

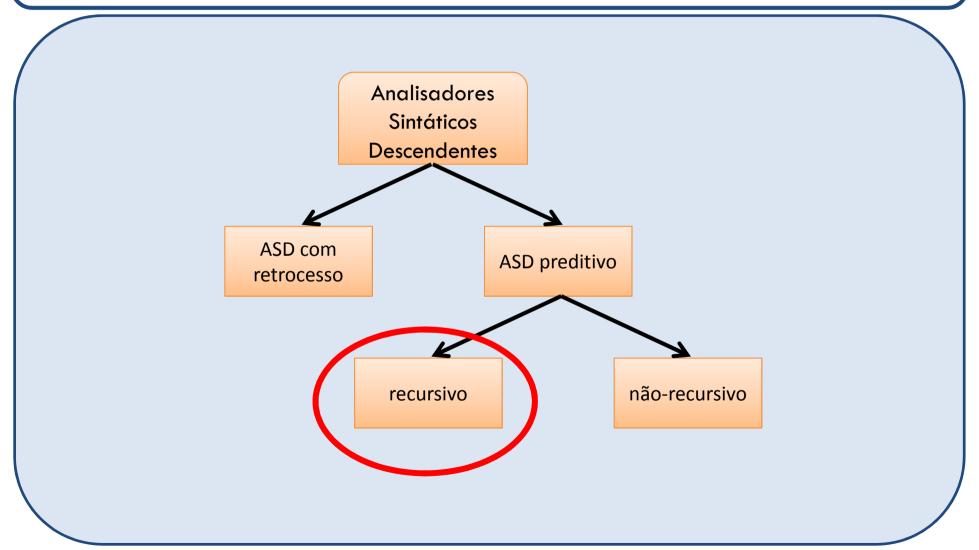
# Compiladores Aula 6

Celso Olivete Júnior

olivete@fct.unesp.br



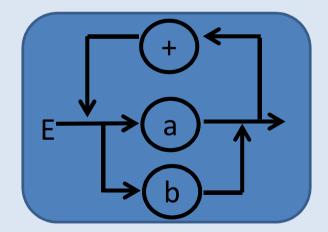
## Na aula passada





Projeto – Parte 2

□ Construa os grafos sintáticos (em número reduzido) e os procedimentos recursivos para declaração de variáveis em LALG



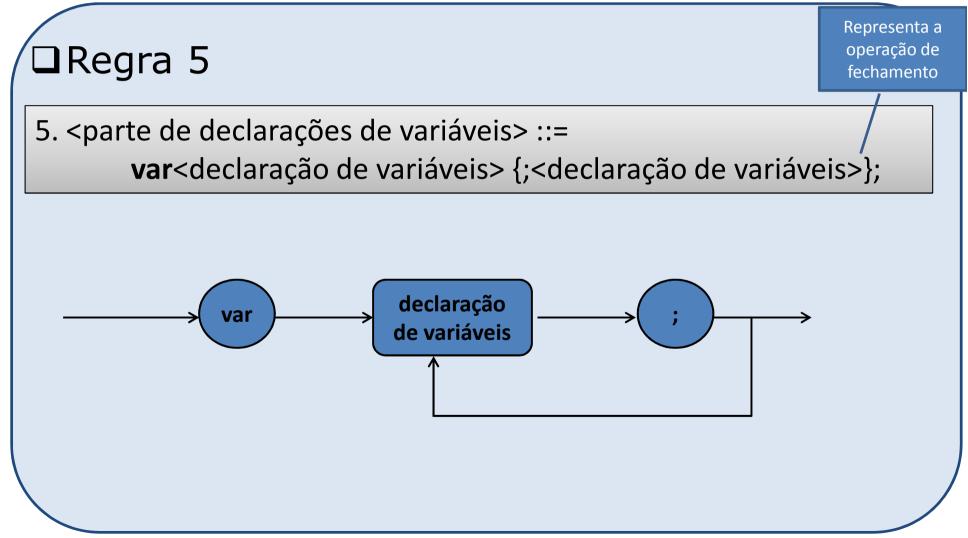


### Grafos sintáticos – declaração de variáveis

#### □LALG - EBNF



### Grafos sintáticos – declaração de variáveis

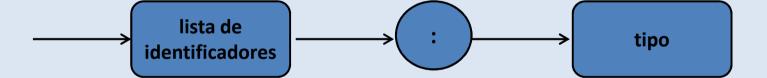




### Grafos sintáticos – declaração de variáveis



6. <declaração de variáveis>::= <lista de identificadores> :<tipo>

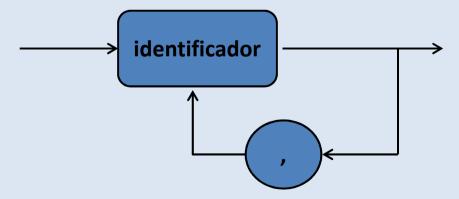




### Grafos sintáticos – declaração de variáveis

#### □Regra 7

7.ta de identificadores> ::= <identificador> {,<identificador>}





Grafos sintáticos – declaração de variáveis

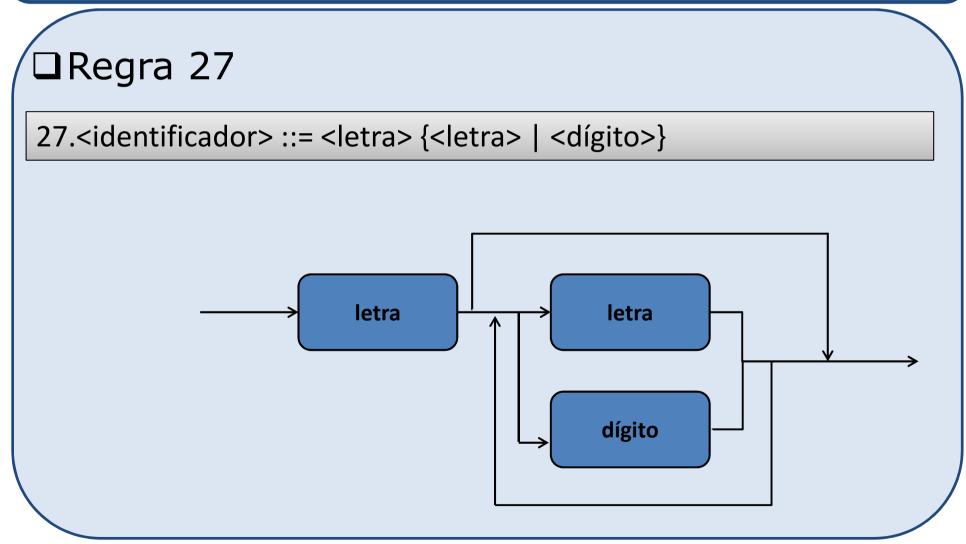
#### ☐Regra 26

26.<dígito> ::= 0 | 1 | 2 | ... 9





### Grafos sintáticos – declaração de variáveis

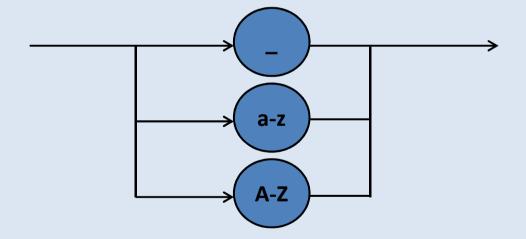




### Grafos sintáticos – declaração de variáveis

### □Regra 28

28.<|etra> ::= \_ | a | ... Z | A | B... | Z

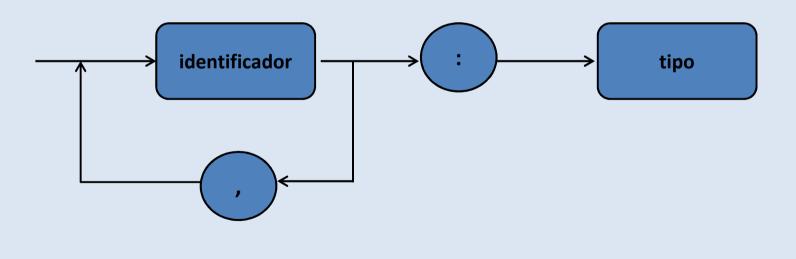




#### Grafos sintáticos – declaração de variáveis

#### □Unindo as regras 6 e 7

- 6. <declaração de variáveis>::= <lista de identificadores> :<tipo>
- 7.ta de identificadores> ::= <identificador> {,<identificador>}

