UFLA – Universidade Federal de Lavras
Departamento de Ciência da Computação
COM162 – Linguagens Formais e Autômatos
Prof. Rudini Sampaio
Monitor: Rodrigo Pereira dos Santos

Segunda Lista de Exercícios – **2004/2**

........................................................................................................................................................

### Exercício 1

**a.** Mostre que a linguagem $A = \{a\, b^n c^n \mid n \geq 0\}$ não é regular.

RESPOSTA = Por contradição, suponha que A é regular. Escolha $s = ab^pc^p$, onde p é de acordo com o lema da iteração.

Como $s \in A$ e $|s| \geq p$, podemos escrever s = xyz, onde para cada $i \geq 0$, $xy^iz \in A$. Consideraremos os seguintes casos:

1 – y consiste apenas de um tipo de símbolo (ou a, ou b, ou c). Neste caso, xyyz tem mais deste tipo de símbolo do que dos demais. Logo xyyz NÃO pertence a A.
2 – y contém a. Neste caso, xyyz tem mais de um símbolo a. Logo, xyyz NÃO pertence a A.
3 – y consiste de b's e c's. Neste caso, xyyz pode até ter o mesmo número de b's e c's, mas estarão fora de ordem. Logo, xyyz NÃO pertence a A.

Contradição. Logo, A não é regular.

■

**b.** Mostre que a linguagem $B = \{a^i b^j c^k \mid i,j,k \geq 0$ e se $i=1$ então $j=k\}$ não é regular.

RESPOSTA = Por contradição, suponha que B é regular. Neste caso, não vamos utilizar o lema da iteração e sim o fato de que a classe de linguagens regulares é fechada sob interseção.

Se B é regular, então $B \cap \{ab^*c^*\} = \{ab^nc^n, n \geq 0\}$, já que as linguagens regulares s ão fechadas sob interseção, e $ab^*c^*$ é regular. Como $A = \{ab^nc^n, n \geq 0\}$ não é regular, contradição.

■

**c.** Mostre que a linguagem $B$ satisfaz as três condições do lema da iteração.

RESPOSTA = Vamos mostrar que B satisfaz as 3 condições do lema da iteração. Para isso, tome para todo $s \in B$, $|s| \geq p$ e p = 3. Vamos dividir s = xyz tal que $s \in \{a^ib^jc^k \mid i,j,k \geq 0$ *e se* $i = 1$ *então* $j = k\}$:

1 – se i > 0. Nesse caso, vamos considerar que se i é par, x = ε, y = aa e z é o restante de s; se i é ímpar, x = ε, y = a e z é o restante de s.
2 – se i = 0 e j > 0. Nesse caso, x = ε, y = b, e a é o restante de s.
3 – se i = 0, j = 0 e k > 0. Neste caso, x = ε, y = c e z é o restante de s.

Logo, $xy^iz \in B$ $(i \geq 0)$. CONDIÇÃO 1 é satisfeita.
Como $|y| > 0$ em quaisquer casos, CONDIÇÃO 2 é satisfeita.
Como p = 3, $|xy| \leq p = 3$. CONDIÇÃO 3 é satisfeita.

Logo, B satisfaz as TRÊS condições do lema da iteração.

■

**d.** Explique porque isso não c ontradiz o lema da iteração.

RESPOSTA = Isso se deve ao fato do lema da iteração ser do tipo *se...então*. Logo, podemos dizer apenas que se uma linguagem é regular, então são satisfeitas as condições do lema, e não que se são satisfeitas as mesmas, a linguagem é regular.

........................................................................................................................................

