

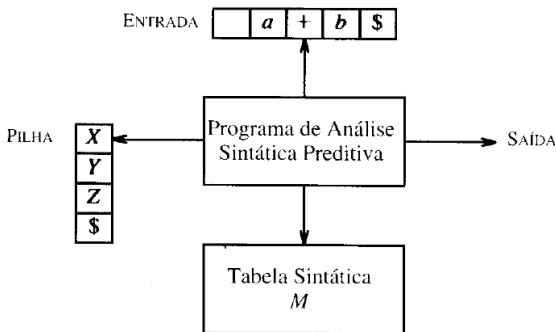
Parsing Descendente (Parte 02)

Sumário

- 1 Parsing Descendente Não-Recursoivo
 - Analisador Preditivo Não-Recursoivo
- 2 Função FOLLOW
 - Função FOLLOW
- 3 Tabela Sintática
 - Tabela Sintática
- 4 Gramática $LL(k)$
 - Gramática $LL(k)$
- 5 Tratamento de Erros
 - Tratamento de Erros

Parsing Preditivo Não-Recursivo

Modelo de Analisador Preditivo Não-Recursivo



Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

```

Faça ip apontar para o primeiro símbolo da entrada w$
repetir
    Seja X o símbolo do topo da pilha e a o símbolo apontado por ip
    se X for um terminal ou '$' então
        se X = a então
            Remover X da pilha e avançar ip
        senão erro()
    senão // X é um não-terminal
        se  $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$  então
            Remover X da pilha
            se  $M[X, a] \neq 'X \rightarrow \lambda'$  então
                Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com  $Y_1$  no topo)
            Escrever a regra  $X \rightarrow Y_1 Y_2 \dots Y_k$  na saída
        senão erro()
até que X = '$'
    
```

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' então

se $X = a$ então

Remover X da pilha e avançar *ip*

senão erro()

senão

// X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ então

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ então

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$
repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*
se X for um terminal ou '\$' então
se $X = a$ então
Remover X da pilha e avançar *ip*
senão erro()
senão // X é um não-terminal
se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ então
Remover X da pilha
se $M[X, a] \neq 'X \rightarrow \lambda'$ então
Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)
Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída
senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$
repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*
se X for um terminal ou '\$' então

se $X = a$ então

Remover X da pilha e avançar *ip*

senão erro()

senão // X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ então

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ então

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$
repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*
se X for um terminal ou '\$' então

se $X = a$ então

Remover X da pilha e avançar *ip*

senão erro()

senão // X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ então

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ então

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada *w*\$
repetir

Seja *X* o símbolo do topo da pilha e *a* o símbolo apontado por *ip*
se *X* for um terminal ou '\$' **então**

se *X* = *a* **então**

Remover *X* da pilha e avançar *ip*

senão erro()

senão // *X* é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover *X* da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que *X* = '\$'

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada *w*\$
repetir

Seja *X* o símbolo do topo da pilha e *a* o símbolo apontado por *ip*
se *X* for um terminal ou '\$' **então**

se *X* = *a* **então**

Remover *X* da pilha e avançar *ip*

senão erro()

senão // *X* é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover *X* da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que *X* = '\$'

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada *w*\$

repetir

Seja *X* o símbolo do topo da pilha e *a* o símbolo apontado por *ip*

se *X* for um terminal ou '\$' **então**

se *X* = *a* **então**

Remover *X* da pilha e avançar *ip*

senão erro()

senão

// *X* é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover *X* da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que *X* = '\$'

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' **então**

se $X = a$ **então**

Remover X da pilha e avançar *ip*

senão erro()

senão // X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' **então**

se $X = a$ **então**

Remover X da pilha e avançar *ip*

senão erro()

senão // X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' **então**

se $X = a$ **então**

Remover X da pilha e avançar *ip*

senão erro()

senão

// X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' **então**

se $X = a$ **então**

Remover X da pilha e avançar *ip*

senão erro()

senão

// X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada *w*\$

repetir

Seja *X* o símbolo do topo da pilha e *a* o símbolo apontado por *ip*

se *X* for um terminal ou '\$' **então**

se *X* = *a* **então**

Remover *X* da pilha e avançar *ip*

senão erro()

senão

// *X* é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover *X* da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que *X* = '\$'