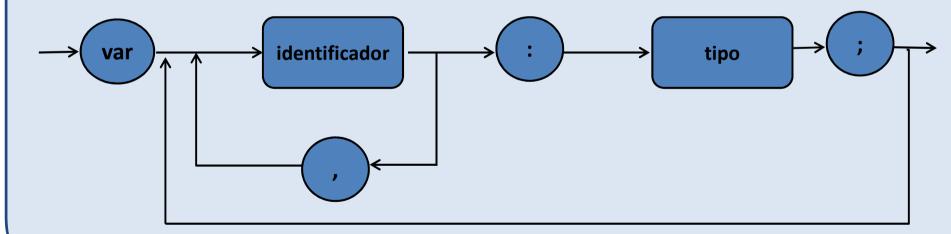




#### Grafos sintáticos – declaração de variáveis

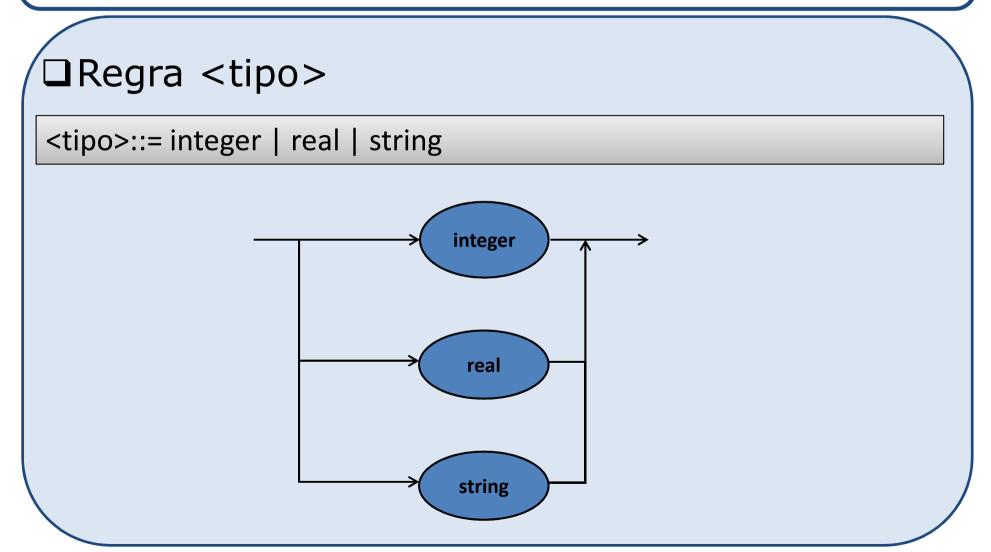
#### ☑Unindo as regras 5, 6 e 7

- 5. <parte de declarações de variáveis> ::=
  - var<declaração de variáveis> {;<declaração de variáveis>};
- 6. <declaração de variáveis>::= <lista de identificadores> :<tipo>
- 7.ta de identificadores> ::= <identificador> {,<identificador>}





Grafos sintáticos – declaração de variáveis



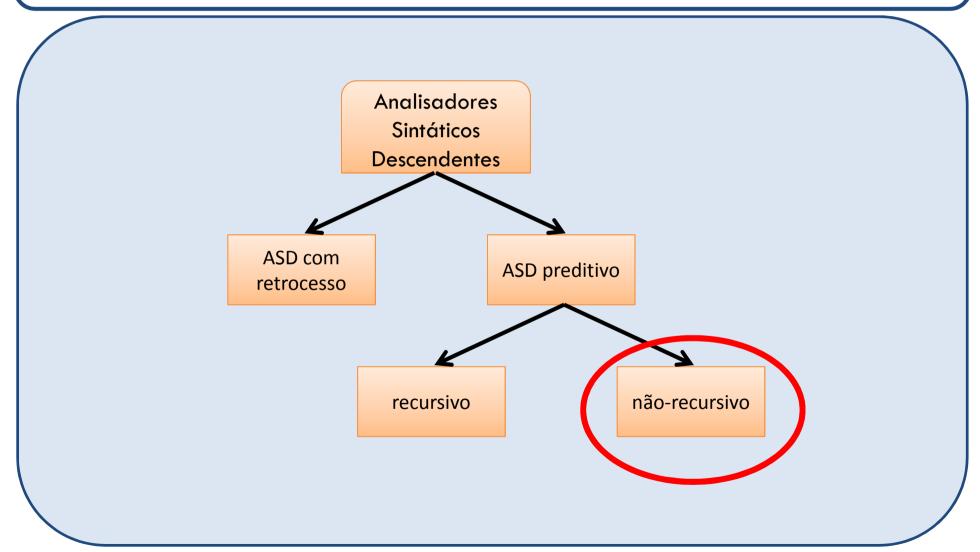


### Analisador sintático descendente

```
<dc_v> ::= var <variaveis> : <tipo_var> ; <dc_v>|ε
procedimento dc v(S)
                                                             Conjunto sincronizador
begin
                                                                {,|;|real|integer}
   se (simb=var) então obter_símbolo()
   senão
         imprimir("Erro: var esperado");
         ERRO(Primeiro(variaveis)+S); //consome até encontrar ID
   variaveis({:}+S);
   se (simb=simb dp) então obter símbolo()
   senão
         imprimir("Erro: ':' esperado");
         ERRO(Primeiro(tipo_var)+S); //consome até encontrar integer ou real
   tipo var({;}+S);
   se (simb=simb pv) então obter símbolo()
   senão
         imprimir("Erro: ';' esperado");
         ERRO(Primeiro(dc_v)+S); //consome até encontrar ;
   dc v(S);
                             <DC V> ::= var <VARIAVEIS> : <TIPO VAR> ; <DC V>
end;
                             <TIPO VAR> ::= integer | real
                             <VARIAVEIS> ::= <ID> <MAIS VAR>
                             <MAIS_VAR> ::= , <VARIAVEIS> \mid \epsilon
```



## Na aula de hoje





#### **Análise Sintática Preditiva Não-Recursiva**

- Não necessitam de retrocesso
- ☐ O símbolo da cadeia de entrada, em análise, é suficiente para determinar qual regra de produção deve ser escolhida
- ☐ São construídos utilizando gramáticas LL(1)
  - > Cadeia de entrada é analisada da esquerda para a direita (Left-to-right)
  - > A derivação das produções é feita mais a esquerda (Leftmost)
  - > A cada passo é observado um (1) símbolo a frente para determinar que ação deve ser tomada



## Análise Sintática Preditiva Não-Recursiva

#### ☐ Condições:

- > Eliminar recursividade à esquerda
- > Fatorar a gramática
- Construir os conjuntos First e Follow
  - ✓ Permitem escolher qual produção deve ser aplicada baseada no próximo símbolo de entrada



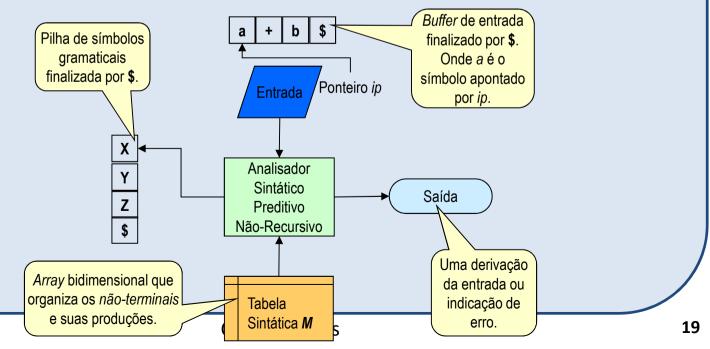
## Reconhecedor sintático preditivo não-recursivo estrutura

Um buffer de entrada → \$ indica o fim da cadeia de entrada ☐ Um fluxo de saída □ Uma pilha cujo fundo é marcado por \$ ☐ Inicializada com o símbolo de início da gramática Uma tabela sintática preditiva Buffer de entrada Pilha de símbolos finalizado por \$. gramaticais Onde a é o finalizada por \$. símbolo apontado Ponteiro ip Entrada por ip. Analisador Sintático Saída **Preditivo** Não-Recursivo Uma derivação Array bidimensional que da entrada ou organiza os *não-terminais*l Tabela indicação de e suas produções. Sintática M erro. 18



## Reconhecedor sintático preditivo funcionamento do parser

- Seja X o símbolo no topo da pilha
- ☐ Seja a o símbolo de entrada (que é terminal) a analisar
  - 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
  - 2. Se X = a e != \$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
  - 3. Se X é não-terminal: Consulta a tabela M(X, a)
    - → Se for vazia: ERRO
    - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).





