

SCC-5832 - TEORIA DA COMPUTAÇÃO - 2012

**Lista de Exercícios – Capítulo 2**

1. Mostre que a linguagem  $\{a^n b^n c^n \mid n > 0\}$  não é livre de contexto.
2. Seja  $G = (\Sigma, V, S, P)$  onde  $\Sigma = \{a, b\}$ ,  $V = \{S\}$  e  $P$  é o seguinte conjunto de produções:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow aSa \\ S &\rightarrow bSa \\ S &\rightarrow bSb \\ S &\rightarrow \lambda \end{aligned}$$

Mostre que  $L(G)$  é uma LLD.

3. Suponha que  $G = (\Sigma, V, S, P)$  seja uma GLC tal que cada produção em  $P$  ou é da forma  $A \rightarrow wB$  ou da forma  $A \rightarrow w$ , onde  $A, B \in V$  e  $w \in \Sigma^*$ .  $L(G)$  é necessariamente uma LLD? Prove esta afirmação ou ache um contra-exemplo.
4. A forma normal de Chomsky diz que “qualquer linguagem livre de contexto (LLC) pode ser gerada por uma gramática na qual todas as produções são da forma  $A \rightarrow BC$  ou  $A \rightarrow a$ , onde  $A, B$  e  $C$  são variáveis e  $a$  é um terminal.” Considere a gramática

$$G = (\{a, b, +, -, *, /, (, )\}, \{S, T, L\}, S, P),$$

onde  $P$  consiste das produções:

$$\begin{aligned} S &\rightarrow T + S \mid T - S \mid T \\ T &\rightarrow L * T \mid L / T \mid L \\ L &\rightarrow (S) \mid a \mid b \end{aligned}$$

Ache uma gramática na forma normal de Chomsky que gere  $L(G)$ .

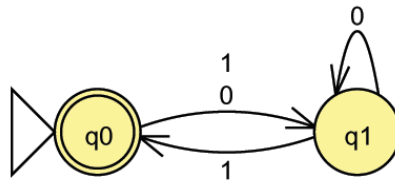
5. Dê um APN que aceite a linguagem dos parênteses casados pelo estado final.
6. Dê um APN que aceite pelo estado final a linguagem gerada pela GLC:

$$S \rightarrow aAA, A \rightarrow aS \mid bS \mid a$$

7. Converta a seguinte gramática à Forma Normal de Chomsky.

$$\begin{aligned} S &\rightarrow 00A \mid B \mid 1 \\ A &\rightarrow 1AA \mid 2 \\ B &\rightarrow 0 \end{aligned}$$

8. Dê um APN de um estado que aceita a linguagem  $\{wcw^R \mid w \in (a + b)^*\}$ .
9. Dê um APN (autômato de pilha) que aceite a linguagem  $\{ww^R \mid w \in \{0, 1\}^*\}$  pela pilha vazia. Teste para 011110.
10. Considere a seguinte gramática sobre  $\Sigma = \{a, b\}$ , com símbolo inicial S:
 
$$S \rightarrow bA \mid aB; A \rightarrow a \mid aS \mid bAA; B \rightarrow b \mid bS \mid aBB$$
 ache uma gramática equivalente na forma normal de Chomsky.
11. Escreva a gramática para a linguagem com cadeias que contenham um único  $a$  a esquerda e  $n$   $b$ 's a direita:  $ab^n$ ,  $n > 0$ . Qual é o tipo desta linguagem?
12. Construa uma gramática livre de contexto  $G$  para a linguagem  $L(G) = \{w = xayb \mid x, y \in \{a, b\}^*\}$ .
13. Mostre que para qualquer linguagem livre de contexto  $L$ ,  $L - \{\lambda\}$  também é livre de contexto.
14. Considere uma gramática  $G = (\Sigma, V, S, P)$ , onde  $\Sigma = \{0, 1\}$ ,  $V = \{S\}$ ,  $P = \{S \rightarrow 0S1, S \rightarrow 01\}$ . Qual o tipo desta gramática segundo a hierarquia de Chomsky? Dê a descrição formal da linguagem gerada por esta gramática. Se for possível, descreva o autômato finito, com o menor número de estados possível, que aceite esta linguagem.
15. Considere a seguinte linguagem livre de contexto  $L = \{0^n1^n \mid n \geq 1\}$ . Escreva um APN (autômato de pilha)  $M$  que processe esta linguagem. Verifique como  $M$  age com as entradas 01 e 011.
16. Dê um APN (autômato de pilha) de um estado que aceite a linguagem  $\{ww^R \mid w \in (a + b)^*\}$  pela pilha vazia. Dê a gramática correspondente. Qual é o tipo desta gramática?
17. Considere uma gramática  $G = (\Sigma, V, S, P)$ , onde  $\Sigma = \{0, 1\}$ ,  $V = \{S\}$ ,  $P = \{S \rightarrow 0S1, S \rightarrow 01\}$ . Qual é o APN (autômato de pilha) equivalente a esta gramática?
18. Como um autômato de pilha (APN) é constituído de uma pilha *last-in first-out* e uma máquina de estados, deve ser possível converter um autômato finito (AFD) em um APN ignorando a pilha. Construa, se possível, um APN a partir do seguinte AFD:



19. Considere a linguagem  $L = \{ w \mid w \in (a + b)^* \text{ com número par de } a\text{'s} \}$ . Por exemplo, a cadeia *abbabaa* seria aceita, enquanto que a cadeia *baabba* não.

- Se possível, escreva um autômato de pilha (APN) que processe  $L$ . Caso não seja possível, explique o porquê.
- Qual é o tipo de  $L$ ? Comente a sua resposta.

20. Considere a gramática  $G = (\{a, b\}, \{S, A, B\}, S, P)$ , onde  $P$  é o conjunto de produções:

$$\begin{aligned} S &\rightarrow aAa \mid bBb \\ A &\rightarrow b \\ B &\rightarrow aA \end{aligned}$$

- Qual é o tipo de menor complexidade de  $G$ ?
- Qual é o tipo de menor complexidade de  $L(G)$ ?
- Ache uma gramática na Forma Normal de Greibach para  $G$ , se possível. Se não for possível, explique o porquê.
- Ache o autômato finito que processe  $L(G)$ , se possível. Se não for possível, explique o porquê.
- Ache o autômato de pilha de um estado que processe  $L(G)$ , se possível. Se não for possível, explique o porquê.

21. Seja o seguinte autômato finito  $(\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$ :

$\delta$	0	1
$q_0$	$q_1$	$q_1$
$q_1$	$q_1$	$q_0$

Escreva o autômato de pilha equivalente. Se não for possível, explique o porquê.

22. Seja o seguinte conjunto de produções da gramática livre de contexto  $G$ :

$$\begin{aligned} S &\rightarrow aaZcc \\ Z &\rightarrow aZc \\ Z &\rightarrow b \end{aligned}$$

- Qual é a linguagem que esta gramática gera?
- $L(G)$  é regular?

Observe agora o seguinte conjunto de produções da gramática linear a direita  $G'$ :

$$\begin{aligned} S &\rightarrow aA \\ A &\rightarrow aB \\ B &\rightarrow aB \mid bC \\ C &\rightarrow cC \mid cD \\ D &\rightarrow c \end{aligned}$$

- c) Qual é a relação entre  $G$  e  $G'$ ? São equivalentes? Por que?  
d) Escreva o autômato a pilha que processa  $L(G)$ .

23. Considerando que:

- toda linguagem livre de contexto também é sensível ao contexto;
- nem toda gramática livre de contexto é sensível ao contexto;
- há quatro tipos de gramáticas/linguagens segundo Chomsky,

as gramáticas abaixo são livres de contexto? São sensíveis ao contexto? Por que?

- a)  $G = (X, V, S, P)$ , onde  $X = \{a, b\}$ ,  $V = \{A, B\}$ ,  $S = A$ ,  $P = \{ A \rightarrow Ba, B \rightarrow BB, Aa \rightarrow Bb, B \rightarrow b, B \rightarrow bA, A \rightarrow a, Ab \rightarrow \lambda \}$   
b)  $G = (X, V, S, P)$ , onde  $X = \{0, 1\}$ ,  $V = \{S\}$ ,  $P = \{S \rightarrow 0S1, S \rightarrow \lambda \}$

Que linguagem é gerada pela gramática do item b?

24. Escreva a gramática para a linguagem com cadeias que contenham um único  $a$  a esquerda e  $n$   $b$ 's a direita:  $ab^n$ ,  $n > 0$ . Qual é o tipo desta linguagem?