Parsing Preditivo Não-Recursivo

Código de Análise Preditiva Não-Recursiva

Faça *ip* apontar para o primeiro símbolo da entrada $w\$$

repetir

Seja X o símbolo do topo da pilha e a o símbolo apontado por *ip*

se X for um terminal ou '\$' **então**

se $X = a$ **então**

Remover X da pilha e avançar *ip*

senão erro()

senão // X é um não-terminal

se $M[X, a] = 'X \rightarrow Y_1 Y_2 \dots Y_k'$ **então**

Remover X da pilha

se $M[X, a] \neq 'X \rightarrow \lambda'$ **então**

Empilhar ' $Y_1 Y_2 \dots Y_k$ ' (com Y_1 no topo)

Escrever a regra $X \rightarrow Y_1 Y_2 \dots Y_k$ na saída

senão erro()

até que $X = '$'$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow \mathbf{id} \mid (E) \end{aligned}$$

Eliminando as recursões à esquerda, tem-se:

G' :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \mathbf{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \mathbf{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

Eliminando as recursões à esquerda, tem-se:

G' :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Para GLC G' :

$$\begin{aligned}
 E &\rightarrow T E' \\
 E' &\rightarrow + T E' \mid \lambda \\
 T &\rightarrow F T' \\
 T' &\rightarrow * F T' \mid \lambda \\
 F &\rightarrow \text{id} \mid (E)
 \end{aligned}$$

Considere a seguinte tabela sintática:

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE' \$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT' E' \$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$id T' E' \$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T' E' \$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E' \$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+ TE' \$$	$+ id * id\$$	Saída: $E' \rightarrow + TE'$
id +	$TE' \$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT' E' \$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$id T' E' \$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T' E' \$$	$* id\$$	\Rightarrow Casamento: id
id + id	$* FT' E' \$$	$* id\$$	Saída: $T' \rightarrow * FT'$
id + id *	$FT' E' \$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$id T' E' \$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T' E' \$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E' \$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
$id +$	$TE'\$$	$id * id\$$	\Rightarrow Casamento: $+$
$id +$	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
$id +$	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
$id + id$	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
$id + id$	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
$id + id *$	$FT'E'\$$	$id\$$	\Rightarrow Casamento: $*$
$id + id *$	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
$id + id * id$	$T'E'\$$	$\$$	\Rightarrow Casamento: id
$id + id * id$	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
$id + id * id$	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	id\$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	id\$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	\$	\Rightarrow Casamento: id
id + id * id	$E'\$$	\$	Saída: $T' \rightarrow \lambda$
id + id * id	\$	\$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	id\$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	id\$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	\$	\Rightarrow Casamento: id
id + id * id	$E'\$$	\$	Saída: $T' \rightarrow \lambda$
id + id * id	\$	\$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	id\$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	id\$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	\$	\Rightarrow Casamento: id
id + id * id	$E'\$$	\$	Saída: $T' \rightarrow \lambda$
id + id * id	\$	\$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$\text{id} + \text{id} * \text{id}\$$	
	$TE'\$$	$\text{id} + \text{id} * \text{id}\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$\text{id} + \text{id} * \text{id}\$$	Saída: $T \rightarrow FT'$
	$\text{id}T'E'\$$	$\text{id} + \text{id} * \text{id}\$$	Saída: $F \rightarrow \text{id}$
id	$T'E'\$$	$+ \text{id} * \text{id}\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ \text{id} * \text{id}\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ \text{id} * \text{id}\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$\text{id} * \text{id}\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$\text{id} * \text{id}\$$	Saída: $T \rightarrow FT'$
id +	$\text{id}T'E'\$$	$\text{id} * \text{id}\$$	Saída: $F \rightarrow \text{id}$
id + id	$T'E'\$$	$* \text{id}\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* \text{id}\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$\text{id}\$$	\Rightarrow Casamento: *
id + id *	$\text{id}T'E'\$$	$\text{id}\$$	Saída: $F \rightarrow \text{id}$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Parser Preditivo Não-Recursivo – Exemplo (cont.)