## Exemplo 1: Construção da tabela preditiva M(X,t)

#### Tabela bi-dimensional

- Dimensão 1: não-terminal X
- > Dimensão 2: símbolo de entrada (terminal) t
- ➤ A entrada (X,t) contém a regra da produção a aplicar → obtida a partir dos conjuntos
  Primeiro e Seguinte

Não	Símbolo de Entrada		
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
Α	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

S 
$$\rightarrow$$
 cAa First(A) = {b, c,  $\varepsilon$ } Follow(A) = {a}  
A  $\rightarrow$  cB | B First(B) = {b,  $\varepsilon$ } Follow(B) = First(B) -  $\varepsilon$  = {a}  
B  $\rightarrow$  bcB |  $\varepsilon$  First(S) = {c} Follow(S) = {\$}

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a e != \$ → desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

6	Entrada:
	"cbca"

Pilha	Entrada	Ação
<b>S</b> \$	<b>c</b> bca\$	

	S→ cAa	
	$A \rightarrow cB \mid B$	
V	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$	

Não	Símbolo de Entrada		
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

Topo da pilha **S** Simb. entrada **a** 

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

4	Entrada:
	"cbca"

Pilha	Entrada	Ação
S <b>\$</b>	cbca\$	S→cAa

	S→ cAa
V	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$

Não	Símbolo de Entrada		
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

Ação 3

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entrada	-
"cbca"	

Pilha	Entrada	Ação
	cbca\$	S→cAa
cAa <mark>\$</mark>	cbca\$	Casar c

	S→ cAa
	$A \rightarrow cB \mid B$
1	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$

Não	Símbolo de Entr		С	
Terminal	a	b	С	
S	ERRO	ERRO	S→cAa	
Α	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$	
В	$B \rightarrow \epsilon$	B →bcB	ERRO	

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a != \$ → desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entrada:	
"cbca"	

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B

	S→ cAa
V	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$

Não	Sím	bolo de Entrada	
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
Α	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a != \$ → desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entrada:
"cbca"

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba\$	bca\$	B→bcB

	S→ cAa
	$A \rightarrow cB \mid B$
1	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$

Não	Sím	bolo de Entr	ada	
Terminal	a	b	С	
S	ERRO	ERRO	S→cAa	
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$	
В	$B \rightarrow \epsilon$	B →bcB	ERRO	

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a != \$ → desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entrada:
"cbca"

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba <b>\$</b>	bca\$	B→bcB
bcBa <b>\$</b>	bca\$	Casar b

	S→ cAa
	$A \rightarrow cB \mid B$
1	$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$

Não	Símbolo de Entrada		
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a != \$ → desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entr	ada	:
"cbd	ca"	

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba <b>\$</b>	bca\$	B→bcB
bcBa <b>\$</b>	bca\$	Casar b
cBa <b>\$</b>	ca\$	Casar c

S→ cAa	
$A \rightarrow cB \mid B$ $B \rightarrow bcB \mid \varepsilon$	

Não	Símbolo de Entrada		
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

## ☐ Entrada: "cbca"

Pilha	Entrada	Ação
<b>S</b> \$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba <b>\$</b>	bca\$	B→bcB
bcBa\$	bca\$	Casar b
cBa <b>\$</b>	ca\$	Casar c
Ba\$	a\$	в→ε

S→ cAa	
$A \rightarrow cB \mid B$	
$B \rightarrow bcB \mid \varepsilon$	

Não	Sím	Símbolo de Entrada		
Terminal	a	b	С	
S	ERRO	ERRO	S→cAa	
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$	
В	$B \rightarrow \epsilon$	B →bcB	ERRO	

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

## ☐ Entrada: "cbca"

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba\$	bca\$	B→bcB
bcBa\$	bca\$	Casar b
cBa <b>\$</b>	ca\$	Casar c
Ba\$	a\$	В→ε
a\$	a\$	Casar a

S→ cAa	
$A \rightarrow cB \mid B$	
$B \rightarrow bcB \mid \varepsilon$	

Não	Sím	ada	
Terminal	a	b	С
S	ERRO	ERRO	S→cAa
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$
В	$B \rightarrow \epsilon$	B →bcB	ERRO

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

## ☐ Entrada: "cbca"

Pilha	Entrada	Ação
S\$	cbca\$	S→cAa
cAa\$	cbca\$	Casar c
Aa\$	bca\$	A→B
Ba\$	bca\$	B→bcB
bcBa <b>\$</b>	bca\$	Casar b
cBa <b>\$</b>	ca\$	Casar c
Ba\$	a\$	B→ε
a\$	a\$	Casar a
\$	\$	aceitou

S→ cAa	
$A \rightarrow cB \mid B$	
$B \rightarrow bcB \mid \varepsilon$	

Não	Símbolo de Entrada			
Terminal	a	b	С	
S	ERRO	ERRO	S→cAa	
А	$A \rightarrow B$	$A \rightarrow B$	$A \rightarrow cB$	
В	$B \rightarrow \epsilon$	B →bcB	ERRO	

□Seja **a** o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

Entrada

Entrada:
"01012"

Piina	Entrada	Ação
<b>S</b> \$	01012\$	

Não Terminal	Símbolo de Entrada				Símbolo de Entrada		
	0 1		2				
S	<s> ::= 0<a></a></s>	<s> ::= 1<b></b></s>	ERRO				
А	ERRO	<a> ::= 1<b></b></a>	<a> ::= 2</a>				
В	<b> ::= 0<a></a></b>	ERRO	<b> ::= 2</b>				

□Seja a o símbolo de entrada (que é terminal) a analisar

- 1. Se X =\$ e a = \$  $\rightarrow$  finaliza e reconhece a entrada.
- 2. Se X = a !=\$  $\rightarrow$  desempilha X e avança de um símbolo na entrada.
- 3. Se X é não-terminal: Consulta a tabela M(X, a)
  - → Se for vazia: ERRO
  - → Se contém X ::= UVW, então substitui na pilha X por UVW (U no topo).

## ☐ Entrada: "01012"

Pilha	Entrada	Ação
<b>S</b> \$	01012\$	<s> ::= 0<a></a></s>
0 <a>\$</a>	01012\$	Casar 0
<a>\$</a>	1012\$	<a> ::= 1<b></b></a>
1 <b>\$</b>	1012\$	Casar 1
<b>\$</b>	012\$	<b> ::= 0<a></a></b>
0 <a>\$</a>	012\$	Casar 0
<a>\$</a>	12\$	<a> ::= 1<b></b></a>
1 <b>\$</b>	12\$	Casar 1
<b>\$</b>	2\$	<b> ::= 2</b>
2\$	2\$	Casar 2
\$	\$	ACEITA

<s> ::= 0<a>  </a></s>	1 <b></b>
<a> ::= 1<b></b></a>	2
<b> ::= 0<a></a></b>	2

Não Terminal	Símbolo de Entrada			
	0	1	2	
S	<s> ::= 0<a></a></s>	<s> ::= 1<b></b></s>	ERRO	
А	ERRO	<a> ::= 1<b></b></a>	<a> ::= 2</a>	
В	<b> ::= 0<a></a></b>	ERRO	<b> ::= 2</b>	



#### **Tabelas Sintáticas Preditivas**

O funcionamento dos analisadores sintáticos preditivos não-recursivos é dependente de tabelas sintáticas preditivas

Como construí-las?



## **Tabelas Sintáticas Preditivas**

NÃO-	IINIAI	Símbolo de Entrada					
TERMINAL	id	+	*	(	)	\$	
Е		E →TE'			$E \rightarrow TE'$		
E'			$E' \rightarrow +TE'$			E' → ε	E'→ ε
Т		T →TF'			T →TF'		
T'			$T' \rightarrow \epsilon$	T' → *FT'		Τ'→ ε	T'→ ε
F		F→id			T →(E)		



#### **Tabelas Sintáticas Preditivas**

- □ A construção de tais tabelas é realizada através de algumas funções associadas a gramática a ser analisada: PRIMEIRO e SEGUINTE.
  - > Ideia: permitem escolher qual produção aplicar com base no próximo símbolo lido.
    - ✓ Durante a <u>recuperação de erro</u>conjunto de **tokens** produzidos por SEGUINTE podem ser usados como **tokens** de sincronismo.



