

**Terceira Lista de Exercícios – Análise Sintática Descendente**

1) Diga quais são as duas abordagens para análise sintática e qual a diferença entre elas.

*R. Há duas abordagens gerais para análise sintática*

- *Descendente ou top-down - constrói-se a árvore de derivação a partir do símbolo inicial da gramática (raiz da árvore), fazendo a árvore crescer até atingir suas folhas. Em cada passo, um lado esquerdo de uma produção é substituído por um lado direito (expansão). A análise é feita da raiz para as folhas. Parte-se do símbolo inicial da gramática e, por meio de derivações, chega-se à sequência de tokens retornada pelo analisador léxico.*
- *Ascendente ou bottom-up – a partir dos tokens do texto fonte (folhas da árvore de derivação) constrói-se a árvore até o símbolo inicial da gramática. Em cada passo, um lado direito de uma produção é substituído por um símbolo não-terminal (redução). A análise é feita das folhas para a raiz. Analisadores desse tipo são também conhecidos como analisadores de empilha-reduz (shift-reduce) e são, geralmente, gerados por geradores automáticos.*

2) Dada a gramática a seguir:

$\langle E \rangle ::= \langle E \rangle \vee \langle T \rangle \mid \langle T \rangle$   
 $\langle T \rangle ::= \langle T \rangle \& \langle F \rangle \mid \langle F \rangle$   
 $\langle F \rangle ::= \neg \langle F \rangle \mid id$

a) Ela é LL(1)? Se não, aplique as transformações necessárias para convertê-la para LL(1).

*R. Não é LL(1). É necessário remover a recursividade à esquerda:*

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$   
 $\langle E' \rangle ::= \vee \langle T \rangle \langle E' \rangle \mid \epsilon$   
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$   
 $\langle T' \rangle ::= \& \langle F \rangle \langle T' \rangle \mid \epsilon$   
 $\langle F \rangle ::= \neg \langle F \rangle \mid id$

b) Construa a tabela sintática correspondente à gramática (alterada na letra a) se for o caso).

*R.*

	<i>id</i>	$\vee$	$\&$	$\neg$	$\$$
<i>E</i>	$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$			$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$	
<i>E'</i>		$\langle E' \rangle ::= \vee \langle T \rangle \langle E' \rangle$			$\langle E' \rangle ::= \epsilon$
<i>T</i>	$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$			$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$	
<i>T'</i>		$\langle T' \rangle ::= \epsilon$	$\langle T' \rangle ::= \& \langle F \rangle \langle T' \rangle$		$\langle T' \rangle ::= \epsilon$
<i>F</i>	$\langle F \rangle ::= id$			$\langle F \rangle ::= \neg \langle F \rangle$	

*Para  $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$  tem-se que  $\text{Primeiro}(\langle T \rangle \langle E' \rangle) = \{\neg, id\}$   $M[E, \neg] = M[E, id] = \langle E \rangle ::= \langle T \rangle \langle E' \rangle$*

*Para  $\langle E' \rangle ::= \vee \langle T \rangle \langle E' \rangle$  tem-se que  $\text{Primeiro}(\vee \langle T \rangle \langle E' \rangle) = \{\vee\}$   $M[E, \vee] = M[E, id] = \langle E' \rangle ::= \vee \langle T \rangle \langle E' \rangle$*

*Para  $\langle E \rangle ::= \epsilon$  tem-se que  $\text{Seguidor}(\langle E' \rangle) = \{\$\}$   $M[E', \$] = \langle E' \rangle ::= \epsilon$*

3) Considere a gramática

$\langle \text{lexp} \rangle ::= \langle \text{átomo} \rangle \mid \langle \text{lista} \rangle$   
 $\langle \text{átomo} \rangle ::= \text{número} \mid \text{identificador}$   
 $\langle \text{lista} \rangle ::= ( \langle \text{lexp-seq} \rangle )$   
 $\langle \text{lexp-seq} \rangle ::= \langle \text{lexp-seq} \rangle \langle \text{lexp} \rangle \mid \langle \text{lexp} \rangle$

a) Remova a recursão à esquerda.

