

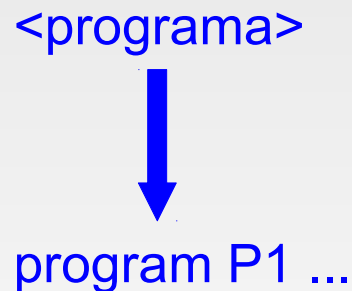
# Construção de Compiladores

## Análise Sintática Descendente – parte2

Profa. Helena Caseli  
helenacaseli@dc.ufscar.br

# Análise Sintática Descendente

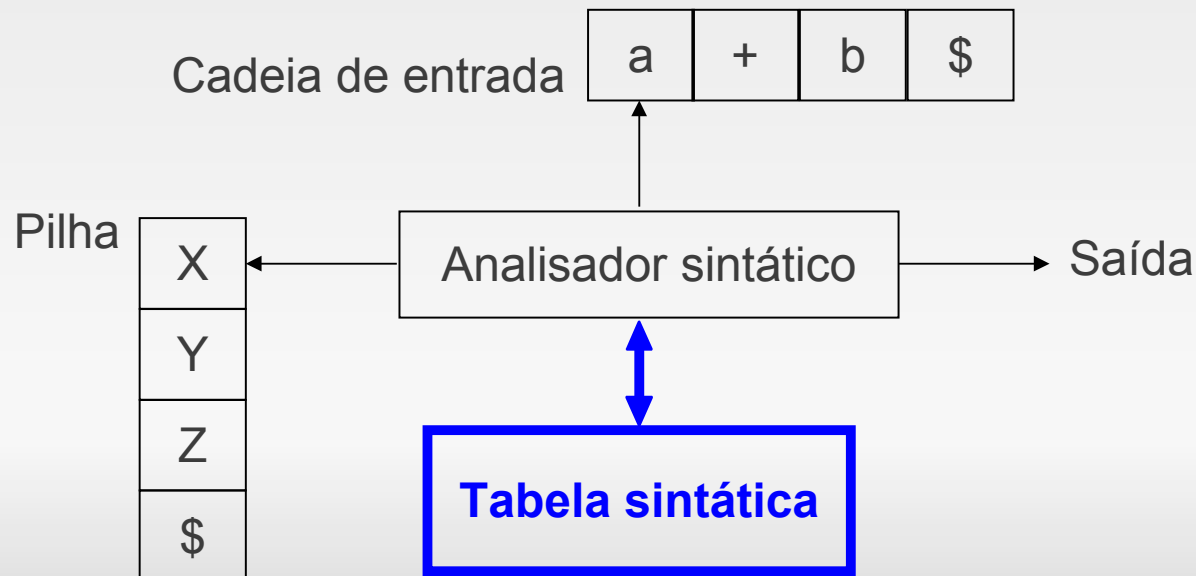
- Como é feita?
  - A análise é feita da raiz para as folhas
  - Parte-se do símbolo inicial da gramática e, por meio de derivações, chega-se à sequência de *tokens* retornada pelo analisador léxico



- Expansão
- Derivação à esquerda

# ASD Preditiva Não Recursiva

- Como é feita?
  - Utiliza-se uma pilha explícita ao invés de ativações recursivas
    - A recursão é substituída pelo uso de uma pilha onde os símbolos sendo expandidos são armazenados
    - Para determinar qual regra gramatical aplicar, consulta-se uma **tabela sintática**



# ASD Preditiva Não Recursiva

- Tabela sintática
  - Uma matriz bidimensional indexada por não-terminais e terminais com escolhas de produções para utilização no passo apropriado da análise
    - N linhas – onde N é o número de não-terminais
    - M+1 colunas – onde M é o número de terminais e a coluna extra equivale ao símbolo de fim de cadeia \$
  - Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \epsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

# ASD Preditiva Não Recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

- Inicialmente, **empilha**

→ O símbolo não-terminal inicial a ser expandido

Cadeia: a+b\$

Pilha



← topo

# ASD Preditiva Não Recursiva

- Exemplo

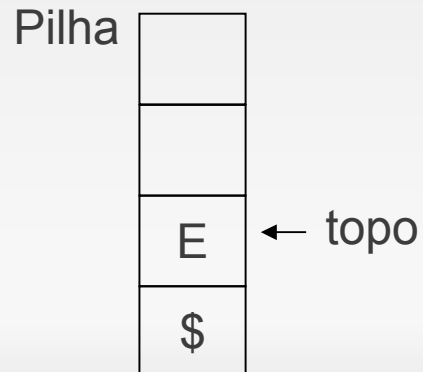
$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

- Ao expandir um não-terminal no topo da pilha, o que acontece?