## Construção de Tabelas Sintáticas Preditivas

- ☐ A ideia central é para uma dada produção A:
  - 1. se o símbolo corrente da entrada é a, a expansão da gramática seja dirigida para  $A \rightarrow \alpha$  dado que  $A \rightarrow \alpha$  dado que  $A \rightarrow \alpha$  dado que  $A \rightarrow \alpha$

2. Se  $\alpha = \varepsilon$  ou  $\alpha \rightarrow \varepsilon$  a expansão se dá através de SEGUINTE $(\alpha)$ .



### Construção de Tabelas Sintáticas Preditivas

lacktriangle Para cada produção Aightarrow lpha da gramática repita os passos abaixo:

- 1. Para cada terminal a em PRIMEIRO $(\alpha)$ , adicione  $A \rightarrow \alpha$  a M[A, a]
- 2. Se  $\epsilon$  estiver em PRIMEIRO( $\alpha$ ), adicione  $A \rightarrow \alpha$  a M[A, b] para cada terminal b que estiver em SEGUINTE(A).
- 3. Cada entrada indefinida de M corresponde a uma situação de erro.



```
\begin{array}{lll} \text{PRIM}(E) = \text{PRIM}(T) = \text{PRIM}(F) = \{(, \text{ id}\} \\ \text{PRIM}(E') = \{+, \epsilon\} \\ \text{PRIM}(T') = \{*, \epsilon\} \\ \text{SEG}(E) = \text{SEG}(E') = \{\$, \}\} \\ \text{SEG}(T) = \text{SEG}(T') = \{\$, +, \}\} \\ \text{SEG}(F) = \{\$, *, +, \} \end{array}
```

								-		
NÃO-				Símbolo de Entrada						
TEKN	MINAL	id	+	*	(	)	\$	· \		
Е		E →TE'			E → TE'					
E'			$E' \rightarrow +TE'$			E' → ε	E'→ ε			
			L /+IL			L / C	L 'G			
T										
	۵	Para cada prod	dução A→ α da gı	ramática repita os	passos abaixo	):				
T'	1	1. Para cada terminal $a$ em PRIMEIRO( $\alpha$ ), adicione A $ ightarrow lpha$ a M[A, $a$ ]								
	. 2	2. Se $\epsilon$ estiver em PRIMEIRO( $\alpha$ ), adicione A $\rightarrow$ $\alpha$ a M[A, b] para cada								
F		terminal	terminal b que estiver em SEGUINTE(A).							



```
\begin{array}{ll} \text{PRIM}(E) = \text{PRIM}(T) = \text{PRIM}(F) = \{(\text{, id}\} \\ \text{PRIM}(E') = \{+, \, \epsilon\} \\ \text{PRIM}(T') = \{*, \, \epsilon\} \\ \text{SEG}(E) = \text{SEG}(E') = \, \{\$, \, \} \\ \text{SEG}(T) = \text{SEG}(T') = \, \{\$, \, +, \, \} \\ \text{SEG}(F) = \{\$, \, *, \, +, \, \} \end{array}
```

	NÃO- TERMINAL			Símbolo de Entrada					
			id	+	*	(	)	\$	
Е	$E$ $\square$ Para cada produção $A \! \to \! \alpha$ da gramática repita os passos abaixo:								
		1.	Para cada	terminal a em P	$PRIMEIRO(\alpha)$ , add	icione A $ ightarrow \alpha$ a	M[A, <i>a</i> ]		
E'		2.	Se $\epsilon$ estiv	er em PRIMEII	RO(lpha), adicione A	$A  ightarrow \alpha$ a M[A, b	o] para cada	$E' \rightarrow \varepsilon$	
			terminal b	que estiver em	SEGUINTE(A).				
T			T →FT'			T →FT'			
T'	T'								
F									



```
\begin{array}{lll} \text{PRIM}(E) = & \text{PRIM}(T) = & \text{PRIM}(F) = & \{(, \text{ id}\} \\ \text{PRIM}(E') = & \{+, \epsilon\} \\ \text{PRIM}(T') = & \{*, \epsilon\} \\ \text{SEG}(E) = & \text{SEG}(E') = & \{\$, \} \} \\ \text{SEG}(T) = & \text{SEG}(T') = & \{\$, +, \} \} \\ \text{SEG}(F) = & \{\$, *, +, \} \end{array}
```

NÃO-				Símbolo de E	ntrada			
TERM	IINAL	id	+	*	(	)	\$	
Е	E $lacksquare$ Para cada produção A $ ightarrow lpha$ da gramática repita os passos abaixo:							
	1.	Para cada	terminal a em P	PRIMEIRO( $\alpha$ ), ad	icione A $ ightarrow \alpha$ a	M[A, <i>a</i> ]		
E'	2.	Se $\epsilon$ estiv	ver em PRIMEII	RO(lpha), adicione A	$\alpha  ightarrow \alpha$ a M[A, b	o] para cada	$E' \rightarrow \varepsilon$	
		terminal b	que estiver em	SEGUINTE(A).			_	
T		T →FT'			T →FT'			
T'			$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		Τ'→ ε	Τ'→ ε	
F		F →id			$F \rightarrow (E)$			

unesp

Casar

#### Funcionamento do analisador preditivo

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

Casar	Pilha	Entrada (w)	Ação	
	E\$	id + id\$		

Reconheça a entrada id+id

NÃO-	Símbolo de Entrada						
TERMINAL	id	+	*	(	)	\$	
E	E →TE'	erro	erro	E → TE'	erro	erro	
E'	erro	E'→+TE'	erro	erro	$\mathbf{E'}  ightarrow \epsilon$	E'→ε	
Т	T →FT'	erro	erro	T →FT'	erro	erro	
T'	erro	$T' \rightarrow \epsilon$	T'→*FT'	erro	Τ'→ ε	T'→ ε	
F	F→id	erro	erro	<b>F</b> →( <b>E</b> )	erro	erro	

unesp

Casar

#### Funcionamento do analisador preditivo

E\$

TE'\$

Pilha

Entrada (w)

Ação

id + id\$  $E \rightarrow TE'$ 

id + id\$

NÃO-			Símbolo de Er	ntrada		
TERMINAL	id	+	*	(	)	\$
Е	E →TE'	erro	erro	E → TE'	erro	erro
Е'	erro	E'→+TE'	erro	erro	$E' \rightarrow \epsilon$	E'→ε
Т	T →FT'	erro	erro	T →FT'	erro	erro
T'	erro	$T'\!\to\!\epsilon$	T'→*FT'	erro	Τ'→ ε	T'→ ε
F	F →id	erro	erro	<b>F</b> →( <b>E</b> )	erro	erro

unesp

Casar

#### Funcionamento do analisador preditivo

E\$

TE'\$

FT'E'\$

Pilha

Entrada (w)

Ação

id + id\$  $E \rightarrow TE'$ 

id + id  $T \rightarrow FT'$ 

id + id\$

NÃO-			Símbolo de Ei	ntrada			
TERMINAL	id	+	*	(	)	\$	
E	E →TE'	erro	erro	E → TE'	erro	erro	
Е'	erro	<b>E</b> ' → + <b>TE</b> '	erro	erro	$E' \rightarrow \epsilon$	E'→ε	
T	T →FT'	erro	erro	T →FT'	erro	erro	
T'	erro	$T' \rightarrow \epsilon$	T'→*FT'	erro	$T' \rightarrow \epsilon$	Τ'→ ε	
F	F →id	erro	erro	$\mathbf{F} \rightarrow (\mathbf{E})$	erro	erro	

$\wedge \wedge \wedge$
unesp

#### Funcionamento do analisador preditivo

Casar	Pilha	Entrada (w)	Ação
	E\$	id + id\$	$E \rightarrow TE'$
	TE'\$	id + id\$	$T \rightarrow FT'$
	FT'E'\$	id + id\$	$F \rightarrow id$
	idT'E'\$	id + id\$	

NÃO-		Símbolo de Entrada											
TERMINAL	id	+	*	(	)	\$							
E	E →TE'	erro	erro	E → TE'	erro	erro							
E'	erro	E'→+TE'	erro	erro	$E' \rightarrow \epsilon$	E'→ε							
Т	T →FT'	erro	erro	T →FT'	erro	erro							
T'	erro	$T' \rightarrow \epsilon$	<b>T'</b> → * <b>FT'</b>	erro	T'→ε	$T'\!\!\to\!\epsilon$							
F	F→id	erro	erro	<b>F</b> →( <b>E</b> )	erro	erro							

unesp

#### Funcionamento do analisador preditivo

 $\begin{array}{c} \mathsf{E} \ \to \mathsf{TE'} \\ \mathsf{E'} \ \to \ + \mathsf{TE'} \ | \ \epsilon \\ \mathsf{T} \ \to \ \mathsf{FT'} \\ \mathsf{T'} \ \to \ * \mathsf{FT'} \ | \ \epsilon \\ \mathsf{F} \ \to \ (\mathsf{E}) \ | \ \text{id} \end{array}$ 