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$3E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$id + id * id\$$	
	$TE'\$$	$id + id * id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id + id * id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id + id * id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$+ id * id\$$	\Rightarrow Casamento: id
id	$E'\$$	$+ id * id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id * id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id * id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id * id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id * id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$* id\$$	\Rightarrow Casamento: id
id + id	$*FT'E'\$$	$* id\$$	Saída: $T' \rightarrow *FT'$
id + id *	$FT'E'\$$	$id\$$	\Rightarrow Casamento: *
id + id *	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id * id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id * id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id * id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW

Quando um não-terminal X puder gerar λ (ou $\lambda \in \mathbf{FIRST}(X)$), deve-se determinar quais terminais podem aparecer depois dele em uma forma sentencial.

Define-se **FOLLOW** da seguinte forma:

- Se S é o símbolo de partida, então coloca-se $\$$ em **FOLLOW**(S), sendo $\$$ um marcador de fim da entrada
- Para toda regra da forma $A \rightarrow \alpha X \beta$, tudo em **FIRST**(β) exceto λ é colocado em **FOLLOW**(X)
- Para toda regra da forma $A \rightarrow \alpha X$ ou $A \rightarrow \alpha X \beta$ em que $\lambda \in \mathbf{FIRST}(\beta)$, tudo em **FOLLOW**(A) é colocado em **FOLLOW**(X)

Dessa forma, se X for o não-terminal corrente, a for o próximo símbolo na entrada e existir regra que permita se derivar λ a partir de X , então deve-se aplicar essa regra somente se a estiver no conjunto **FOLLOW** de X .

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

$\text{FOLLOW}(E) = \{ \$ \}$

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned}
 E &\rightarrow T E' \\
 E' &\rightarrow + T E' \mid \lambda \\
 T &\rightarrow F T' \\
 T' &\rightarrow * F T' \mid \lambda \\
 F &\rightarrow \text{id} \mid (E)
 \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

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

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

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

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

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

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

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

$\text{FOLLOW}(F) = \{ '*'\}$

Função FOLLOW

FOLLOW – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

FIRST(E) = **FIRST**(TE') = **FIRST**(T) = **FIRST**(FT') = **FIRST**(F) = $\{ '(', \text{id} \}$

FIRST(E') = **FIRST**($+TE'$) \cup **FIRST**(λ) = $\{ '+', \lambda \}$

FIRST(T') = **FIRST**($*FT'$) \cup **FIRST**(λ) = $\{ '*', \lambda \}$

FOLLOW(E) = **FOLLOW**(E') = $\{ ')', \$ \}$

FOLLOW(T) = **FOLLOW**(T') = $\{ '+', ')', \$ \}$

FOLLOW(F) = $\{ '*', '+', ')', \$ \}$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva

A tabela sintática preditiva M é uma matriz bidimensional cuja entrada $M[A, a]$ indica qual regra do não-terminal A deve ser usada para se obter uma derivação mais à esquerda iniciada pelo terminal a .

Constrói-se a tabela sintática M da seguinte forma:

- Para cada produção $A \rightarrow \alpha$, faça:
 - 1 Para cada terminal a em **FIRST**(α), adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, a]$
 - 2 Se $\lambda \in \mathbf{FIRST}(\alpha)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, b]$, para cada terminal b em **FOLLOW**(A)
 - 3 Se $\lambda \in \mathbf{FIRST}(\alpha)$ e $\$ \in \mathbf{FOLLOW}(A)$, adicionar a regra $A \rightarrow \alpha$ à entrada $M[A, \$]$

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E						
E'						
T						
T'						
F						

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

G: $E \rightarrow T E'$

$E' \rightarrow + T E' \mid \lambda$

$T \rightarrow F T'$

$T' \rightarrow * F T' \mid \lambda$

$F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

$\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$

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

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

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

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	
T			$T \rightarrow FT'$			
T'			$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	
F	$F \rightarrow \text{id}$		$F \rightarrow (E)$			

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

FIRST(E) = **FIRST**(TE') = { '(', id }
FIRST(E') = **FIRST**(+TE') \cup **FIRST**(λ) = { '+', λ }
FIRST(T) = **FIRST**(FT') = { '(', id }
FIRST(T') = **FIRST**(*FT') \cup **FIRST**(λ) = { '*', λ }
FIRST(F) = **FIRST**(id) \cup **FIRST**((E)) = { id, '(' }
FOLLOW(E) = **FOLLOW**(E') = { ')', \$ }
FOLLOW(T) = **FOLLOW**(T') = { '+', ')', \$ }
FOLLOW(F) = { '*', '+', ')', \$ }