**Exercício 2** Descreva gramáticas Livres de Contexto que geram as seguintes linguagens, todas sobre o alfabeto {0, 1}.

**a.** {*w* | *w* contém pelo menos três 0s}

RESPOSTA = G = ({S, A}, {0, 1}, R, S), onde:

- R: S -> A0A0A0A
  A -> A0 | A1 | ε

**b.** {*w* | |*w*| é ímpar e o símbolo do meio é o 1}

RESPOSTA = G = ({S, A}, {0, 1}, R, S), onde:

- R: S -> ASA
  A -> 0 | 1

**c.** $\{0^r1^s0^t \mid r, s, t \geq 0 \text{ e } s = r+t\}$

RESPOSTA = G = ({S, A, B}, {0, 1}, R, S), onde:

- R: S -> AB
  A -> 0A1 | ε
  B -> 1B0 | ε

Obs) $0^r1^s0^t = 0^r1^{r+t}0^t = 0^r1^r1^t0^t$

**d.** {*w* | $w = w^R$, isto é, *w* é um palíndromo}}

RESPOSTA = G = ({S}, {0, 1}, R, S), onde:

- R: S -> 0S0 | 1S1 | 0 | 1 | ε

**e.** {*w* | o número de 0s em *w* é o dobro do número de 1s}

RESPOSTA = G = ({S}, {0, 1}, R, S), onde:

- R: S -> 1S0S0 | 0S1S0 | 0S0S1 | ε

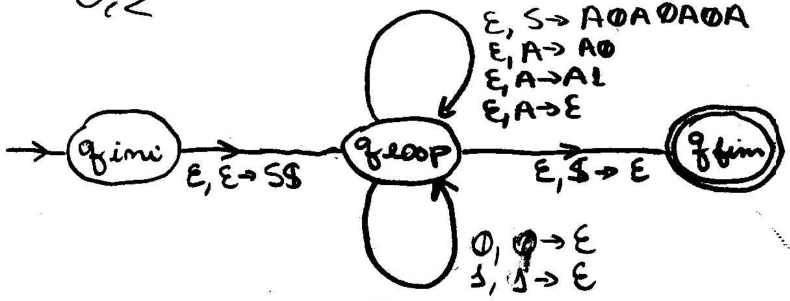
........................................................................................................................................

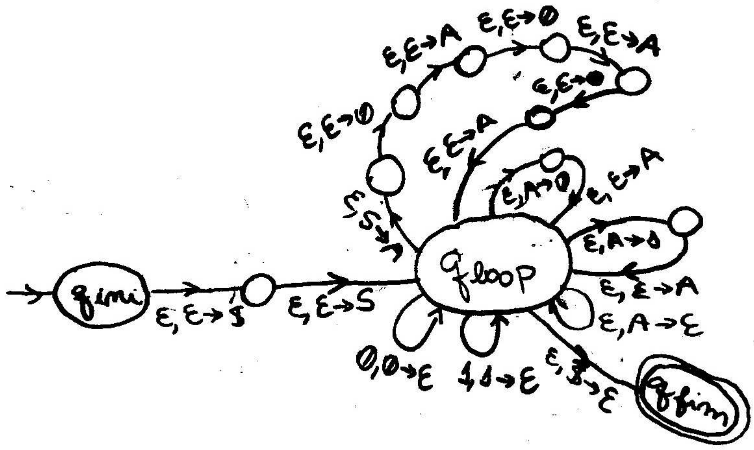
**Exercício 3** Represente autômatos com pilha para as linguagens do Exercício 2 através de diagramas de estados.

RESPOSTA =

0,2

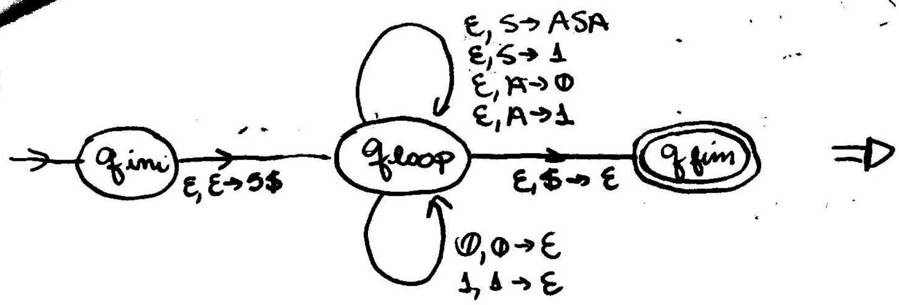
3 a)