С	asar	Pilha		Entrada (v	w)	Açâ	io		
			E\$	id -	+ id\$	E →	TE'		
			TE'\$	id -	+ id\$	T <del>)</del>	FT'		
			FT'E'\$	id -	+ id\$	F →	id		
			idT'E' <mark>\$</mark>	id -	+ id\$	Cas	ar id		
id			T'E'\$	-	+ id\$				
	NÃO-			Símbolo de Ei	ntrada				
	TERMINAL	id	+	*	(		)	\$	
	E	E →TE'	erro	erro	$\mathbf{E} \to \mathbf{T}$	E'	erro	erro	
	E'	erro	E'→+TE'	erro	erro		$E' \rightarrow \epsilon$	E'→ ε	
	Т	T →FT'	erro	erro	T →FT	Γ',	erro	erro	

 $T' \rightarrow *FT'$ 

erro

 $T'\!\to \epsilon$ 

erro

erro

 $F \rightarrow id$ 

erro

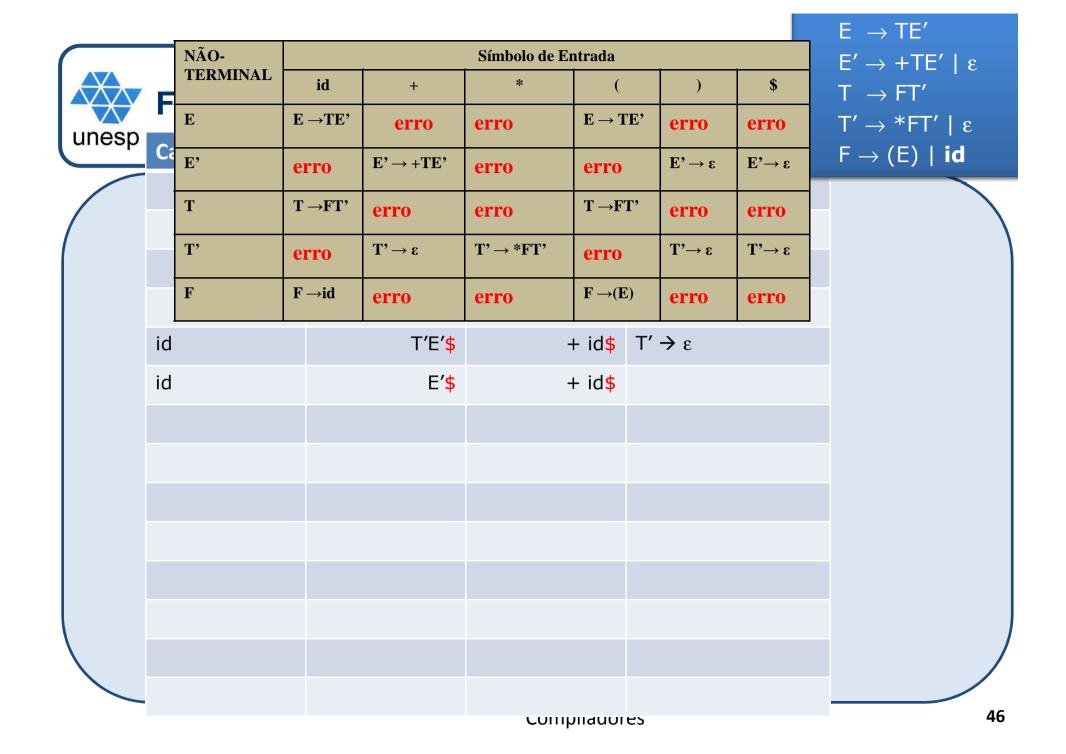
 $\mathbf{F} \rightarrow (\mathbf{E})$ 

 $T' \rightarrow \epsilon$ 

erro

 $T' \rightarrow \epsilon$ 

erro



									$E \;  o TE' \ E'  o + TE' \mid \epsilon$
		NÃO-			Símbolo de E	entrada			$T \rightarrow FT'$
	F	TERMINAL	id	+	*	(	)	\$	$T' \rightarrow *FT' \mid \epsilon$
unesp	Ca	Е	E →TE'	erro	erro	E → TE'	erro	erro	F → (E)   <b>id</b>
		E'	erro	E'→+TE'	erro	erro	$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$	
		T	T →FT'	erro	erro	T →FT'	erro	erro	
		Т'	erro	$\mathbf{T'} \rightarrow \mathbf{\epsilon}$	<b>T'</b> → * <b>FT'</b>	erro	T'→ε	$T' \rightarrow \epsilon$	
	id	F	F →id	erro	erro	<b>F</b> →( <b>E</b> )	erro	erro	
	id			E'\$		+ id\$ E	' → +TE'		
	id			+TE'\$		+ id\$			
					COIII	piiauui es			47

									_	$E \to TE'$	
		NÃO-			Símbolo de E	ntrada				$E' \rightarrow +TE' \mid \epsilon$ $T \rightarrow FT'$	
	F	TERMINAL	id	+	*	(		)	\$	$T' \rightarrow *FT' \mid \epsilon$	
unesp	Cá	E	E →TE'	erro	erro	$E \rightarrow T$	E'	erro	erro	F → (E)   <b>id</b>	
		E'	erro	<b>E</b> ' → + <b>TE</b> '	erro	erro		$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$		
		T	T →FT'	erro	erro	$T \rightarrow F'$	Т'	erro	erro		
		Т'	erro	$\mathbf{T'} \rightarrow \mathbf{\epsilon}$	<b>T'</b> → * <b>FT'</b>	erro		T'→ε	$\textbf{T'}{\rightarrow}\epsilon$		
	id	F	F →id	erro	erro	<b>F</b> → ( <b>E</b>	E)	erro	erro		
	id			E'\$		+ id\$	E'	→ +TE′			
	id			+TE′ <mark>\$</mark>		+ id\$	Ca	sar +			
	id	+		TE'\$		id\$					
					Comp	Jiiauui	CO				48

									-	$E \rightarrow TE'$ $E' \rightarrow +TE' \mid \epsilon$
		NÃO-			Símbolo de E	ntrada				$T \rightarrow FT'$
	_	TERMINAL	id	+	*	(		)	\$	$T' \rightarrow *FT' \mid \epsilon$
unesp	Ca	E	E →TE'	erro	erro	$E \rightarrow T$	TE'	erro	erro	F → (E)   <b>id</b>
		E'	erro	<b>E</b> ' → + <b>TE</b> '	erro	erro		$E' \rightarrow \epsilon$	E'→ε	
		T	T→FT'	erro	erro	$T \rightarrow F'$	T'	erro	erro	
		T'	erro	$\mathbf{T'} \rightarrow \boldsymbol{\epsilon}$	<b>T'</b> → * <b>FT'</b>	erro		T'→ε	T'→ ε	
	id	F	F →id	erro	erro	<b>F</b> → ( <b>E</b>	E)	erro	erro	
	id			E'\$		+ id <b>\$</b>	E'	→ +TE′		
	id			+TE′ <b>\$</b>		+ id <b>\$</b>	Ca	sar +		
	id	+		TE'\$		id <b>\$</b>	Т-	→ FT′		
	id	+		FT'E'\$		id <b>\$</b>				
										)
					COM	Jiiauui	CO			49

unesp	F	NÃO- TERMINAL E	id E →TE'									Ε΄ ΤΕ΄   Γ΄ =Τ΄   ε )   <b>id</b>	ε
		Ε'	erro	E' → +TE'	erro	erro		$E' \rightarrow \epsilon$	<b>Ε'</b> → ε		/ (_		
		T	T →FT'	erro	erro	$T \rightarrow F'$	Т'	erro	erro				
		Т'	erro	$T' \rightarrow \epsilon$	T' → *FT'	erro		T'→ ε	T'→ ε				
	id	F	F→id	erro	erro	<b>F</b> → ( <b>F</b>	E)	erro	erro	Ь			
	id			E'\$		+ id <b>\$</b>	E'	→ +TE′					
	id			+TE′ <mark>\$</mark>		+ id <b>\$</b>	Ca	sar +					
	id	+		TE'\$		id\$	Τ-	→ FT′					
	id	+		FT'E'\$		id\$	F-	→ id					
	id	+		idT'E'\$		id\$							
						Jiiauui							50