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$

$E' \rightarrow + T E' \mid \lambda$

$T \rightarrow F T'$

$T' \rightarrow * F T' \mid \lambda$

$F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

$\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$

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

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

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

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow \text{id} \mid (E)$

$\text{FIRST}(E) = \text{FIRST}(TE') = \{ '(', \text{id} \}$
 $\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$
 $\text{FIRST}(T) = \text{FIRST}(FT') = \{ '(', \text{id} \}$
 $\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$
 $\text{FIRST}(F) = \text{FIRST}(\text{id}) \cup \text{FIRST}((E)) = \{ \text{id}, '(' \}$
 $\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{ ')', \$ \}$
 $\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{ '+', ')', \$ \}$
 $\text{FOLLOW}(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Parsing Não-Recursivo – Tabela Sintática

Tabela Sintática Preditiva – Exemplo

$G: E \rightarrow T E'$
 $E' \rightarrow + T E' \mid \lambda$
 $T \rightarrow F T'$
 $T' \rightarrow * F T' \mid \lambda$
 $F \rightarrow id \mid (E)$

$FIRST(E) = FIRST(TE') = \{ '(', id \}$
 $FIRST(E') = FIRST(+TE') \cup FIRST(\lambda) = \{ '+', \lambda \}$
 $FIRST(T) = FIRST(FT') = \{ '(', id \}$
 $FIRST(T') = FIRST(*FT') \cup FIRST(\lambda) = \{ '*', \lambda \}$
 $FIRST(F) = FIRST(id) \cup FIRST((E)) = \{ id, '(' \}$
 $FOLLOW(E) = FOLLOW(E') = \{ ')', \$ \}$
 $FOLLOW(T) = FOLLOW(T') = \{ '+', ')', \$ \}$
 $FOLLOW(F) = \{ '*', '+', ')', \$ \}$

Tabela Sintática

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Gramática LL(k)

Gramática LL(k)

Uma gramática LL(k) é aquela que pode ser analisada por um *parser* LL(k), isto é, um analisador sintático que faz uma varredura da entrada da esquerda para direita (1º L) produzindo uma derivação mais à esquerda (2º L) utilizando-se de k símbolos (ou *tokens*) da entrada como *lookahead*.

Gramática LL(1)

Seja G uma gramática LL(1) então para todo par de regras distintas de um não terminal A em que $A \rightarrow \alpha \mid \beta$ valem as seguintes condições:

- α e β não derivam ao mesmo tempo sentenças que se iniciam pelo mesmo terminal a , isto é, $\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \emptyset$
- No máximo uma das formas sentenciais produz vazio, isto é, ou $\lambda \in \text{FIRST}(\alpha)$ ou $\lambda \in \text{FIRST}(\beta)$ mas não ambos
- Se $\beta \xRightarrow{*} \lambda$, então α não produz nenhuma sentença que se inicie por um terminal em $\text{FOLLOW}(A)$, isto é, $\text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \emptyset$

Gramática LL(k)

Gramática LL(k)

Uma gramática **LL(k)** é aquela que pode ser analisada por um *parser* **LL(k)**, isto é, um analisador sintático que faz uma varredura da entrada da esquerda para direita (1° L) produzindo uma derivação mais à esquerda (2° L) utilizando-se de k símbolos (ou *tokens*) da entrada como **lookahead**.

Gramática LL(1)

Seja G uma gramática **LL(1)** então para todo par de regras distintas de um não terminal A em que $A \rightarrow \alpha \mid \beta$ valem as seguintes condições:

- α e β não derivam ao mesmo tempo sentenças que se iniciam pelo mesmo terminal a , isto é, $\mathbf{FIRST}(\alpha) \cap \mathbf{FIRST}(\beta) = \emptyset$
- No máximo uma das formas sentenciais produz vazio, isto é, ou $\lambda \in \mathbf{FIRST}(\alpha)$ ou $\lambda \in \mathbf{FIRST}(\beta)$ mas não ambos
- Se $\beta \xRightarrow{*} \lambda$, então α não produz nenhuma sentença que se inicie por um terminal em $\mathbf{FOLLOW}(A)$, isto é, $\mathbf{FIRST}(\alpha) \cap \mathbf{FOLLOW}(A) = \emptyset$