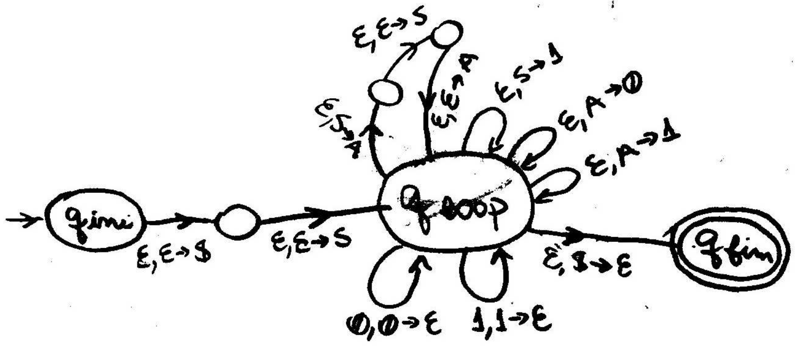

ε, S → AØAØAØA
ε, A → AØ
ε, A → A1
ε, A → ε
qini
ε, ε → S$
qloop
ε, $ → ε
qfim
Ø, Ø → ε
1, 1 → ε


ε,ε→A
ε,ε→Ø
ε,ε→A
ε,ε→Ø
ε,ε→Ø
ε,ε→A
ε,S→A
ε,A→Ø
ε,ε→A
ε,A→1
ε,ε→A
ε,A→ε
qini
ε,ε→$
ε,ε→S
qloop
Ø,Ø→ε
1,1→ε
ε,$→ε
qfim

0,2

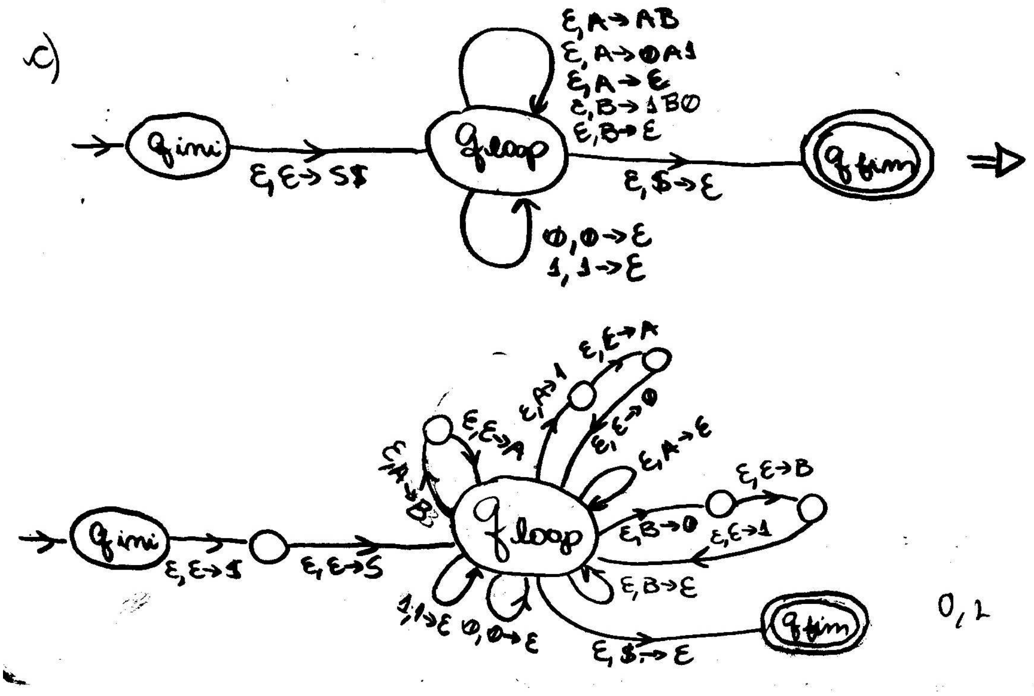
b)