	,	NÃO-			Símbolo de E	ntrada			_	1	$E \rightarrow TE'$ $E' \rightarrow +TE' \mid \epsilon$ $T \rightarrow FT'$
		TERMINAL	id	+	*	(		)	\$		$T' \rightarrow *FT' \mid \epsilon$
unesp	Ca	Е	E →TE'	erro	erro	$E \rightarrow T$	ΓE'	erro	erro		F → (E)   <b>id</b>
		Е'	erro	<b>E</b> '→+ <b>TE</b> '	erro	erro		$E' \rightarrow \epsilon$	E'→ε	Г	
		Т	T →FT'	erro	erro	T →F	T'	erro	erro		
		T'	erro	$T'\!\to\epsilon$	T'→*FT'	erro		$T' \rightarrow \epsilon$	T'→ ε		
	id	F	F→id	erro	erro	<b>F</b> →( <b>F</b>		erro	erro		
						. : d &		> . TE/			
	id			E'\$	-	+ 10\$	E.	→ +TE′			
	id			+TE'\$	-	+ id <b>\$</b>	Ca	sar +			
	id	+		TE'\$		id\$	Т-	→ FT′			
	id	+		FT'E'\$		id <b>\$</b>	F-	→ id			
	id	+		idT'E' <mark>\$</mark>		id\$	Ca	sar id			
	id	+ id		T'E'\$		\$					
					COM	niauvi	CO				51

	F	NÃO- TERMINAL	id	+	Símbolo de E	intrada (		)	\$		$E \rightarrow TE'$ $E' \rightarrow +TE' \mid$ $T \rightarrow FT'$ $T' \rightarrow *FT' \mid$
unesp	Ca	E	E →TE'	erro	erro	$E \to T$	E' er	rro	erro		$T' \rightarrow *FT' \mid \epsilon$ $F \rightarrow (E) \mid id$
		Ε'	erro	<b>E</b> '→+ <b>TE</b> '	erro	erro	E'	$^{\prime} \rightarrow \epsilon$	E'→ ε	Г	
		T	T →FT'	erro	erro	$T \rightarrow F'$	r' er	ro	erro		
		T'	erro	$T' \rightarrow \epsilon$	<b>T'</b> → * <b>FT'</b>	erro	T'	'→ ε	$\textbf{T'}\!\rightarrow\!\boldsymbol{\epsilon}$		
	id	F	F→id	erro	erro	<b>F</b> → ( <b>E</b>	c) er	ro	erro	H	
	id			E'\$		+ id <b>\$</b>					
	id			+TE'\$		+ id\$	Casai	r +			
	id	+		TE'\$		id <b>\$</b>	$T \rightarrow F$	-T′			
	id	+		FT'E'\$		id\$	$F \rightarrow i$	d			
	id	+		idT'E' <mark>\$</mark>		id\$	Casai	r id			
	id	+ id		T'E'\$		\$	$T' \rightarrow s$	3			
	id	+ id		E'\$		\$					
					Com	piiauoi	CS				

		NÃO-			Símbolo de E	ntrada			
	F	TERMINAL	id	+	*	(	)	\$	
ınesp	Ca	E	E →TE'	erro	erro erro		E' erro	erro	
		E'	erro	<b>E</b> ' → + <b>TE</b> '	erro	erro	$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$	
		Т	T →FT'	erro	erro	$T \rightarrow F'$	r' erro	erro	
		T'	erro	$\textbf{T'} \rightarrow \boldsymbol{\epsilon}$	T'→*FT'	erro	T'→ε	$T' \rightarrow \epsilon$	
	id	F	F →id	erro	erro	<b>F</b> → ( <b>E</b>	erro erro	erro	
							1 / / C		
	id			E'\$		+ id\$	E' → +TE'		
	id			+TE'\$		+ id\$	Casar +		
	id	+		TE'\$		id\$	$T \rightarrow FT'$		
	id	+		FT'E'\$		id\$	$F \rightarrow id$		
	id	+		idT'E' <mark>\$</mark>		id\$	Casar id		
	id	+ id		T'E'\$		\$	$T' \rightarrow \epsilon$		
	id	+ id		E <b>′\$</b>		\$	$E' \rightarrow \epsilon$		
	id	+ id		\$		\$			

Compliadores

									_	$E^{\prime}  ightarrow TE^{\prime}$ $E^{\prime}  ightarrow +TE^{\prime} \mid \epsilon$
		NÃO-			Símbolo de E	Entrada				$T \rightarrow FT'$
	F	TERMINAL	id	+	*	(		)	\$	$T' \rightarrow *FT' \mid \epsilon$
unesp	Ca	E	E →TE'	erro	erro	$\mathbf{E}  o \mathbf{T}$	E'	erro	erro	F → (E)   <b>id</b>
		E'	erro	E'→+TE'	erro	erro		$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$	
		T	T →FT'	erro	erro	$\mathbf{T} \rightarrow \mathbf{F}'$	Т'	erro	erro	
		T'	erro	$T' \rightarrow \epsilon$	T'→*FT'	erro		T'→ ε	T'→ε	
	id	F	F →id	erro	erro	<b>F</b> →( <b>E</b>		erro	erro	
	id			E' <b>\$</b>		ㅗ i서 <b>¢</b>		→ +TE′		
	id			+TE'\$		+ id <b>\$</b>	Cas	sar +		
	id	+		TE'\$		id <b>\$</b>	T -	→ FT′		
	id	+		FT'E'\$		id\$	F <del>-</del>	→ id		
	id	+		idT'E' <mark>\$</mark>		id\$	Cas	sar id		
	id	+ id		T'E'\$	\$ T' → ε		<b>⇒</b> ε			
	id	+ id		E'\$		\$	E' ·	<b>⇒</b> ε		
	id	+ id		\$		\$	Re	conhece	eu!	



#### Funcionamento do analisador preditivo

Casamento	Pilha	Entrada (w)	Ação
	E\$	id + id\$	E → TE′
	TE'\$	id + id\$	T → FT′
	FT'E'\$	id + id\$	F → id
	idT'E'\$	id + id\$	Casar id
id	T'E'\$	+ id\$	$T' \rightarrow \varepsilon$
id	E'\$	+ id\$	E' → +TE'
id	+TE'\$	+ id\$	Casar +
id +	TE'\$	id\$	$T \rightarrow FT'$
id +	FT'E'\$	id\$	F → id
id +	idT'E'\$	id\$	Casar id
id + id	T'E'\$	\$	$T' \rightarrow \varepsilon$
id + id	E'\$	\$	$E' \rightarrow \varepsilon$
id + id	\$	\$	Reconheceu!



### Código para o analisador sintático preditivo LL(1) baseado em tabela

```
E \rightarrow TE'
E' \rightarrow +TE' \mid \in
T \rightarrow FT'
T' \rightarrow *FT' \mid \in
F \rightarrow (E) \mid id
```

```
while ((topo da pilha for <> $) and (próximo símbolo for <> $)) do
       if (topo da pilha for o terminal a) and (próximo símbolo de entrada for = a)
2.
3.
         then (casar)
              retira da pilha;
4.
5.
              avança entrada;
6.
         else if (topo da pilha for um não-terminal A)
                and (próximo símbolo de entrada for terminal a)
7.
                 and (célula da tabela M[A,a] conter a produção AX_1X_2...X_n)
8.
9.
                   then(gera)
10.
                   retira da pilha;
                   for i:= n downto1 do
11.
12.
                          coloca Xi na pilha;
13.
                   else erro;
     if (topo da pilha for = $) and (marca SEGUINTE na entrada for = $)
14.
15.
         then aceita;
16.
         else erro;
```



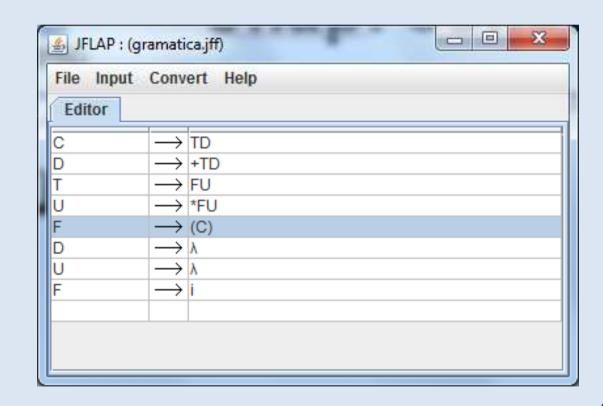
### Jflap: construção da gramática

🗖 É possível gerar uma gramática, neste exemplo LL, a

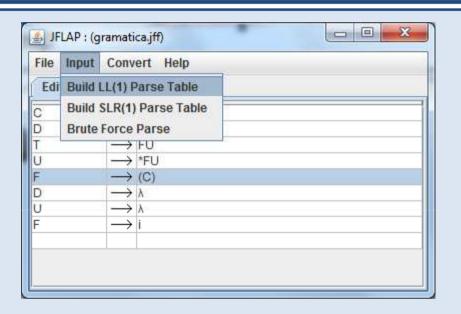
partir do JFlap

```
\begin{array}{c} \mathsf{E} \ \to \mathsf{TE'} \\ \mathsf{E'} \to +\mathsf{TE'} \mid \epsilon \\ \mathsf{T} \ \to \mathsf{FT'} \\ \mathsf{T'} \to *\mathsf{FT'} \mid \epsilon \\ \mathsf{F} \to (\mathsf{E}) \mid \mathsf{id} \end{array}
```

