero da regra } Y \rightarrow \beta, |\beta| = r$)

Desempilha $2r$ símbolos, procura $TAB[E_{m-r}, Y] = E_y$

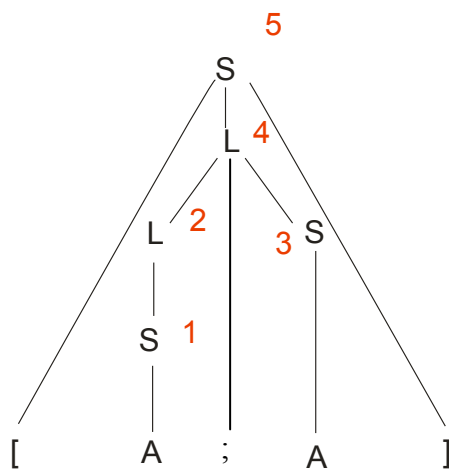
Configuração = $\langle \$ E_0 X_1 \dots E_{m-r} Y E_y; a_i a_{i+1} \dots a_n \rangle$

Gramática

1. $S \rightarrow a$
2. $S \rightarrow [L]$
3. $L \rightarrow L;S$
4. $L \rightarrow S$

	ação					Transição	
	a	[]	;	\$	S	L
0	E2	E3				1	
1					AC		
2			R1	R1	R1		
3	E2	E3				5	4
4			E6	E7			
5			R4	R4			
6			R2	R2	R2		
7	E2	E3				8	
8			R3	R3			

Pilha	Cadeia	Ação
\$0	[a;a]\$	Empilha [
\$0[3	a;a]\$	Empilha a
\$0[3a2	;a]\$	Reduz S → a
\$0[3S5	;a]\$	Reduz L → S
\$0[3L4	;a]\$	Empilha ;
\$0[3L4;7	a]\$	Empilha a
\$0[3L4;7 a2]\$	Reduz S → a
\$0[3L4;7 S8]\$	Reduz L → L;S
\$0[3L4]\$	Empilha]
\$0[3L4]6	\$	Reduz S → [L]
\$0S1	\$	Aceita



$S \Rightarrow^5 [L] \Rightarrow^4 [L;S] \Rightarrow^3 [L;a] \Rightarrow^2 [S;a] \Rightarrow^1 [a;a]$

SLR (1) (Simple LR(1))

Construção da Tabela

Baseado nos itens LR (0), que indicam o percurso de reconhecimento.

$A \rightarrow \cdot XYZ \Rightarrow X \cdot YZ \Rightarrow XY \cdot Z \Rightarrow XYZ \cdot$

(•) = cursor;

Função de Apoio para cálculo dos itens

- Closure (I) {ou Fecho (I)}: composto por todas as regras de I e:
Se $A \rightarrow \cdot B\alpha \in I$ Então as regras $B \rightarrow \cdot \beta$ são acrescentadas de Closure (I)
- GOTO (I,A): calcula o item destino a partir de I quando recebe A
($A \in N \cup \Sigma$), usando a função Closure aplicada ao resultado de GOTO.

Gramática:

1. $S \rightarrow a$
2. $S \rightarrow [L]$
3. $L \rightarrow L;S$
4. $L \rightarrow S$

Exemplo:

$$I = \{L \rightarrow L;S\}$$

$$\begin{aligned} \text{Closure}(I) &= \{ \bullet L \Rightarrow L \rightarrow \bullet L;S \} \\ &= \{ L \rightarrow \bullet L;S, L \rightarrow \bullet S, S \rightarrow \bullet a, S \rightarrow \bullet [L] \} \end{aligned}$$

$$\text{GOTO}(I, []) = \{ S \rightarrow [\bullet L], L \rightarrow \bullet L;S, L \rightarrow \bullet S, S \rightarrow \bullet a, S \rightarrow \bullet [L] \}$$

Algoritmo:

Construir o conjunto C composto por todos os itens marcados, $C = \{I_0, I_1, \dots, I_n\}$. O estado inicial será 0. O item I_0 é obtido de Closure (incluir a regra $S' \rightarrow \bullet S$, para que se tenha um ponto de partida e de término sobre a raiz da gramática).

A partir de C faça:

Se I_j incluir $S' \rightarrow S \bullet$ então

$$\text{TAB}[j, \$] = \text{AC (aceita)}$$

Se $\text{GOTO}(I_j, a) = I_k, a \in \Sigma$ entãoAção $\text{TAB}[j, a] = e k$ (empilha o estado = k)Se $\text{GOTO}(I_j, A) = I_k, A \in N$ entãoTransição $\text{TAB}[j, A] = k$ (transição para o estado k)Se $\{S' \rightarrow S \bullet\} \in I_j$ então

$$\text{TAB}[j, \$] = \text{ACEITA}$$

Se $A \rightarrow \alpha \bullet \in I_j$ então $\forall a \in \text{FOLLOW}(A)$ Faça:

$$\text{TAB}[j, a] = R_n, n = \text{número da regra } A \rightarrow \alpha \bullet \text{ (Redução pela regra } A \rightarrow \alpha)$$

$$I_0 = \{S' \rightarrow \bullet S\} = \{S' \rightarrow \bullet S, S \rightarrow \bullet a, S \rightarrow \bullet [L]\}$$

$$\text{GOTO}(I_0, S) = \{S' \rightarrow S \bullet\} = I_1;$$

$$\text{GOTO}(I_0, a) = \{S \rightarrow a \bullet\} = I_2;$$

$$\text{GOTO}(I_0, []) = \{S \rightarrow [\bullet L], L \rightarrow \bullet L; S, L \rightarrow \bullet S, S \rightarrow \bullet a, S \rightarrow \bullet [L]\} = I_3$$

$$\text{GOTO}(I_3, a) = I_2,$$

$$\text{GOTO}(I_3, []) = I_3$$

$$\text{GOTO}(I_3, L) = \{S \rightarrow [L \bullet], L \rightarrow L \bullet; S\} = I_4$$

$$\text{GOTO}(I_3, S) = \{L \rightarrow S \bullet\} = I_5$$

$$\text{GOTO}(I_4, []) = \{S \rightarrow [L] \bullet\} = I_6$$

$$\text{GOTO}(I_4, ;) = \{L \rightarrow L; \bullet S, S \rightarrow \bullet a, S \rightarrow \bullet [L]\} = I_7$$

$$\text{GOTO}(I_7, S) = \{L \rightarrow L; S \bullet\} = I_8$$

$$\text{GOTO}(I_7, a) = I_2,$$

$$\text{GOTO}(I_7, []) = I_3$$

Cálculo do FOLLOW para a finalização das regras de redução:

FOLLOW (L) = { ; , }

FOLLOW(S) = { \$, ; , }

	ação					Transição	
	a	[]	;	\$	S	L
0	E2	E3				1	
1					AC		
2			R1	R1	R1		
3	E2	E3				5	4
4			E6	E7			
5			R4	R4			
6			R2	R2	R2		
7	E2	E3				8	
8			R3	R3			

Exemplo:

Gramática

1. $E \rightarrow E \vee T$
2. $E \rightarrow T$
3. $T \rightarrow T \& F$
4. $T \rightarrow F$
5. $F \rightarrow (E)$
6. $F \rightarrow id$

Construa a tabela de análise a partir das regras acima.