ε, S → ASA
ε, S → 1
ε, A → Ø
ε, A → 1
qini
ε, ε → S$
qloop
ε, $ → ε
qfim
Ø, Ø → ε
1, 1 → ε
⇒


ε,ε→S
ε,S→A
ε,ε→A
ε,S→1
ε,A→Ø
ε,A→1
qini
ε,ε→$
ε,ε→S
qloop
ε,$→ε
qfim
Ø,Ø→ε
1,1→ε

0,2

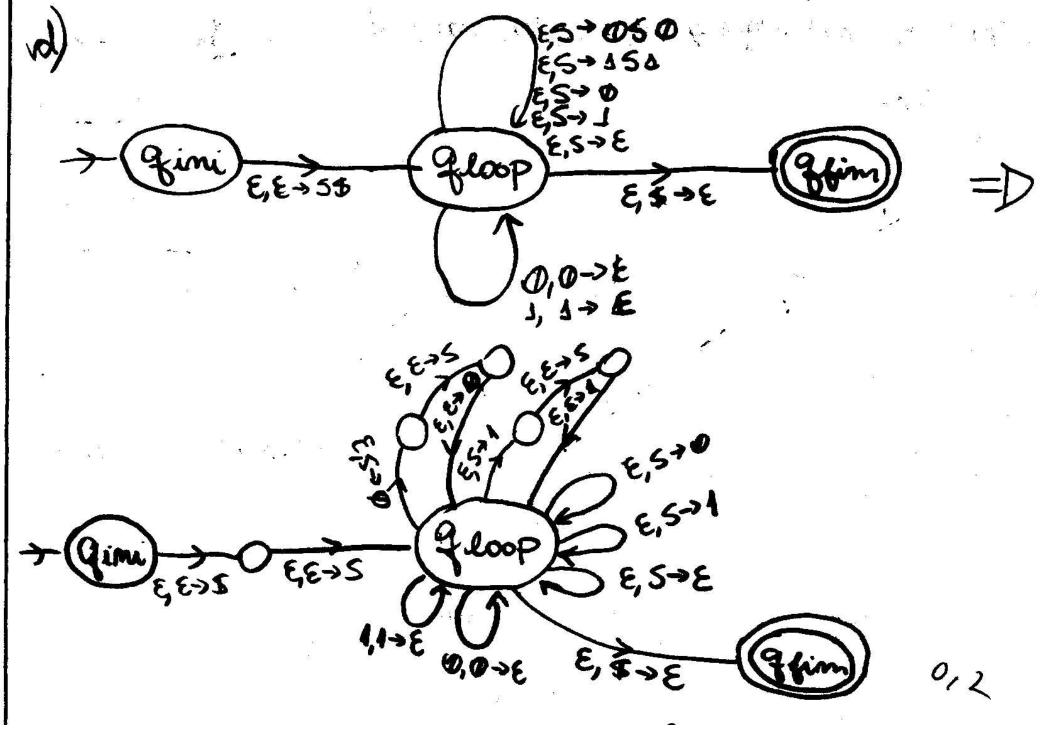
c)



qini
ε,ε→S$
qloop
ε,A→AB
ε,A→0A1
ε,A→ε
ε,B→1B0
ε,B→ε
ε,$→ε
qfim
0,0→ε
1,1→ε
qini
ε,ε→$
ε,ε→S
qloop
ε,ε→A
ε,A→1
ε,ε→A
ε,ε→0
ε,A→ε
ε,ε→B
ε,B→0
ε,ε→1
ε,B→ε
1,1→ε
0,0→ε
ε,$→ε
qfim