25. Construa uma gramática livre de contexto  $G$  para a linguagem  $L(G) = \{w = xayb \mid x, y \in \{a, b\}^*\}$ .

26. Seja  $G = (\{0, 1\}, \{A, B\}, A, \{ A \rightarrow B0, B \rightarrow BB, A0 \rightarrow B1, B \rightarrow 1, B \rightarrow 1A, A \rightarrow 0, A1 \rightarrow \lambda \})$ . Como você descreveria essa gramática? Essa gramática é livre de contexto? Por que? Essa gramática é sensível ao contexto? Por que?

27. Construa gramáticas que gerem as seguintes linguagens:

- (a)  $\{a^i b^j \mid j = i\}$   
(b)  $\{a^i b^j \mid j = 2i\}$   
(c)  $\{a^i b^j \mid j = i \text{ ou } j = 2i\}$   
(d)  $\{a^i b^j \mid j \neq i\}$   
(e)  $\{a^i b^j \mid j < i \text{ ou } j > 2i\}$   
(f)  $\{a^i b^j \mid i \leq j \leq 2i\}$   
(g)  $\{a^i b^j c^k \mid i = j \text{ ou } j = k \text{ ou } k = i\}$

28. Seja o seguinte trecho de um programa em C:

```

do
{
    switch (entrada )
    {
        case 'A':
        case 'a': cont_a++; break;
    }
}
while ( int(entrada) != 27);

```

- construa uma gramática livre de contexto realista que gere esta cadeia;
- realize o parsing *top-down* da cadeia;
- realize o parsing *bottom-up* da cadeia.

29. Descreva um *parse* “bottom-up” da cadeia  $2*(2+2)$  usando a seguinte gramática:

O símbolo inicial para a gramática é  $E$ . As regras de produção são:

- $E \rightarrow E + T$
- $E \rightarrow T$
- $T \rightarrow T * F$
- $T \rightarrow F$
- $F \rightarrow (E)$
- $F \rightarrow 2$

30. Descreva um *parse* “top-down” do texto de programa

```

begin while x ≠ y do
    begin x := 0;
        y := 0
    end
end

```

usando a gramática:

$\Sigma = \{\text{begin, end, pred, succ, :=, } \neq, \text{ while, do, }, (, ), 0, x, y\}$

$V = \{C, S, S_1, S_2, A, W, U, T\}$

Símbolo Inicial =  $C$

Produções =  $C \rightarrow \text{begin } S_1 \text{ end } \{C \text{ para comando composto}\}$

$S_1 \rightarrow SS_2$

$S_2 \rightarrow ;S_1 \mid \lambda$

$S \rightarrow A \mid W \mid C$

$A \rightarrow V := T \{A \text{ para comando de atribuição}\}$

$T \rightarrow \text{pred}(V) \mid \text{succ}(V) \mid 0$

$W \rightarrow \text{while } V \neq V \text{ do } S \{W \text{ para comando while}\}$

$V \rightarrow x \mid y$

31. Desenhe uma árvore de *parsing* (análise) para a sentença “A bola está sobre a grande mesa branca.”
32. Usando o algoritmo *bottom-up* descreva como você poderia realizar o *parse* da sentença do exercício anterior.
33. Considere a gramática

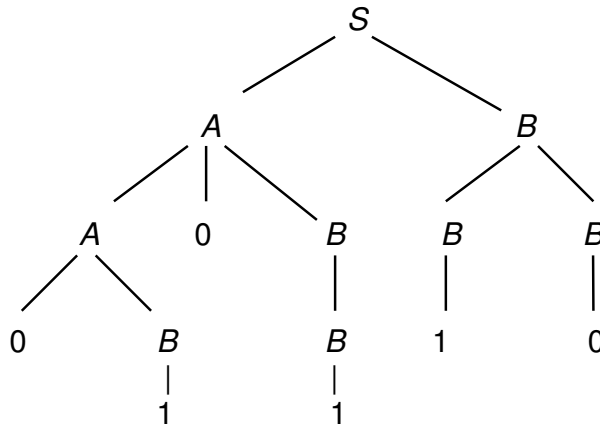
$$G = (\{a, b, +, -, *, /, (, )\}, \{S, T, L\}, S, P),$$

onde  $P$  consiste das produções:

$$\begin{aligned} S &\rightarrow T + S \mid T - S \mid T \\ T &\rightarrow L * T \mid L / T \mid L \\ L &\rightarrow (S) \mid a \mid b \end{aligned}$$

Informalmente descreva  $L(G)$ . Depois descreva um *parse* da cadeia  $((a + b) / b) - a$  usando a técnica “bottom-up” para a gramática acima.