Gramática LL(k)

Gramática LL(1) – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

$\text{FIRST}(E) = \text{FIRST}(TE') = \text{FIRST}(T) = \text{FIRST}(FT') = \text{FIRST}(F) = \{ '(', \text{id} \}$

$\text{FIRST}(E') = \text{FIRST}(+TE') \cup \text{FIRST}(\lambda) = \{ '+', \lambda \}$

$\text{FIRST}(T') = \text{FIRST}(*FT') \cup \text{FIRST}(\lambda) = \{ '*', \lambda \}$

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

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

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

Como todas as regras de G atendem as condições, então G é LL(1)!

Gramática **LL(k)**

Gramática **LL(1)** – Exemplo

Seja GLC G :

$$\begin{aligned} E &\rightarrow T E' \\ E' &\rightarrow + T E' \mid \lambda \\ T &\rightarrow F T' \\ T' &\rightarrow * F T' \mid \lambda \\ F &\rightarrow \text{id} \mid (E) \end{aligned}$$

FIRST(E) = **FIRST**(TE') = **FIRST**(T) = **FIRST**(FT') = **FIRST**(F) = $\{ '(', \text{id} \}$

FIRST(E') = **FIRST**($+TE'$) \cup **FIRST**(λ) = $\{ '+', \lambda \}$

FIRST(T') = **FIRST**($*FT'$) \cup **FIRST**(λ) = $\{ '*', \lambda \}$

FOLLOW(E) = **FOLLOW**(E') = $\{ ')', \$ \}$

FOLLOW(T) = **FOLLOW**(T') = $\{ '+', ')', \$ \}$

FOLLOW(F) = $\{ '*', '+', ')', \$ \}$

Como todas as regras de G atendem as condições, então G é **LL(1)**!

Gramática **LL(k)**

Gramática **LL(1)** – Contra-Exemplo

Seja GLC G : $S \rightarrow i E t S S' \mid a$
 $S' \rightarrow e S \mid \lambda$
 $E \rightarrow b$

FIRST(S) = {**i**, **a**}

FIRST(S') = {**e**, λ }

FIRST(E) = {**b**}

FOLLOW(S) = {**e**, **\$**}

FOLLOW(S') = {**e**, **\$**}

FOLLOW(E) = {**t**}

Então G não é **LL(1)**, pois **FIRST**(eS) \cap **FOLLOW**(S') $\neq \emptyset$

Gramática **LL(k)**

Gramática **LL(1)** – Contra-Exemplo

Seja GLC G : $S \rightarrow i E t S S' \mid a$
 $S' \rightarrow e S \mid \lambda$
 $E \rightarrow b$

FIRST(S) = {**i**, **a**}

FIRST(S') = {**e**, λ }

FIRST(E) = {**b**}

FOLLOW(S) = {**e**, **\$**}

FOLLOW(S') = {**e**, **\$**}

FOLLOW(E) = {**t**}

Então G **não é LL(1)**, pois **FIRST**(eS) \cap **FOLLOW**(S') $\neq \emptyset$

Tratamento de Erros

A pilha de uma analisador preditivo não-recursivo torna explícitos os terminais e não-terminais que o mesmo espera reconhecer com o restante da entrada.

Um erro pode ser detectado durante a análise preditiva quando:

- o terminal do topo da pilha não coincide com o próximo da entrada ou
- a entrada da tabela sintática $M[A, a]$ está vazia, em que A representa o não-terminal que está no topo da pilha e a é o próximo terminal na entrada.

A recuperação de erros na modalidade do desespero (*panic mode*) está baseada na ideia de se ignorar símbolos na entrada até que surja um *token* pertencente a um conjunto pré-selecionado de *tokens* de sincronização.

A efetividade dessa abordagem depende da escolha de tais conjuntos de sincronização.

Esses conjuntos deveriam ser escolhidos de forma que o analisador se recuperasse rapidamente de erros que tendessem a ocorrer na prática.

Tratamento de Erros

A pilha de uma analisador preditivo não-recursivo torna explícitos os terminais e não-terminais que o mesmo espera reconhecer com o restante da entrada.

