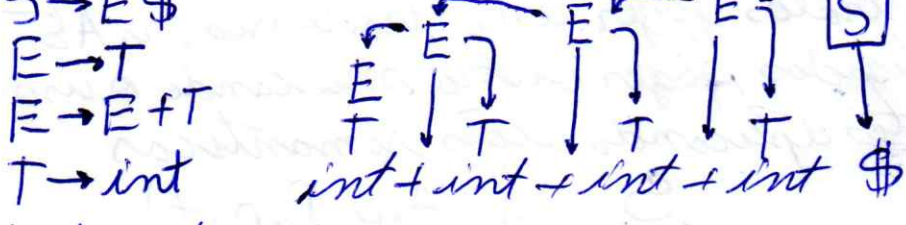


Tabela de Parser, a gramática e LL(1) se um espaço não tiver mais de uma produção

Terminal 'a'  $\in \text{First}(\alpha)$ , adicione  $A \rightarrow \alpha$  em  $M[A, a]$   
 se  $\epsilon \in \text{First}(\alpha)$  então  $\forall$  terminal 'b'  $\in \text{Follow}(A)$ , adicione  $A \rightarrow \epsilon$  em  $M[A, b]$

Gramática	Expansão	Tabela de Parser	First	Follow
$S \rightarrow aAB\epsilon$	$S \rightarrow aAB\epsilon$	$S \rightarrow aAB\epsilon$	$S \{a\}$	$\{ \$ \}$
$A \rightarrow Abck/b$	$A \rightarrow bk$	$A \rightarrow bk$	$A \{b\}$	$\{d\}$
$B \rightarrow d$	$k \rightarrow bck$	$k \rightarrow bck$	$k \{b, \epsilon\}$	$\{d\}$
	$B \rightarrow d$	$B \rightarrow d$	$B \{d\}$	$\{ \epsilon \}$
	$k \rightarrow \epsilon$			

Parser Bottom UP reduz handles, subpalavra igual a direita de uma produção, pelo lado esquerdo, até o símbolo inicial



Automato estado começa com produção inicial ou as que a marca LR • desloca, adiciona as produções dos NT logo após •

Gramática LR(0) usa um automato sem lookahead

Para cada transição de estado  $i \xrightarrow{x} j$   
 se x for terminal  $M[i, x] = S_j$ , shift desloca  $i$  e empilha o estado  $j$   
 se x for NT  $M[i, x] = G_j$ , go to empilha o estado  $j$   
 se  $X \cdot$ , reduce em todo terminal, reduz n símbolos e desempilha m estados  
 se  $S \rightarrow X \cdot \$$ , aceita em \$