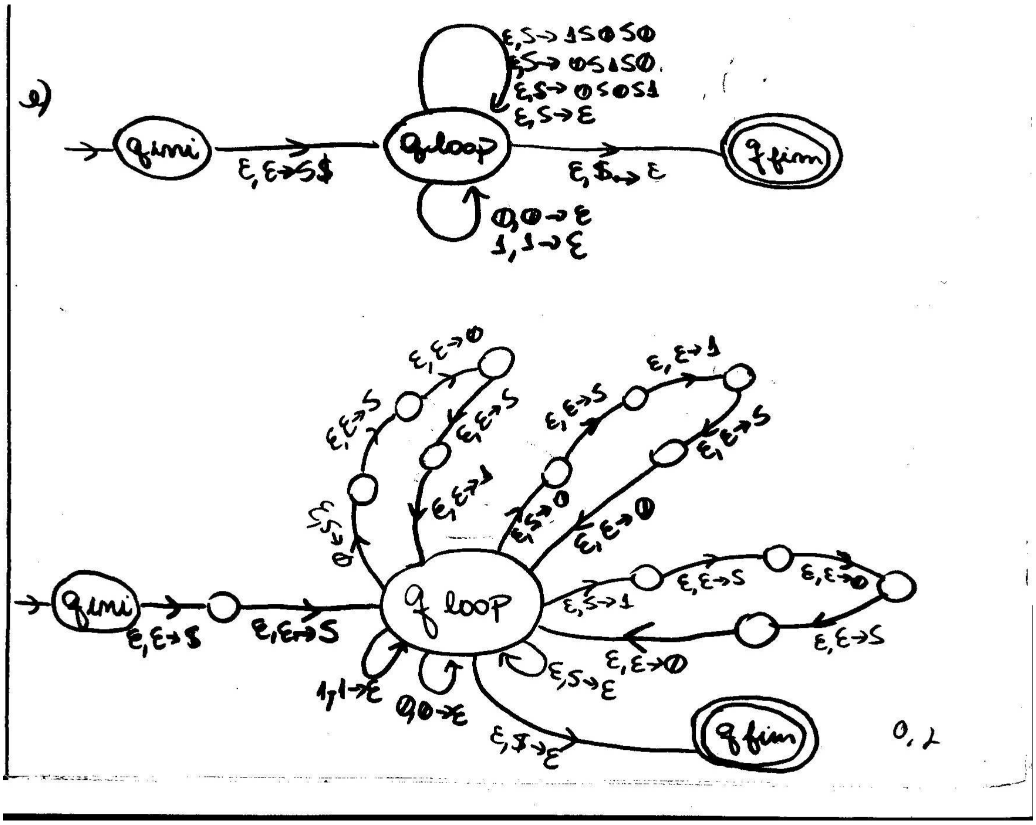

0,2

d)



qini
ε,ε→S$
qloop
ε,S→0S0
ε,S→1S1
ε,S→0
ε,S→1
ε,S→ε
ε,$→ε
qfim
0,0→ε
1,1→ε
qini
ε,ε→$
ε,ε→S
qloop
ε,ε→S
ε,ε→0
ε,S→1
ε,ε→S
ε,ε→1
ε,S→0
ε,S→1
ε,S→ε
1,1→ε
0,0→ε
ε,$→ε
qfim


0,2




.................................................................................................................................................

**Exercício 4** [2.2 Sipser]

**a.** Use as linguagens $A=\{a^m b^n c^n \mid m, n \geq 0\}$, $B=\{a^n b^m c^n \mid m, n \geq 0\}$ e $C=\{a^n b^n c^n \mid n \geq 0\}$ para mostrar que a classe de linguagens livre de contexto não é fechada sob intersecção. (Você pode usar sem prova que $C$ não é livre de contexto).