Cadeia: a+b\$



# ASD Preditiva Não Recursiva

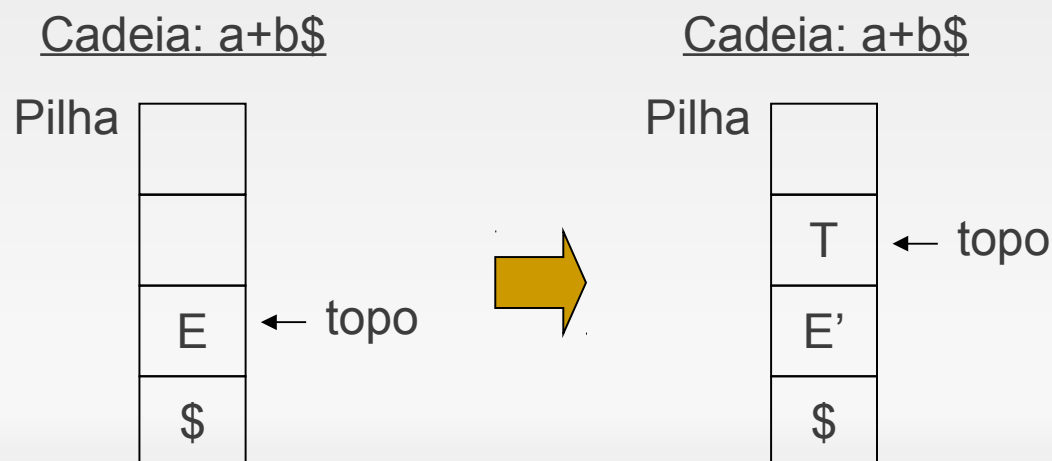
- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

- Ao expandir um não-terminal no topo da pilha
  - Desempilha** o não-terminal
  - Empilha** o lado direito da regra gramatical (em sentido inverso) para expansão (esquerda  $\rightarrow$  direita)



# ASD Preditiva Não Recursiva

- Exemplo

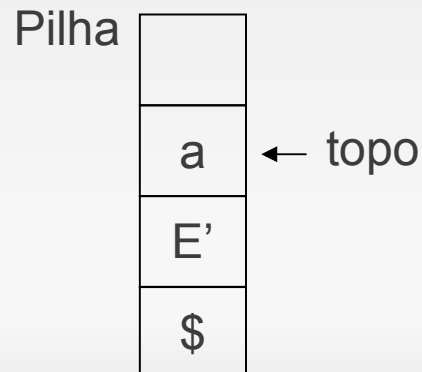
$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

- Ao encontrar um terminal no topo da pilha, o que acontece?

Cadeia: a+b\$





# ASD Preditiva Não Recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

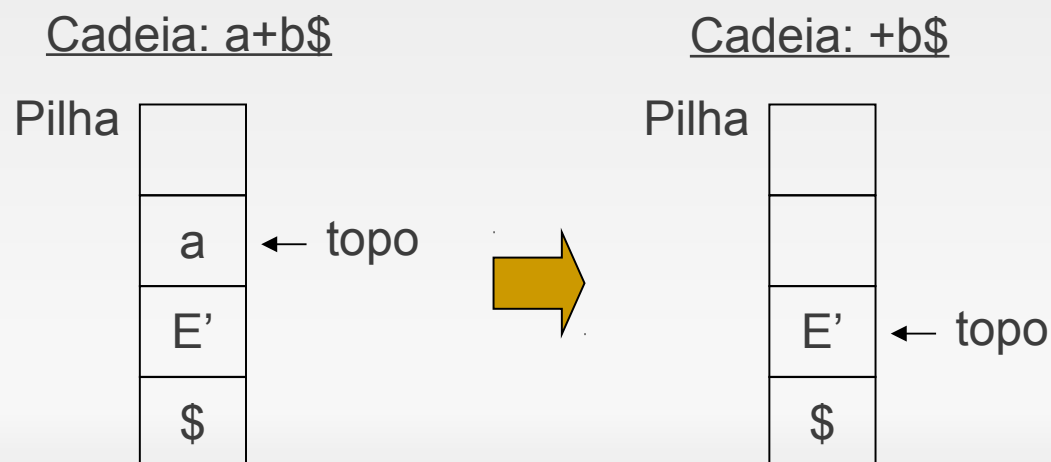
$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