Um erro pode ser detectado durante a análise preditiva quando:

- o terminal do topo da pilha não coincide com o próximo da entrada ou
- a entrada da tabela sintática $M[A, a]$ está vazia, em que A representa o não-terminal que está no topo da pilha e a é o próximo terminal na entrada.

A recuperação de erros na modalidade do desespero (*panic mode*) está baseada na ideia de se ignorar símbolos na entrada até que surja um *token* pertencente a um conjunto pré-selecionado de *tokens* de sincronização.

A efetividade dessa abordagem depende da escolha de tais conjuntos de sincronização.

Esses conjuntos deveriam ser escolhidos de forma que o analisador se recuperasse rapidamente de erros que tendessem a ocorrer na prática.

Tratamento de Erros

A pilha de uma analisador preditivo não-recursivo torna explícitos os terminais e não-terminais que o mesmo espera reconhecer com o restante da entrada.

Um erro pode ser detectado durante a análise preditiva quando:

- o terminal do topo da pilha não coincide com o próximo da entrada ou
- a entrada da tabela sintática $M[A, a]$ está vazia, em que A representa o não-terminal que está no topo da pilha e a é o próximo terminal na entrada.

A recuperação de erros na modalidade do desespero (*panic mode*) está baseada na ideia de se ignorar símbolos na entrada até que surja um *token* pertencente a um conjunto pré-selecionado de *tokens* de sincronização.

A efetividade dessa abordagem depende da escolha de tais conjuntos de sincronização.

Esses conjuntos deveriam ser escolhidos de forma que o analisador se recuperasse rapidamente de erros que tendessem a ocorrer na prática.

Tratamento de Erros

A pilha de uma analisador preditivo não-recursivo torna explícitos os terminais e não-terminais que o mesmo espera reconhecer com o restante da entrada.

Um erro pode ser detectado durante a análise preditiva quando:

- o terminal do topo da pilha não coincide com o próximo da entrada ou
- a entrada da tabela sintática $M[A, a]$ está vazia, em que A representa o não-terminal que está no topo da pilha e a é o próximo terminal na entrada.

A recuperação de erros na modalidade do desespero (*panic mode*) está baseada na ideia de se ignorar símbolos na entrada até que surja um *token* pertencente a um conjunto pré-selecionado de *tokens* de sincronização.

A efetividade dessa abordagem depende da escolha de tais conjuntos de sincronização.

Esses conjuntos deveriam ser escolhidos de forma que o analisador se recuperasse rapidamente de erros que tendessem a ocorrer na prática.

Tratamento de Erros

A pilha de uma analisador preditivo não-recursivo torna explícitos os terminais e não-terminais que o mesmo espera reconhecer com o restante da entrada.

Um erro pode ser detectado durante a análise preditiva quando:

- o terminal do topo da pilha não coincide com o próximo da entrada ou
- a entrada da tabela sintática $M[A, a]$ está vazia, em que A representa o não-terminal que está no topo da pilha e a é o próximo terminal na entrada.

A recuperação de erros na modalidade do desespero (*panic mode*) está baseada na ideia de se ignorar símbolos na entrada até que surja um *token* pertencente a um conjunto pré-selecionado de *tokens* de sincronização.

A efetividade dessa abordagem depende da escolha de tais conjuntos de sincronização.

Esses conjuntos deveriam ser escolhidos de forma que o analisador se recuperasse rapidamente de erros que tendessem a ocorrer na prática.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização:

- 1 Como ponto de partida, pode-se usar todos os símbolos de **FOLLOW**(A) no conjunto de *tokens* de sincronização para o não-terminal A .

Assim se ignorarmos todas as *tokens* até que um elemento de **FOLLOW**(A) seja visto e removermos o não-terminal A da pilha, é provável que a análise possa continuar.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização:

- 1 Como ponto de partida, pode-se usar todos os símbolos de **FOLLOW**(A) no conjunto de *tokens* de sincronização para o não-terminal A .

Assim se ignorarmos todas as *tokens* até que um elemento de **FOLLOW**(A) seja visto e removermos o não-terminal A da pilha, é provável que a análise possa continuar.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização:

- 1 Como ponto de partida, pode-se usar todos os símbolos de **FOLLOW**(A) no conjunto de *tokens* de sincronização para o não-terminal A .

