

UNIVERSIDADE FEDERAL DE LAVRAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Curso: BCC/BSI

GCC122 – Ling. Formais e Autômatos

Professor: Ricardo Terra

Pontuação: 3.33 pontos [30 questão(ões)]

**REVISÃO P2**

Data: Antes P2

INFORMAÇÕES SOBRE REVISÃO P2:

1. A atividade é *individual*.
2. Cópias (total ou parcial) serão devidamente *penalizadas*.
3. Todo semestre as questões são levemente modificadas e, caso seja constatada a resposta de uma questão diferente do enunciado do semestre vigente, a lista será desconsiderada (i.e., nota 0).
4. A entrega **DEVE** ser *manuscrita* e entregue de forma *virtual* por meio da submissão de um único arquivo PDF até a data/hora limite especificada no Moodle. Mais importante, **deve** constar o nome do aluno à caneta em toda lauda e os exercícios **devem** estar legíveis e ordenados. Caso contrário, a lista não será considerada.
5. Não serão aceitas colagens de questões. Devem ser escaneadas laudas completas tal como uma folha de caderno.
6. O critério de correção é simples: será analisada completude (25%) e serão corrigidas normalmente três questões (25% cada). Isto indica que, por exemplo, caso o aluno tenha feito metade das questões da avaliação porém nenhuma daquelas a serem corrigidas pelo professor, a nota do aluno será de apenas 12,5%.
7. A escolha das questões a serem corrigidas pelo professor será arbitrária, porém igual para todos os discentes.

*Linguagens e Gramáticas Livres de Contexto*

1. [Bla11] Qual a importância do estudo das linguagens livre de contexto?
2. [Bla11] Desenvolva gramáticas livres do contexto que gerem as seguintes linguagens:
  - (a)  $L_a = \{w \mid w \text{ é palavra composta por } \{x, y, (, )\}^* \text{ com parênteses balanceados}\}$
  - (b)  $L_b = \{a^i b^j c^k \mid i = j \text{ ou } j = k \text{ e } i, j, k \geq 0\}$

*Recursividade*

3. Dada a gramática G, defina quais de suas regras tem recursão, recursão à esquerda, e recursão à direita:  
 $G : S \rightarrow BSA \mid A$   
 $A \rightarrow aA \mid \lambda$   
 $B \rightarrow Bba \mid \lambda$

*Ambiguidade e Árvore de Derivação*

4. [Bla11] Considere a seguinte gramática:  $G = (\{S\}, \{a, b\}, P, S)$  na qual:  $P = S \rightarrow SS \mid aSa \mid bSb \mid \lambda$   
Prove a ambiguidade para a palavra  $aabbaaaa$ :
  - (a) Construa duas árvores de derivação diferentes.
  - (b) Mostre duas derivações mais à esquerda diferentes.
  - (c) Mostre duas derivações mais à direita diferentes.
5. Converta as GLC da questão 2 para BNF.

*Formas Normais*

6. [Bla11] Por que ao tentar remover os símbolos inúteis se aplicarmos o “*reach*” antes do “*term*” temos um resultado diferente?
7. [Bla11] Quais algoritmos de simplificação apenas excluem regras da gramática original? E quais modificam a gramática original, mas não apenas excluindo regras?
8. [Bla11] Uma gramática pode simultaneamente estar na forma normal de Chomsky e de Greibach? Justifique e apresente um exemplo.

*Remoção de recursividade no símbolo inicial e eliminação de regras  $\lambda$*

9. [Sud05] Construa a gramática equivalente a G, essencialmente não contrátil, e sem recursão para o símbolo inicial. Descreva **cada** transformação realizada.

- (a)  $G : S \rightarrow aS \mid bS \mid B$   
 $B \rightarrow bb \mid C \mid \lambda$   
 $C \rightarrow cC \mid \lambda$
- (b)  $G : S \rightarrow BSA \mid A$   
 $A \rightarrow aA \mid \lambda$   
 $B \rightarrow Bba \mid \lambda$

*Eliminação de regra da cadeia*

10. [Sud05] Construa a gramática equivalente a  $G$ , sem regras de cadeia. Descreva **cada** transformação realizada.

- (a)  $G : S \rightarrow A \mid B \mid C$   
 $A \rightarrow aa \mid B$   
 $B \rightarrow bb \mid C$   
 $C \rightarrow cc$
- (b)  $G : S \rightarrow AS \mid A$   
 $A \rightarrow aA \mid bB \mid C$   
 $B \rightarrow bB \mid b$   
 $C \rightarrow cC \mid B$

*Remoção de símbolos inúteis*

11. [Sud05] Aplique os algoritmos *TERM* e *REACH* e construa a gramática equivalente a  $G$ . Descreva **cada** transformação realizada.

- (a)  $G : S \rightarrow AA \mid CD \mid bB$   
 $A \rightarrow aA \mid a$   
 $B \rightarrow bB \mid bC$   
 $C \rightarrow cB$   
 $D \rightarrow dD \mid d$
- (b)  $G : S \rightarrow aA \mid BD$   
 $A \rightarrow aA \mid aAB \mid aD$   
 $B \rightarrow aB \mid aC \mid BF$   
 $C \rightarrow Bb \mid aAC \mid aE$   
 $D \rightarrow bD \mid bC \mid b$   
 $E \rightarrow aB \mid bC$   
 $F \rightarrow aF \mid aG \mid a$   
 $G \rightarrow a \mid b$

*Forma Normal de Chomsky*

12. [Sud05] Converta a gramática  $G$  para a Forma Normal de Chomsky (as gramáticas já estão sem recursividade no símbolo inicial, sem regras de cadeia, sem regras inúteis e são essencialmente não-contráteis).