*R. Para remover a recursão imediata à esquerda simples, presente na gramática acima, é necessário converter as regras do tipo*

$A \rightarrow A C \mid B$

em

$A \rightarrow BA'$

$A \rightarrow CA' \mid \epsilon$

Assim, a gramática após a remoção da recursão imediata à esquerda fica:

$\langle \text{lexp} \rangle ::= \langle \text{átomo} \rangle \mid \langle \text{lista} \rangle$

$\langle \text{átomo} \rangle ::= \text{número} \mid \text{identificador}$

$\langle \text{lista} \rangle ::= ( \langle \text{lexp-seq} \rangle )$

$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$

$\langle \text{lexp-seq}' \rangle ::= \langle \text{lexp-seq} \rangle \mid \epsilon$

b) Construa os conjuntos Primeiros e Seguidores para os não-terminais da gramática resultante (letra a).

R.

$\text{Primeiro}(\text{lexp}) = \text{Primeiro}(\text{átomo}) + \text{Primeiro}(\text{lista}) = \{\text{número}, \text{identificador}, ( \}$

$\text{Primeiro}(\text{átomo}) = \{\text{número}, \text{identificador}\}$

$\text{Primeiro}(\text{lista}) = \{( \}$

$\text{Primeiro}(\text{lexp-seq}) = \text{Primeiro}(\text{lexp}) = \{\text{número}, \text{identificador}, ( \}$

$\text{Primeiro}(\text{lexp-seq}') = \text{Primeiro}(\text{lexp-seq}) + \epsilon = \{\text{número}, \text{identificador}, (, \epsilon\}$

$\text{Seguidor}(\text{lexp}) = \$ + \text{Primeiro}(\text{lexp-seq}') - \epsilon + \text{Seguidor}(\text{lexp-seq}') = \{\$, \text{número}, \text{identificador}, (, )\}$

$\text{Seguidor}(\text{átomo}) = \text{Seguidor}(\text{lexp}) = \{\$, \text{número}, \text{identificador}, (, )\}$

$\text{Seguidor}(\text{lista}) = \text{Seguidor}(\text{lexp}) = \{\$, \text{número}, \text{identificador}, (, )\}$

$\text{Seguidor}(\text{lexp-seq}) = ) + \text{Seguidor}(\text{lexp-seq}') = \{ ) \}$

$\text{Seguidor}(\text{lexp-seq}') = \text{Seguidor}(\text{lexp-seq}) = \{ ) \}$

c) Construa a tabela de análise sintática a ser usada por um método de ASD preditiva não recursiva, a partir da gramática resultante (letra a).

R.

	número	identificador	(	)	\$
$\langle \text{lexp} \rangle$	$\langle \text{lexp} \rangle ::= \langle \text{átomo} \rangle$	$\langle \text{lexp} \rangle ::= \langle \text{átomo} \rangle$	$\langle \text{lexp} \rangle ::= \langle \text{lista} \rangle$		
$\langle \text{átomo} \rangle$	$\langle \text{átomo} \rangle ::= \text{número}$	$\langle \text{átomo} \rangle ::= \text{identificador}$			
$\langle \text{lista} \rangle$			$\langle \text{lista} \rangle ::= ( \langle \text{lexp-seq} \rangle )$		
$\langle \text{lexp-seq} \rangle$	$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$	$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$	$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$		
$\langle \text{lexp-seq}' \rangle$	$\langle \text{lexp-seq}' \rangle ::= \langle \text{lexp-seq} \rangle$	$\langle \text{lexp-seq}' \rangle ::= \langle \text{lexp-seq} \rangle$	$\langle \text{lexp-seq}' \rangle ::= \langle \text{lexp-seq} \rangle$	$\langle \text{lexp-seq}' \rangle ::= \epsilon$	

d) Mostre as ações do analisador preditivo não recursivo correspondente (de acordo com a tabela criada na letra c) dada a cadeia de entrada **(a (b (2)) (c))**. OBS.: a, b e c são identificadores e 2, número.

R.