- Ao encontrar um terminal no topo da pilha = símbolo no início da cadeia

1. **Desempilha** o terminal

2. **Consome** o símbolo inicial da cadeia



# ASD Preditiva Não Recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

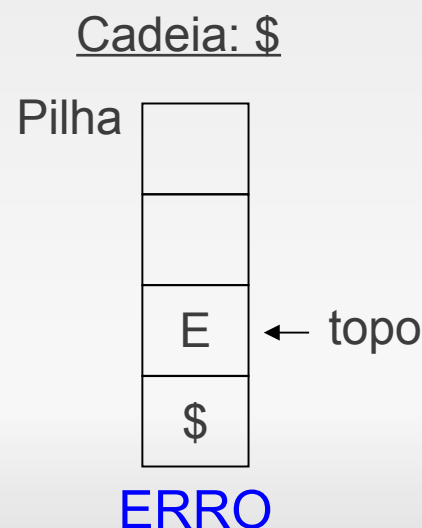
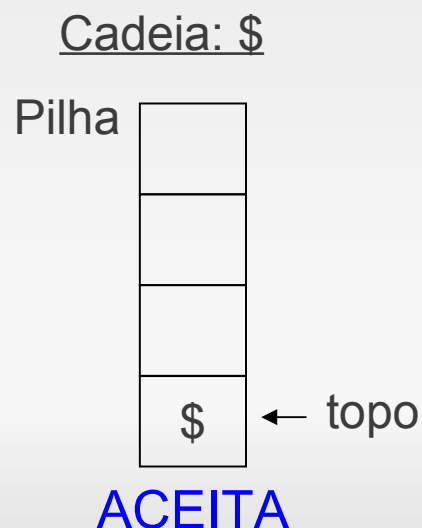
$\langle T \rangle ::= a \mid b$

- Aceita cadeia quando

→ Símbolo topo da pilha = símbolo no início da cadeia = \$

- ERRO

→ Demais casos



# ASD Preditiva Não Recursiva

- Construção da tabela sintática,  $M$ 
  - Para cada não-terminal  $A$  em  $A \rightarrow \alpha$  repita
    1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  a  $M[A,b]$
    2. Se  $\epsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou  $\$$ ) adicione  $A \rightarrow \alpha$  a  $M[A,b]$
- ➔ As entradas não definidas na tabela indicam ERRO

# ASD Preditiva Não Recursiva

- Construção da tabela sintática
  - Exemplo
$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \varepsilon \\ \langle T \rangle &::= a \mid b \end{aligned}$$

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

A tabela sintática é uma matriz  $M[N, T+1]$  em que:

- N é o conjunto de não-terminais da gramática
- T é o conjunto de terminais
- a coluna extra corresponde ao símbolo \$

Tabela sintática

	a	b	+	\$
E				
E'				
T				

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  a  $M[A, b]$

Tabela sintática

	a	b	+	\$
E				
E'				
T				

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  
 $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$   
 $\text{Primeiro}(E') = \{+, \varepsilon\}$

Tabela sintática

	a	b	+	\$
E				
E'				
T				

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  
 $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$   
 $\text{Primeiro}(E') = \{+, \varepsilon\}$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'				
T				

$E \rightarrow TE'$   
 $M[E, a] = M[E, b] = E \rightarrow TE'$



# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  
 $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$   
 $\text{Primeiro}(E') = \{+, \varepsilon\}$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	
T				

$E' \rightarrow + E$   
 $M[E', +] = E' \rightarrow + E$

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  
 $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$   
 $\text{Primeiro}(E') = \{+, \varepsilon\}$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	
T	$T \rightarrow a$	$T \rightarrow b$		

$T \rightarrow a$                        $T \rightarrow b$   
 $M[T, a] = T \rightarrow a$        $M[T, b] = T \rightarrow b$

# ASD Preditiva Não Recursiva

- Construção da tabela sintática
  - Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= a \mid b$

2. Se  $\varepsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou \$) adicione  $A \rightarrow \alpha$  a  $M[A, b]$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	
T	$T \rightarrow a$	$T \rightarrow b$		