RESPOSTA = Dadas as linguagens A, B e C. Vamos mostrar que a classe de linguagens livre de contexto não é fechada sob interseção, usando o fato de que C não é livre de contexto.

Primeiramente, vamos mostrar que as linguagens A e B são livres de contexto.

1 – Queremos construir uma gramática que gera A:

$G_A = (\{S_A, S_1, S_2\}, \{a, b, c\}, R_A, S_A)$, onde:

- $R_A$: $S_A \to S_1S_2$
  $S_1 \to aS_1 \mid \varepsilon$
  $S_2 \to bS_2c \mid \varepsilon$

Como $G_A$ gera A e $G_A$ é livre de contexto, então A é livre de contexto.

2 – Queremos construir uma gramática que gera B:

$G_A = (\{S_B, B\}, \{a, b, c\}, R_B, S_B)$, onde:

- $R_B$: $S_B \to aS_Bc \mid B$
  $B \to Bb \mid \varepsilon$

Como $G_B$ gera B e $G_B$ é livre de contexto, então B é livre de contexto.

3 – Agora, vamos verificar $A \cap B$:

| | | | | |
|---|---|---|---|---|
| A | $\cap$ | B | = | C |
| $\{ a^m b^n c^n \mid m, n \geq 0\}$ | $\cap$ | $\{ a^n b^m c^n \mid m, n \geq 0\}$ | = | $\{ a^n b^n c^n \mid n \geq 0\}$ |
| nº de b’s = nº de c’s | $\cap$ | nº de a’s = nº de c’s | = | nº de a’s = nº de b’s = nº de c’s |

Como C não é livre de contexto, $A \cap B$ não é livre de contexto. Portanto, a classe de linguagens livre de contexto não é fechada sob interseção.

■

**b.** A Lei de Morgan diz que para quaisquer dois conjuntos vale que o complemento da união é a intersecção dos complementos. Use esse fato para mostrar que a classe das linguagens livres de contexto não é fechada sob complementação.

RESPOSTA = Tome linguagens livre de contexto A e B. Pela lei de Morgan, $\overline{A \cup B} = \overline{A} \cap \overline{B}$. Como será mostrado no exercícios 5 desta lista, a classe de linguagens livre de contexto é fechada sob união. Logo, $A \cup B$ é livre de contexto. No entanto, ao tomarmos o complemento dessa união, ó obtida uma interseção dos complementos de cada linguagem. Mas como foi mostrado no item a deste exercício, a classe de linguagens livre de contexto não é fechada sob interseção. Logo:

$\overline{A} \cap \overline{B}$ pode não ser livre de contexto.

■

.............................................................................................................................................

**Exercício 5** [2.15 Sipser] Mostre que a classe de linguagens livres de contexto é fechada sob as operações de união, concatenação e estrela.

RESPOSTA = Seja A uma linguagem livre de contexto. Portanto, existe uma gramática livre de contexto $G_A = (V_A, \Sigma, R_A, S_A)$ que gera A.

Seja B uma linguagem livre de contexto. Portanto, existe uma gramática livre de contexto $G_B = (V_B, \Sigma, R_B, S_B)$ que gera B.

Tome $V_A \cup V_B = \varnothing$.

Queremos mostrar que a classe de linguagens livre de contexto é fechada sob as operações de união, interseção e estrela. Para tal, vamos usar A e B:

1 – Seja uma linguagem C tal que $C = A \cup B$. Vamos construir uma gramática $G_C$ que gera C:

$G_C = (V_C, \Sigma, R_C, S_C)$, onde:

- $V_C = V_A \cup V_B \cup \{S_C\}$
- $R_C$: $R_A \cup R_B \cup \{S_C \text{->} S_A \mid S_B\}$
- $S_C \text{->} S_A \mid S_B$

Logo, como $C = A \cup B$, $G_C$ gera C e $G_C$ é livre de contexto, então $A \cup B$ é livre de contexto.