Pilha	Cadeia	Regra
$\$ \langle \text{lexp} \rangle$	$(a (b (2)) (c)) \$$	$\langle \text{lexp} \rangle ::= \langle \text{lista} \rangle$
$\$ \langle \text{lista} \rangle$	$(a (b (2)) (c)) \$$	$\langle \text{lista} \rangle ::= ( \langle \text{lexp-seq} \rangle )$
$\$ ) \langle \text{lexp-seq} \rangle ($	$(a (b (2)) (c)) \$$	---
$\$ ) \langle \text{lexp-seq} \rangle$	$a (b (2)) (c)) \$$	$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$
$\$ ) \langle \text{lexp-seq}' \rangle \langle \text{lexp} \rangle$	$a (b (2)) (c)) \$$	$\langle \text{lexp} \rangle ::= \langle \text{átomo} \rangle$
$\$ ) \langle \text{lexp-seq}' \rangle \langle \text{átomo} \rangle$	$a (b (2)) (c)) \$$	$\langle \text{átomo} \rangle ::= \text{identificador}$
$\$ ) \langle \text{lexp-seq}' \rangle \text{identificador}$	$a (b (2)) (c)) \$$	---
$\$ ) \langle \text{lexp-seq}' \rangle$	$(b (2)) (c)) \$$	$\langle \text{lexp-seq}' \rangle ::= \langle \text{lexp-seq} \rangle$
$\$ ) \langle \text{lexp-seq} \rangle$	$(b (2)) (c)) \$$	$\langle \text{lexp-seq} \rangle ::= \langle \text{lexp} \rangle \langle \text{lexp-seq}' \rangle$
$\$ ) \langle \text{lexp-seq}' \rangle \langle \text{lexp} \rangle$	$(b (2)) (c)) \$$	$\langle \text{lexp} \rangle ::= \langle \text{lista} \rangle$
$\$ ) \langle \text{lexp-seq}' \rangle \langle \text{lista} \rangle$	$(b (2)) (c)) \$$	$\langle \text{lista} \rangle ::= ( \langle \text{lexp-seq} \rangle )$
$\$ ) \langle \text{lexp-seq}' \rangle ) \langle \text{lexp-seq} \rangle ($	$(b (2)) (c)) \$$	---

\$)<lexp-seq'>)<lexp-seq>	b (2)) (c))\$	<lexp-seq> ::= <lexp> <lexp-seq'>
\$)<lexp-seq'>)<lexp-seq'><lexp>	b (2)) (c))\$	<lexp> ::= <átomo>
\$)<lexp-seq'>)<lexp-seq'><átomo>	b (2)) (c))\$	<átomo> ::= identificador
\$)<lexp-seq'>)<lexp-seq'>identificador	b (2)) (c))\$	---
\$)<lexp-seq'>)<lexp-seq'>	(2)) (c))\$	<lexp-seq'> ::= <lexp-seq>
\$)<lexp-seq'>)<lexp-seq>	(2)) (c))\$	<lexp-seq> ::= <lexp> <lexp-seq'>
\$)<lexp-seq'>)<lexp-seq'><lexp>	(2)) (c))\$	<lexp> ::= <lista>
\$)<lexp-seq'>)<lexp-seq'><lista>	(2)) (c))\$	<lista> ::= ( <lexp-seq> )
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq>(	(2)) (c))\$	---
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq>	2)) (c))\$	<lexp-seq> ::= <lexp> <lexp-seq'>
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq'><lexp>	2)) (c))\$	<lexp> ::= <átomo>
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq'><átomo>	2)) (c))\$	<átomo> ::= número
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq'>número	2)) (c))\$	---
\$)<lexp-seq'>)<lexp-seq'>)<lexp-seq'>	) (c))\$	<lexp-seq'> ::= $\epsilon$
\$)<lexp-seq'>)<lexp-seq'>)	) (c))\$	---
\$)<lexp-seq'>)<lexp-seq'>	) (c))\$	<lexp-seq'> ::= $\epsilon$
\$)<lexp-seq'>)	) (c))\$	---
\$)<lexp-seq'>	(c))\$	<lexp-seq'> ::= <lexp-seq>
\$)<lexp-seq>	(c))\$	<lexp-seq> ::= <lexp> <lexp-seq'>
\$)<lexp-seq'><lexp>	(c))\$	<lexp> ::= <lista>
\$)<lexp-seq'><lista>	(c))\$	<lista> ::= ( <lexp-seq> )
\$)<lexp-seq'>)<lexp-seq>(	(c))\$	---
\$)<lexp-seq'>)<lexp-seq>	c))\$	<lexp-seq> ::= <lexp> <lexp-seq'>
\$)<lexp-seq'>)<lexp-seq'><lexp>	c))\$	<lexp> ::= <átomo>
\$)<lexp-seq'>)<lexp-seq'><átomo>	c))\$	<átomo> ::= identificador
\$)<lexp-seq'>)<lexp-seq'>identificador	c))\$	---
\$)<lexp-seq'>)<lexp-seq'>	)\$	<lexp-seq'> ::= $\epsilon$
\$)<lexp-seq'>)	)\$	---
\$)<lexp-seq'>	)\$	<lexp-seq'> ::= $\epsilon$
\$)	)\$	---
\$	\$	<b>Aceita</b>

4) Dada a gramática a seguir

$\langle S \rangle ::= ( \langle L \rangle ) \mid a$

$\langle L \rangle ::= \langle L \rangle , \langle S \rangle \mid \langle S \rangle$

a) Elimine a recursividade à esquerda.