# ASD Preditiva Não Recursiva

- Construção da tabela sintática
  - Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= a \mid b$

2. Se  $\varepsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou \$) adicione  $A \rightarrow \alpha$  a  $M[A, b]$

$\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$

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

$\text{Seguidor}(E') = \text{Seguidor}(E) = \{\$ \}$

$\text{Seguidor}(T) = \{+, \$ \}$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	
T	$T \rightarrow a$	$T \rightarrow b$		

# ASD Preditiva Não Recursiva

- Construção da tabela sintática
  - Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= a \mid b$

2. Se  $\varepsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou \$) adicione  $A \rightarrow \alpha$  a  $M[A, b]$

$\text{Primeiro}(E) = \text{Primeiro}(T) = \{a, b\}$

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

$\text{Seguidor}(E') = \text{Seguidor}(E) = \{\$ \}$

$\text{Seguidor}(T) = \{+, \$ \}$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \varepsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

$E' \rightarrow \varepsilon$   
 $M[E', \$] = E' \rightarrow \varepsilon$

# ASD Preditiva Não Recursiva

- Algoritmo de análise sintática preditiva não recursiva

```
(* assume que $ marca o fim da pilha e da entrada *)  
coloca o símbolo de começo no topo da pilha; (* empilha inicial *)  
while topo da pilha for  $\neq$  $ or próximo token for  $\neq$  $ do  
    if topo da pilha for o terminal  $\alpha$  and próximo token de entrada for  $= \alpha$   
    then (* casamento – desempilha terminal e consome entrada *)  
        retira da pilha;  
        avança entrada;  
    else if topo da pilha for um não-terminal  $A$  and próximo token de entrada for  
        o terminal  $\alpha$  and célula da tabela  $M[A, \alpha]$  contiver a produção  $A ::= X_1 X_2 \dots X_n$   
    then (* gera – desempilha não-terminal e empilha lado direito *)  
        retira da pilha;  
        for  $i := n$  downto 1 do coloca  $X_i$  na pilha;  
    else erro;  
end while;  
if topo da pilha for = $ and próximo token na entrada for = $  
then aceita  
else erro;
```

# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

Reconhecer  $a+b$

Pilha	Cadeia	Regra
\$E	a+b\$	

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \varepsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$

$\langle T \rangle ::= a \mid b$

Reconhecer  $a+b$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \epsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

Pilha	Cadeia	Regra
\$E	a+b\$	$E \rightarrow TE'$
\$E'T	a+b\$	$T \rightarrow a$
\$E'a	a+b\$	---
\$E'	+b\$	$E' \rightarrow +E$
\$E+	+b\$	---
\$E	b\$	$E \rightarrow TE'$
\$E'T	b\$	$T \rightarrow b$
\$E'b	b\$	---
\$E'	\$	$E' \rightarrow \epsilon$
\$	\$	<b>Aceita</b>



# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

Reconhecer ab

Pilha	Cadeia	Regra
\$E	ab\$	

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \varepsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exemplo

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= a \mid b$

Tabela sintática

	a	b	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow +E$	$E' \rightarrow \varepsilon$
T	$T \rightarrow a$	$T \rightarrow b$		

Reconhecer ab

Pilha	Cadeia	Regra
\$E	ab\$	$E \rightarrow TE'$
\$E'T	ab\$	$T \rightarrow a$
\$E'a	ab\$	---
\$E'	b\$	<b>Erro</b>

A cadeia não pertence à linguagem!

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exercício

- $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

- $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$

- $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

- $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

- $\langle F \rangle ::= (E) \mid \text{id}$

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

$\langle F \rangle ::= (E) \mid \text{id}$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  a  $M[A, b]$
2. Se  $\varepsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou  $\$$ ) adicione  $A \rightarrow \alpha$  a  $M[A, b]$

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

$\langle F \rangle ::= (E) \mid \text{id}$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione

$A \rightarrow \alpha$  a  $M[A, b]$

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

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

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

2. Se  $\varepsilon \in \text{Primeiro}(\alpha)$  então para cada

$b \in \text{Seguidor}(A)$  (um *token* ou  $\$$ ) adicione

$A \rightarrow \alpha$  a  $M[A, b]$

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

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

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

# ASD Preditiva Não Recursiva

- Construção da tabela sintática

- Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \epsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \epsilon$   
 $\langle F \rangle ::= (E) \mid id$