Assim se ignorarmos todas as *tokens* até que um elemento de **FOLLOW**(A) seja visto e removermos o não-terminal A da pilha, é provável que a análise possa continuar.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW(A)** como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW**(A) como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW**(A) como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW**(A) como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW**(A) como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Geralmente, não é suficiente usar **FOLLOW**(A) como conjunto de sincronização para A.

Por exemplo, se pontos-e-vírgulas terminarem os enunciados (como em C, C++ e Java), então as palavras-chave que iniciam os enunciados não devem aparecer no conjunto **FOLLOW** do não terminal que gera expressões.

Assim, um ponto-e-vírgula ausente após uma atribuição poderia resultar na desconsideração da palavra-chave que inicia o próximo enunciado.

Solução: explorar a estrutura hierárquica que existe nas construções de uma linguagem (por exemplo: expressões aparecem em enunciados que figuram dentro de blocos e assim por diante).

Pode-se adicionar ao conjunto de sincronização de uma estrutura mais baixa os símbolos que começam as construções mais altas (por exemplo, adicionar as palavras-chave que iniciam comandos aos conjuntos de sincronização dos não-terminais que geram expressões).

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Se adicionarmos os símbolos de $FIRST(A)$ ao conjunto de sincronização para o não-terminal A , pode ser possível retornar a análise a partir de A , caso um símbolo de $FIRST(A)$ apareça na entrada.
- 3 Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.
Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.
Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.
- 4 Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.
Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Se adicionarmos os símbolos de **FIRST(A)** ao conjunto de sincronização para o não-terminal A , pode ser possível retornar a análise a partir de A , caso um símbolo de **FIRST(A)** apareça na entrada.

- 3 Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- 4 Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- ② Se adicionarmos os símbolos de **FIRST**(A) ao conjunto de sincronização para o não-terminal A, pode ser possível retornar a análise a partir de A, caso um símbolo de **FIRST**(A) apareça na entrada.
- ③ Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- ④ Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Se adicionarmos os símbolos de **FIRST**(A) ao conjunto de sincronização para o não-terminal A , pode ser possível retornar a análise a partir de A , caso um símbolo de **FIRST**(A) apareça na entrada.

- 3 Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- 4 Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- 2 Se adicionarmos os símbolos de **FIRST**(A) ao conjunto de sincronização para o não-terminal A, pode ser possível retornar a análise a partir de A, caso um símbolo de **FIRST**(A) apareça na entrada.

- 3 Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- 4 Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- ② Se adicionarmos os símbolos de **FIRST**(A) ao conjunto de sincronização para o não-terminal A , pode ser possível retornar a análise a partir de A , caso um símbolo de **FIRST**(A) apareça na entrada.
- ③ Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- ④ Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Tratamento de Erros

Eis algumas heurísticas para escolha do conjunto de sincronização (cont.):

- ② Se adicionarmos os símbolos de **FIRST**(A) ao conjunto de sincronização para o não-terminal A , pode ser possível retornar a análise a partir de A , caso um símbolo de **FIRST**(A) apareça na entrada.
- ③ Se um não-terminal puder gerar a cadeia vazia, então a produção que deriva λ pode ser usada como *default*.

Agindo-se assim, pode-se postergar a detecção de algum erro, mas não se pode fazer com que um erro seja perdido.

Essa abordagem reduz o número de não-terminais que devem ser considerados na recuperação de erros.

- ④ Se um terminal do topo da pilha não puder ser reconhecido, uma ideia é simplesmente desempilhá-lo, emitir uma mensagem de que o terminal foi inserido e prosseguir na análise.

Consequentemente, essa abordagem faz que com o conjunto de sincronização de um *token* consista em todos os demais *tokens*.