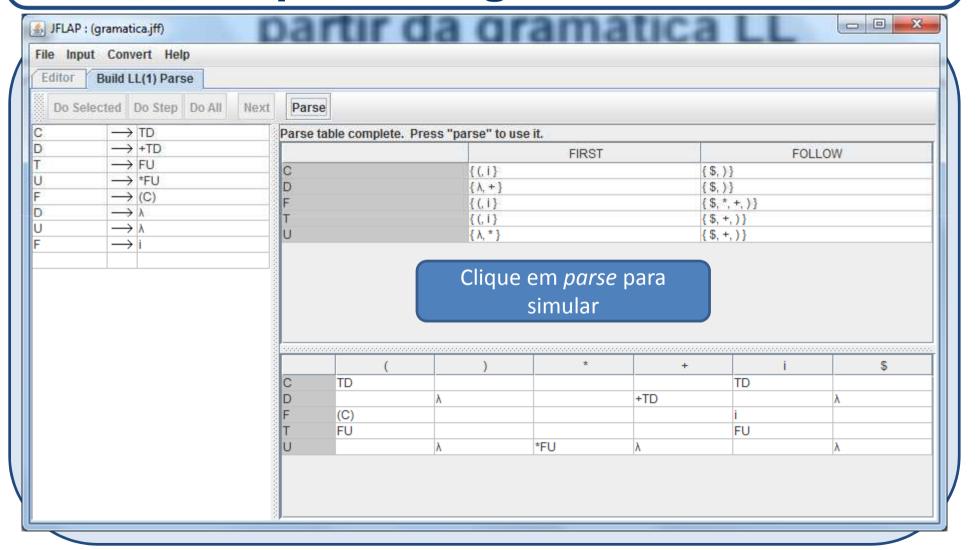
```
E' neste caso foi substituído por D
E por C
T' por U
```