R. Para remover a recursão imediata à esquerda simples, presente na gramática acima, é necessário converter as regras do tipo

$A \rightarrow A C \mid B$

em

$A \rightarrow BA'$

$A \rightarrow CA' \mid \epsilon$

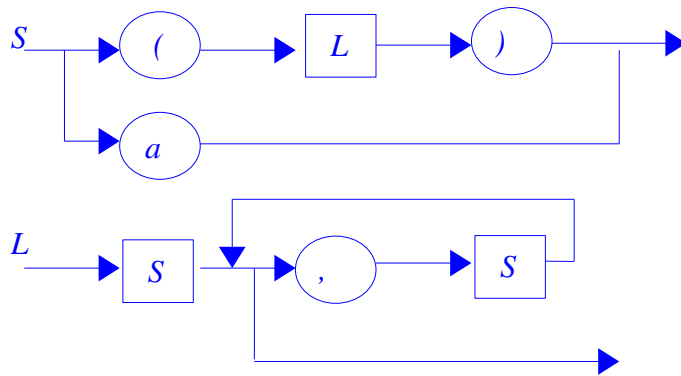
Assim, a gramática após a remoção da recursão imediata à esquerda fica:

$\langle S \rangle ::= ( \langle L \rangle ) \mid a$

$\langle L \rangle ::= \langle S \rangle \langle L' \rangle$

$\langle L' \rangle ::= , \langle S \rangle \langle L' \rangle \mid \epsilon$

b) Desenhe os grafos sintáticos correspondentes (em número reduzido).



c) Construa os procedimentos recursivos para os grafos sintáticos construídos na letra b) bem como o programa principal.

5) Qual a diferença entre a análise sintática descendente preditiva recursiva e a não recursiva? Por que são chamadas 'preditivas'?

*R. A análise sintática descendente preditiva recursiva se baseia em procedimentos possivelmente recursivos, um para cada não-terminal; enquanto a análise sintática descendente preditiva não recursiva utiliza uma pilha explícita e tabela sintática para governar o processo de análise. Elas recebem o nome de "preditivas" porque "são capazes" de olhar um passo à frente para determinar qual regra de produção deve ser aplicada num determinado momento. Assim, com base no conjunto de Primeiros em gramáticas LL(1), elas conseguem verificar 1 símbolo a frente e determinar inequivocamente qual é a regra de produção a ser aplicada.*

6) Considere a gramática abaixo:

$$\begin{aligned} \langle E \rangle &::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle \\ \langle T \rangle &::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle \\ \langle F \rangle &::= ( \langle E \rangle ) \mid id \end{aligned}$$

a) Verifique se é LL(1). Se não for, transforme-a.

*R. A gramática será LL(1) se:*

- não for ambígua
- não for recursiva à esquerda
- estiver fatorada à esquerda

*Assim, a gramática acima não é LL(1), pois é preciso remover a recursão à esquerda da primeira e da segunda regra de produção:*

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

b) Faça passo-a-passo (como no exercício 3) a análise sintática descendente preditiva não recursiva para a cadeia **id\*id+id**.

*R. É necessário construir a tabela e mostrar passo-a-passo os estados da pilha, cadeia e qual a regra aplicada.*

7) Construa a tabela sintática para a gramática

$$\begin{aligned} \langle \text{declaração} \rangle &::= \langle \text{if-decl} \rangle \mid \text{outra} \\ \langle \text{if-decl} \rangle &::= \text{if} ( \langle \text{exp} \rangle ) \langle \text{declaração} \rangle \langle \text{else-parte} \rangle \\ \langle \text{else-parte} \rangle &::= \text{else} \langle \text{declaração} \rangle \mid \epsilon \\ \langle \text{exp} \rangle &::= 0 \mid 1 \end{aligned}$$

*R.*

*Primeiro(<declaração>) = { if, outra }*  
*Primeiro(<if-decl>) = { if }*  
*Primeiro(<else-parte>) = { else, ε }*

*Seguidor(<declaração>) = { \$, else }*  
*Seguidor(<if-decl>) = { \$, else }*  
*Seguidor(<else-parte>) = { \$, else }*

$Primeiro(<exp>) = \{ 0, 1 \}$

$Seguidor(<exp>) = \{ \ ) \}$

*Não dá para construir a tabela*