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: $+$
		$TE' \$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT' E' \$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$id T' E' \$$	$id * + id\$$	Saída: $F \rightarrow id$
id		$T' E' \$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT' E' \$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *		$FT' E' \$$	$+ id\$$	\Rightarrow Casamento: $+$
id		$T' E' \$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E' \$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE' \$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE' \$$	$id\$$	\Rightarrow Casamento: $+$
id +		$FT' E' \$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$id T' E' \$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T' E' \$$	$\$$	\Rightarrow Casamento: id
id + id		$E' \$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE' \$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT' E' \$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$id T' E' \$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T' E' \$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT' E' \$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT' E' \$$	$+ id\$$	\Rightarrow Casamento: *
id	$T' E' \$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E' \$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE' \$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE' \$$	$id\$$	\Rightarrow Casamento: +
id +	$FT' E' \$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$id T' E' \$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T' E' \$$	$\$$	\Rightarrow Casamento: id
id + id	$E' \$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE' \$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT' E' \$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$id T' E' \$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T' E' \$$	$* + id\$$	\Rightarrow Casamento: id
id	$* FT' E' \$$	$* + id\$$	Saída: $T' \rightarrow * FT'$
id *	$FT' E' \$$	$+ id\$$	\Rightarrow Casamento: +
id	$T' E' \$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E' \$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+ TE' \$$	$+ id\$$	Saída: $E' \rightarrow + TE'$
id +	$TE' \$$	$id\$$	\Rightarrow Casamento: +
id +	$FT' E' \$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$id T' E' \$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T' E' \$$	$\$$	\Rightarrow Casamento: id
id + id	$E' \$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE' \$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT' E' \$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$id T' E' \$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T' E' \$$	$* + id\$$	\Rightarrow Casamento: id
id	$* FT' E' \$$	$* + id\$$	Saída: $T' \rightarrow * FT'$
id *	$FT' E' \$$	$+ id\$$	\Rightarrow Casamento: +
id	$T' E' \$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E' \$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+ TE' \$$	$+ id\$$	Saída: $E' \rightarrow + TE'$
id +	$TE' \$$	$id\$$	\Rightarrow Casamento: +
id +	$FT' E' \$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$id T' E' \$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T' E' \$$	$\$$	\Rightarrow Casamento: id
id + id	$E' \$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: +
		$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
		$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id		$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: +
id		$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +		$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id		$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: +
		$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id		$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *		$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id		$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +		$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id		$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: +
		$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id		$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *		$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id		$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +		$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id		$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: +
		$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id		$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *		$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id		$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +		$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id		$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

	Casamento	Pilha	Entrada	Ação
		$E\$$	$+id * + id\$$	
		$E\$$	$id * + id\$$	\Rightarrow Pular: +
		$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
		$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
		$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id		$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id		$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *		$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id		$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id		$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id		$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +		$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +		$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +		$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id		$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id		$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id		$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$

Parsing Preditivo Não-Recursivo

Tratamento de Erros – Exemplo

Não-Terminal	Símbolo de Entrada					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	sinc	sinc
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$	sinc		$T \rightarrow FT'$	sinc	sinc
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$	sinc	sinc	$F \rightarrow (E)$	sinc	sinc

Simulação:

Casamento	Pilha	Entrada	Ação
	$E\$$	$+id * + id\$$	
	$E\$$	$id * + id\$$	\Rightarrow Pular: +
	$TE'\$$	$id * + id\$$	Saída: $E \rightarrow TE'$
	$FT'E'\$$	$id * + id\$$	Saída: $T \rightarrow FT'$
	$idT'E'\$$	$id * + id\$$	Saída: $F \rightarrow id$
id	$T'E'\$$	$* + id\$$	\Rightarrow Casamento: id
id	$*FT'E'\$$	$* + id\$$	Saída: $T' \rightarrow *FT'$
id *	$FT'E'\$$	$+ id\$$	\Rightarrow Casamento: *
id	$T'E'\$$	$+ id\$$	\Rightarrow Desempilhar: F
id	$E'\$$	$+ id\$$	Saída: $T' \rightarrow \lambda$
id	$+TE'\$$	$+ id\$$	Saída: $E' \rightarrow +TE'$
id +	$TE'\$$	$id\$$	\Rightarrow Casamento: +
id +	$FT'E'\$$	$id\$$	Saída: $T \rightarrow FT'$
id +	$idT'E'\$$	$id\$$	Saída: $F \rightarrow id$
id + id	$T'E'\$$	$\$$	\Rightarrow Casamento: id
id + id	$E'\$$	$\$$	Saída: $T' \rightarrow \lambda$
id + id	$\$$	$\$$	Saída: $E' \rightarrow \lambda$