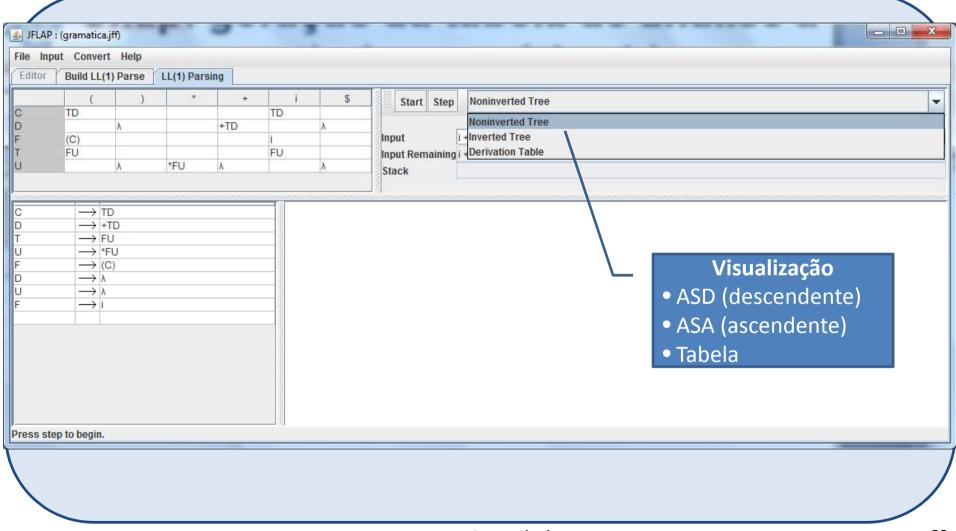




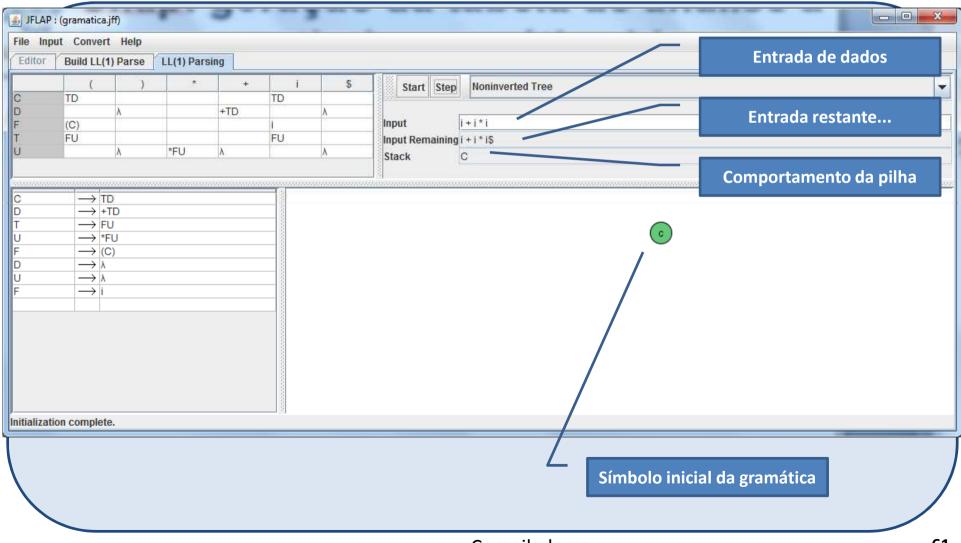




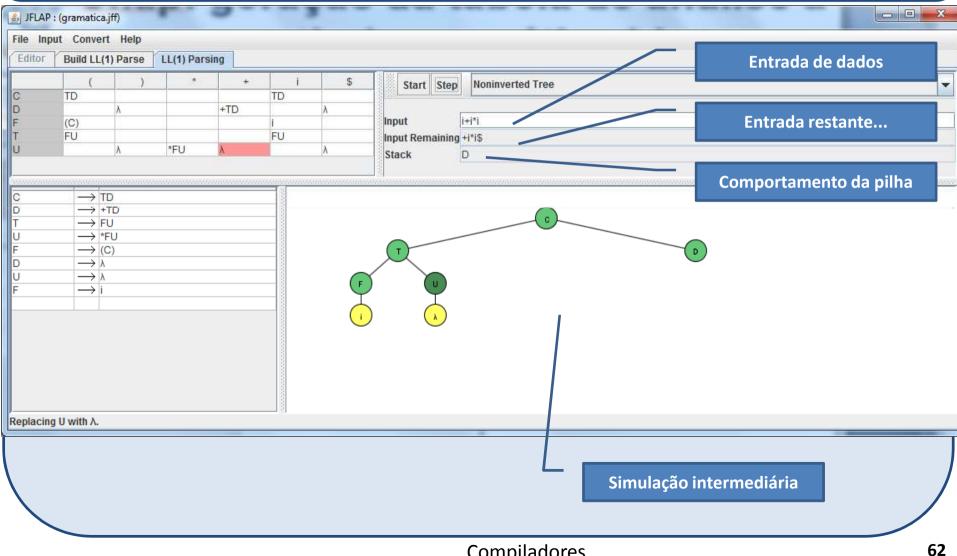




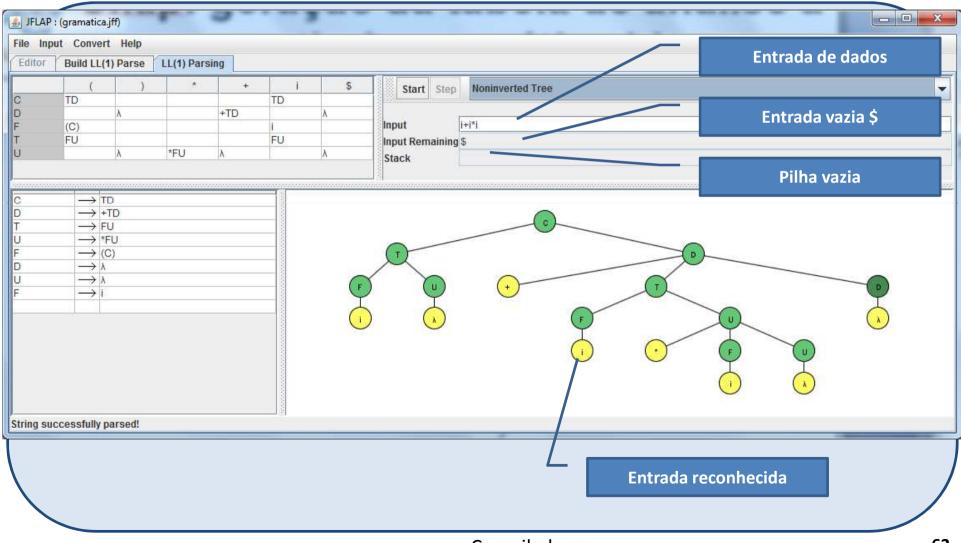




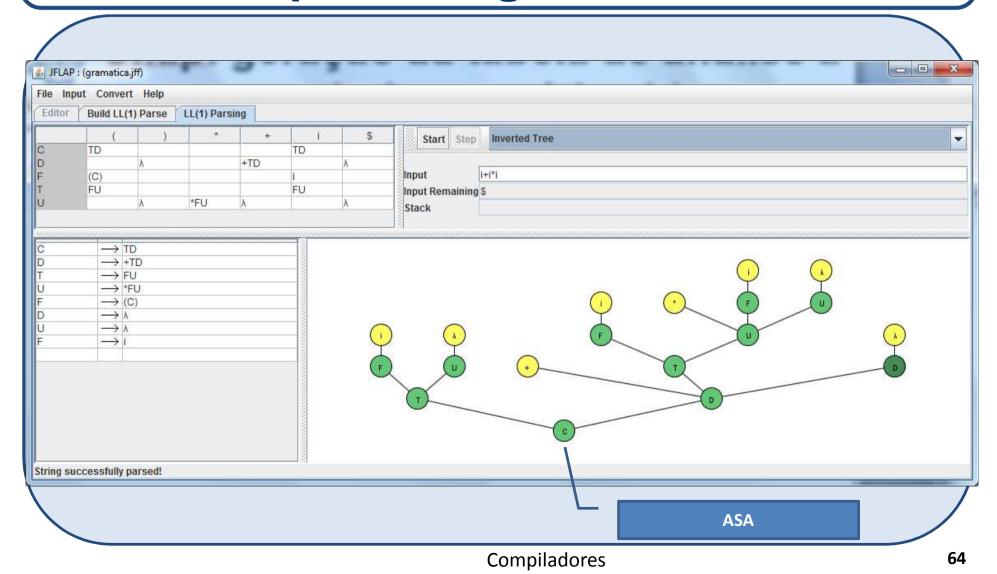




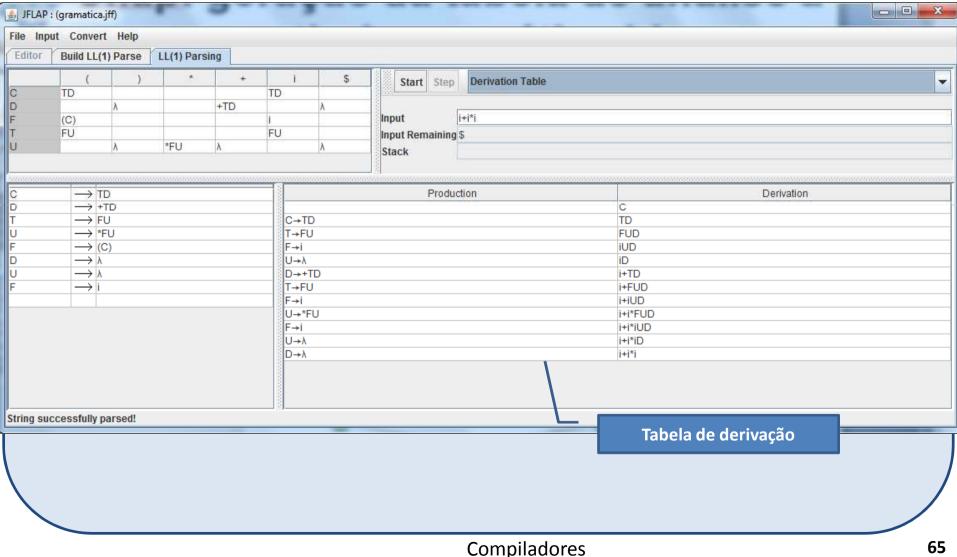














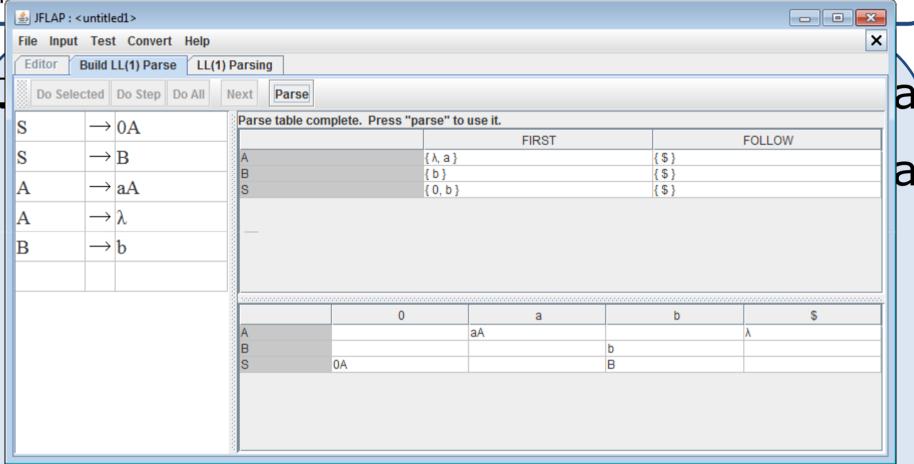
### Exercício (1)

□Construa a tabela sintática para a gramática abaixo e reconheça a sentença **0aa** 

```
<S> ::= 0<A> | <B> <A> ::= a<A> | ε <B> ::= b
```



### Exercício (1)





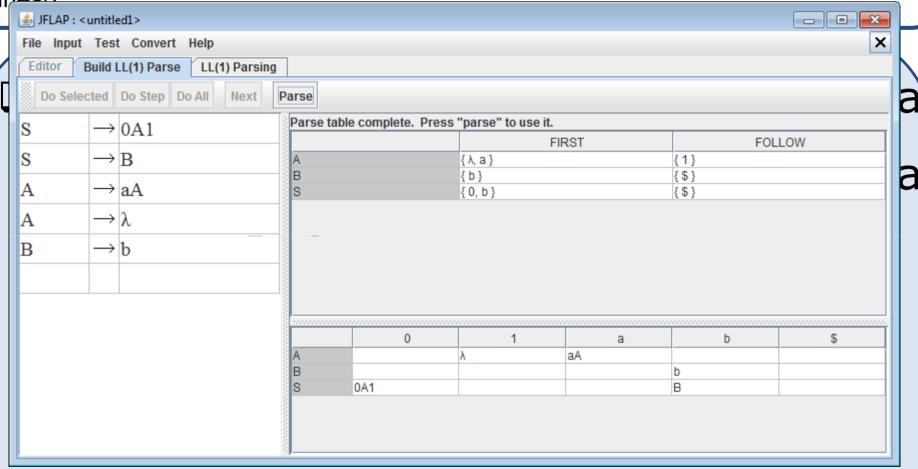
### Exercício (2)

□Construa a tabela sintática para a gramática abaixo e reconheça a sentença **0a1** 

```
<S> ::= 0<A>1 | <B> <A> ::= a<A> | ε <B> ::= b
```



### Exercício (2)





### Exercício (3)

- 🗖 Dada a gramática ambígua:
- S > if E then S | if E then S else S | a
- **E** → **b** 
  - > transforme em LL(1) e :
    - ✓ construa a tabela de análise sintática
    - √ mostrem a análise da sentença

#### If b then if b then a else a

√ mostre o comportamento da Pilha, Entrada e Regra usada



### Exercício (3)

□Dada a gramática ambígua:

S -> if E then S | if E then S else S | a

E →b

> transforme em LL(1)

<S> ::= if <E> then <S> <S'>

<S'> ::= else <S> |ε



### Exercício (3)

Dada a gramática ambígua:

S -> if E then S | if E then S else S | a

E →b

- > transforme em LL(1) e:
  - ✓ mostrem a análise da sentença

#### If b then if b then a else a

✓ mostre o comportamento da Pilha, Entrada e Regra
usada