1. Para cada *token*  $b \in \text{Primeiro}(\alpha)$ , adicione  $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Primeiro}(E) = \text{Primeiro}(T) = \text{Primeiro}(F) = \{ (, id \}$   
 $\text{Primeiro}(E') = \{ +, \epsilon \}$   
 $\text{Primeiro}(T') = \{ *, \epsilon \}$   
 2. Se  $\epsilon \in \text{Primeiro}(\alpha)$  então para cada  $b \in \text{Seguidor}(A)$  (um *token* ou \$) adicione  $A \rightarrow \alpha$  a  $M[A, b]$   
 $\text{Seguidor}(E') = \text{Seguidor}(E) = \{ ), \$ \}$   
 $\text{Seguidor}(T') = \text{Seguidor}(T) = \{ +, ), \$ \}$   
 $\text{Seguidor}(F) = \{ *, +, ), \$ \}$

Tabela sintática

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

# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exercício

Reconhecer  $\text{id}^*(\text{id}+\text{id})$

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

$\langle F \rangle ::= (E) \mid \text{id}$

Pilha	Cadeia	Regra
\$E	$\text{id}^*(\text{id}+\text{id})\$$	

Tabela sintática

	id	+	*	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva

- Exercício

Reconhecer  $\text{id}*\text{id}+\text{id}$ )

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

$\langle F \rangle ::= (E) \mid \text{id}$

Pilha	Cadeia	Regra
\$E	$\text{id}*\text{id}+\text{id})\$$	

Tabela sintática

	id	+	*	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		



# ASD Preditiva Não Recursiva

- Análise sintática descendente preditiva não recursiva
  - Vantagens
    - Permite visualizar os passos da análise sintática com maior clareza
    - Apresenta um algoritmo de análise sintática geral
  - Desvantagens
    - Implica na construção da tabela sintática, o que é muito custoso
    - É aplicável a um conjunto restrito de gramáticas

# ASD Preditiva Não Recursiva

- Exercício
  - Dada a gramática a seguir
$$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid id$$
  - 1. Transforme-a em LL(1)
  - 2. Construa os conjuntos Primeiros e Seguidores
  - 3. Faça um ASD preditivo não recursivo para a gramática LL(1)
    - Construa a tabela sintática
  - 4. Reconheça as cadeias usando o analisador criado
    - $id ** id$
    - $id + id id$

# ASD Preditiva Não Recursiva

- Exercício
  - Dada a gramática a seguir
$$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$$
  - 1. Transforme-a em LL(1)
    - A gramática é ambígua?
    - A gramática é recursiva à esquerda?
    - A gramática está fatorada à esquerda?

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

- $$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$$

1. Transforme-a em LL(1)

- Removendo a ambiguidade da gramática

- Separa operadores de precedência diferente em não terminais diferentes
    - Garante que os operadores com precedência  $<$  ocorram mais próximos da raiz e os de precedência  $>$ , mais distantes da raiz
    - Garante a associatividade de operadores de mesma precedência

Sabendo-se que

- $**$  tem maior precedência e é associativo à direita;
- $*$  tem precedência intermediária e é associativo à esquerda;
- $+$  tem menor precedência e é associativo à esquerda

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Gramática resultante após a remoção da ambiguidade

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- A gramática é recursiva à esquerda?

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Removendo a recursão à esquerda

- Transforma regras do tipo

$\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$  em

$\langle A \rangle ::= \beta \langle A' \rangle$   
 $\langle A' \rangle ::= \alpha \langle A' \rangle \mid \varepsilon$

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Gramática resultante após a remoção da recursão à esquerda



# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- A gramática está fatorada à esquerda?

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle^{**} \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Fatorando à esquerda

- Transforma regras do tipo

$\langle A \rangle ::= \alpha \beta \mid \alpha \gamma$

em

$\langle A \rangle ::= \alpha \langle A' \rangle$

$\langle A' \rangle ::= \beta \mid \gamma$

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle \langle F' \rangle$

$\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Gramática resultante após a fatoração à esquerda