34. Suponha que  $G$  é uma gramática livre de contexto (GLC) a partir da qual a cadeia 010110 pode ser derivada. Suponha que a árvore de derivação para 010110, dada na figura abaixo, inclui todas as produções de  $G$ . Mostre que  $G$  é ambígua.



35. Seja o seguinte trecho de um programa em C:

```
if (ano % 100 == 0)
{
    if (ano % 400 == 0)
        exit(0);
    bis = 1;
}
```

Construa uma gramática livre de contexto que gere esta cadeia. Realize o parsing *top-down* e o parsing *bottom-up* desta cadeia.

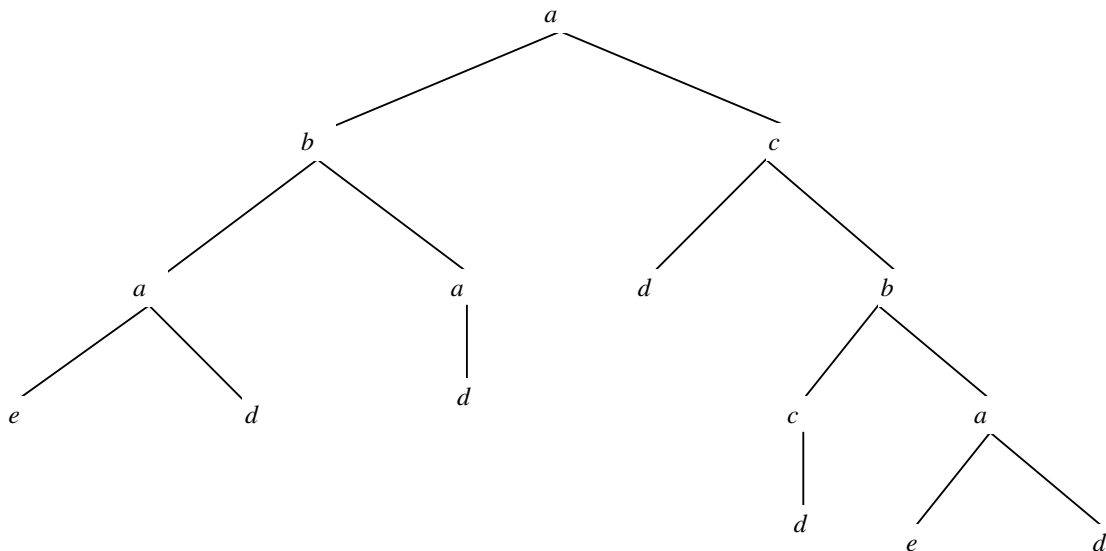
36. Construa uma gramática livre de contexto que produza um fragmento não trivial da língua portuguesa. Dê um exemplo de uma sentença (usando apenas as palavras introduzidas como terminais) que não é produzida pela sua gramática.

37. Seja o seguinte trecho de um programa em C:

```
do
{
    switch (entrada )
    {
        case 'A':
        case 'a': cont_a++; break;
    }
}
while ( int(entrada) != 27);
```

- construa uma gramática livre de contexto realista que gere esta cadeia;
- realize o parsing *top-down* da cadeia;
- realize o parsing *bottom-up* da cadeia.

38. Seja a seguinte figura:



Esqueça a nossa convenção de que letras minúsculas denotam símbolos terminais e letras maiúsculas denotam variáveis. Esta figura é uma árvore de derivação de alguma gramática livre de contexto,  $G = (\Sigma, V, S, P)$ , para a qual as produções e símbolos não são conhecidos. Responda: **a)** Qual é a cadeia

extraída desta árvore? **b)** Que símbolos devem estar em  $\Sigma$ ? **c)** Que símbolos estão necessariamente em  $V$ ? **d)** Você acha que  $d$  e  $e$  devem estar em  $\Sigma$ , ou eles poderiam estar em  $V$ ? **e)** Que produções devem estar em  $P$ ? **f)** Mostre que a cadeia  $deddedd$  pertence a  $L(G)$  usando o *parsing bottom-up*.

39. Seja o seguinte trecho de um programa em C:

```
if (ano % 100 == 0)
{
    if (ano % 400 == 0)
        exit(0);
    bis = 1;
}
```

Construa uma gramática livre de contexto que gere esta cadeia. Realize o *parsing top-down* e o *parsing bottom-up* desta cadeia.

40. Construa uma gramática livre de contexto que produza um fragmento não trivial da língua portuguesa. Dê um exemplo de uma sentença (usando apenas as palavras introduzidas como terminais) que não é produzida pela sua gramática.