- (a)  $G : S \rightarrow aA \mid ABa$   
 $A \rightarrow AA \mid a$   
 $B \rightarrow AbB \mid bb$
- (b)  $G : S \rightarrow aAbB \mid ABC \mid a$   
 $A \rightarrow aA \mid a$   
 $B \rightarrow bBcC \mid bC$   
 $C \rightarrow abc$

13. [Sud05] Converta a gramática  $G$  para a Forma Normal de Chomsky. Descreva **cada** procedimento seguido para chegar à FNC.

- $G : S \rightarrow A \mid ABa \mid AbA$   
 $A \rightarrow Aa \mid \lambda$   
 $B \rightarrow Bb \mid BC$   
 $C \rightarrow CB \mid CA \mid bB$

*Remoção de Recursividade à esquerda*

14. [Bla11] Explique, em palavras, como recursões diretas à esquerda de GLCs podem ser substituídas por recursões à direita.

### Forma Normal de Greibach

15. [Sud05] Converta a gramática abaixo da Forma Normal de Chomsky para a Forma Normal de Greibach. Descreva **cada** transformação realizada.

$$G : S \rightarrow AB \mid BC$$

$$A \rightarrow AB \mid a$$

$$B \rightarrow AA \mid CB \mid b$$

$$C \rightarrow a \mid b$$

16. [Sud05] Construa a Gramática na Forma Normal de Greibach equivalente. Descreva **cada** procedimento seguido para chegar à FNG.

(a)  $G : S \rightarrow Sa \mid bA$

$$A \rightarrow Ab \mid Bc$$

$$B \rightarrow Aa \mid b$$

(b)  $G : S \rightarrow aAb \mid a$

$$A \rightarrow SS \mid b$$

(c) [Sud05]  $G : S \rightarrow A \mid B$

$$A \rightarrow AAA \mid a \mid B$$

$$B \rightarrow BBb \mid b$$

### Autômatos com Pilha

17. [Sud05] Construa o menor Autômato com pilha que aceite a seguinte linguagem
- (a)  $L_a = \{a^i b^j \mid i \geq 0 \text{ e } j \geq i\}$
- (b)  $L_b = \{a^i c^j b^i \mid i, j \geq 0\}$
- (c)  $L_c = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ e } (i = j \text{ ou } j = k)\}$
18. [Sud05] Construa o menor Autômato com pilha de apenas dois estados que aceite a linguagem  $\{wdw^R \mid w \in \{a, b, c\}^*\}$ .

### Variantes

19. [Sud05] Dado  $L = \{a^{2i} b^{3i} \mid i \geq 0\}$
- (a) Construa o menor autômato com pilha que aceite a linguagem  $L$ .
- (b) Construa o menor autômato com pilha atômico que aceite a linguagem  $L$ .
- (c) Construa o menor autômato com pilha estendido que aceite a linguagem  $L$ .

### Critérios de Aceitação

20. Construa o menor Autômato com pilha cujo critério de aceitação seja por estado final para a seguinte linguagem  $L = \{a^i b^j \mid i > j\}$ .
21. Explique porque, independentemente do critério de aceitação, todos os Autômatos com Pilha possuem mesmo poder computacional.

### Algoritmos de Reconhecimento

22. Diferencie os algoritmos “Autômato com Pilha como Reconhecedor” e “Autômato com Pilha Descendente”.
23. Sobre *Top-down* e *Bottom-up*, responda:
- (a) Qual a diferença entre esses algoritmos em determinar se uma palavra  $p$  pertence a uma determinada linguagem?
- (b) Defina uma GLC  $G$  e uma palavra  $w \in L(G)$ , e mostre como seria o processo de derivação utilizando cada uma das abordagens.

(c) Represente as duas árvores de derivação do item (b). O que pode se concluir?

24. Usando o Algoritmo CYK, demonstre que

(a) a palavra *aaabbb* pertence (ou não) à linguagem abaixo:

$$S \rightarrow AT \mid AB$$

$$T \rightarrow XB$$

$$X \rightarrow AT \mid AB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

(b) a palavra *she eats a fish with a fork* pertence (ou não) à linguagem abaixo:

$$S \rightarrow NP VP$$

$$VP \rightarrow VP PP \mid V NP \mid \mathbf{eats}$$

$$PP \rightarrow P NP$$

$$NP \rightarrow DET N \mid \mathbf{she}$$

$$V \rightarrow \mathbf{eats}$$

$$P \rightarrow \mathbf{with}$$

$$N \rightarrow \mathbf{fish} \mid \mathbf{fork}$$

$$DET \rightarrow \mathbf{a}$$

25. Qual a complexidade dos algoritmos de reconhecimento abordados na disciplina? E qual seria uma complexidade viável no desenvolvimento de um compilador?

#### *Transformação de gramáticas para autômato com pilha*

26. [Sud05] Com base no algoritmo estudado, construa o autômato com pilha que aceite a linguagem da gramática na Forma Normal de Greibach a seguir:

$$G : S \rightarrow aABA \mid aBB$$

$$A \rightarrow bA \mid b$$

$$B \rightarrow cB \mid c$$

#### *Propriedades*

27. Linguagens de Programação são LLC?

28. Prove que LLCs são fechadas para União, Concatenação e Kleene.

29. Prove que LLCs não são fechadas para Interseção e Complemento.

30. Como demonstrar que um LLC é finita ou infinita?

## Referências

- [Bla11] Blauth Menezes, P. F. *Linguagens formais e autômatos*. 6ª ed. Vol. 3. Bookman, 2011.
- [Sud05] Sudkamp, T. A. *Languages and machines: an introduction to the theory of Computer Science*. 3ª ed. Addison-Wesley, 2005.