# ASD Preditiva Não Recursiva

- Exercício

- Dada a gramática a seguir

- $\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

- Gramática LL(1) a ser usada no restante do exercício

- $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

- $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

- $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

- $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

- $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$

- $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$

- $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

# ASD Preditiva Não Recursiva

- Exercício

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \varepsilon \\ \langle T \rangle &::= \langle F \rangle \langle T' \rangle \\ \langle T' \rangle &::= * \langle T \rangle \mid \varepsilon \\ \langle F \rangle &::= \langle P \rangle \langle F' \rangle \\ \langle F' \rangle &::= ** \langle F \rangle \mid \varepsilon \\ \langle P \rangle &::= \text{id} \mid (\langle E \rangle) \end{aligned}$$

2. Construa os conjuntos Primeiros e Seguidores

# ASD Preditiva Não Recursiva

## ■ Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$   
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$   
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$   
 $\langle P \rangle ::= \text{id} \mid ( \langle E \rangle )$

2. Construa os conjuntos Primeiros e Seguidores

	Primeiros	Seguidores
E	{ id, ( }	{ \$, ) }
E'	{ +, $\varepsilon$ }	{ \$, ) }
T	{ id, ( }	{ +, \$, ) }
T'	{ *, $\varepsilon$ }	{ +, \$, ) }
F	{ id, ( }	{ *, +, \$, ) }
F'	{ **, $\varepsilon$ }	{ *, +, \$, ) }
P	{ id, ( }	{ **, *, +, \$, ) }

# ASD Preditiva Não Recursiva

- Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$   
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$   
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$   
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

3. Faça um ASD preditivo não recursivo para a gramática LL(1)
  - Construa a tabela sintática

# ASD Preditiva Não Recursiva

## Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= +\langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle \langle F' \rangle$

$\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

	+	*	**	id	(	)	\$
E				$E \rightarrow TE'$	$E \rightarrow TE'$		
E'	$E' \rightarrow +E$					$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T				$T \rightarrow FT'$	$T \rightarrow FT'$		
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow *T$				$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F				$F \rightarrow PF'$	$F \rightarrow PF'$		
F'	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow **F$			$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$
P				$P \rightarrow \text{id}$	$P \rightarrow (E)$		

3. Faça um ASD preditivo não recursivo para a gramática LL(1)

- Construa a tabela sintática



# ASD Preditiva Não Recursiva

- Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$   
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$   
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$   
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

4. Reconheça as cadeias usando o analisador criado

- id \*\* id
- id + id id

# ASD Preditiva Não Recursiva

## Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$   
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$   
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$   
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

## 4. Reconheça as cadeias usando tabela

- id \*\* id
- id + id id

Reconhecer id\*\*id

Pilha	Cadeia	Regra
\$E	id**id\$	$E \rightarrow TE'$
\$E'T	id**id\$	$T \rightarrow FT'$
\$E'T'F	id**id\$	$F \rightarrow PF'$
\$E'T'F'P	id**id\$	$P \rightarrow \text{id}$
\$E'T'F'id	id**id\$	--
\$E'T'F'	**id\$	$F' \rightarrow **F$
\$E'T'F'**	**id\$	--
\$E'T'F	id\$	$F \rightarrow PF'$
\$E'T'F'P	id\$	$P \rightarrow \text{id}$
\$E'T'F'id	id\$	--
\$E'T'F'	\$	$F' \rightarrow \varepsilon$
\$E'T'	\$	$T' \rightarrow \varepsilon$
\$E'	\$	$E' \rightarrow \varepsilon$
\$	\$	ACEITA

# ASD Preditiva Não Recursiva

## Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$   
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$   
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$   
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

## 4. Reconheça as cadeias usando tabela

- id \*\* id
- id + id id

Reconhecer id + id id

Pilha	Cadeia	Regra
\$E	id + id id\$	$E \rightarrow TE'$
\$E'T	id + id id\$	$T \rightarrow FT'$
\$E'T'F	id + id id\$	$F \rightarrow PF'$
\$E'T'F'P	id + id id\$	$P \rightarrow \text{id}$
\$E'T'F'id	id + id id\$	--
\$E'T'F'	+ id id\$	$F' \rightarrow \varepsilon$
\$E'T'	+ id id\$	$T' \rightarrow \varepsilon$
\$E'	+ id id\$	$E' \rightarrow +E$
\$E+	+ id id\$	--
\$E	id id\$	$E \rightarrow TE'$
\$E'T	id id\$	$T \rightarrow FT'$
\$E'T'F	id id\$	$F \rightarrow PF'$
\$E'T'F'P	id id\$	$P \rightarrow \text{id}$
\$E'T'F'id	id id\$	--
\$E'T'F'	id\$	ERRO