■

2 – Seja uma linguagem D tal que $D = A \bullet B$. Vamos construir uma gramática $G_D$ que gera D:

$G_D = (V_D, \Sigma, R_D, S_D)$, onde:

- $V_D = V_A \cup V_B \cup \{S_D\}$
- $R_D$: $R_A \cup R_B \cup \{S_D \text{->} S_A S_B\}$
- $S_D \text{->} S_A S_B$

Logo, como $D = A \bullet B$, $G_D$ gera D e $G_D$ é livre de contexto, então $A \bullet B$ é livre de contexto.

■

3 – Seja uma linguagem E tal que $E = A^*$. Vamos construir uma gramática $G_E$ que gera E:

$G_E = (V_E, \Sigma, R_E, S_E)$, onde:

- $V_E = V_A \cup \{S_E\}$
- $R_E$: $R_A \cup \{S_E \text{ -> } S_AS_E \mid \varepsilon\}$
- $S_E \text{ -> } S_AS_E \mid \varepsilon$

Logo, como E = A*, $G_E$ gera E e $G_E$ é livre de contexto, então A* é livre de contexto.

■

.....................................................................................................................................

**Exercício 6(!)** Mostre que se *G* é uma gramática livre de contexto na Forma Normal de Chomsky, então qualquer derivação de uma palavra $w \in L(G)$ de comprimento $n \geq 1$ utiliza exatamente $2n$-1 passos.

RESPOSTA = Seja G uma gramática livre de contexto na forma normal de Chomsky. Queremos provar que qualquer derivação de uma palavra w € L(G) de comprimento $n \geq 1$ utiliza exatamente 2n – 1 passos.

Como cada variável gera uma única derivação, então o número de derivações é o número de variáveis que surgem nas derivações.

Como G está na forma normal de Chomsky, as variáveis só podem surgir a partir de regras do tipo A -> BC, pois S -> ε e A -> a não geram variáveis.

Se cada variável gera duas, temos uma estrutura semelhante a uma árvore binária. As folhas dessa árvore binária de variáveis geram terminais pela regra A -> a. Logo, o número de folhas dessa árvore binária de variáveis é o número de terminais, que é o comprimento n de w.

Em analogia à teoria dos grafos, temos que, por teorema, o número de nós de uma árvore binária está relacionado com seu número de nós folhas por:

nº de nós = 2n – 1, onde n é o número de nós-folha

A estrutura de derivação A -> BC descreve uma árvore binária.

Como o número de passos de derivação é igual ao número de variáveis que surgem durante a derivação, isso é igual ao número de nós da árvore binária de variáveis.

Como o comprimento de w é igual ao número de literais da palavra, que são gerados pela regra A -> a, então esse número é igual ao número de nós-folhas da árvore binária de variáveis.

Assim:

n = nº de nós-folha da árvore binária de variáveis = |w|

Logo:

nº de passos de derivação = nº de nós da árvore binária de variáveis = 2n – 1

■

.....................................................................................................................................

**Exercício 7(!)** [5.1.8 Hopcroft] Considere a gramática livre de contexto *G* definida pelas produções: $S \rightarrow 0S1S \mid 1S0S \mid \varepsilon$. Prove que *G* gera apenas palavras com mesmo número de 0s e 1s e que todo palavra assim é gerado por *G*.

RESPOSTA = Seja G uma gramática livre de contexto definida pelas produções S -> 0S1S | 1S0S | ε.

Queremos provar que G gera apenas pelavras com mesmo número de 0's e 1's e que toda palavra assim é gerada por G. Para isso, vamos dividir a prova em DUAS partes (prova tipo *se e somente se...*):

1 – G gera apenas palavras com mesmo número de 0's e 1's.

Podemos provar este fato com o argumento de que a cada derivação de S em G, a ocorrência de um 0 implica na ocorrência de um 1 e voce-versa. Logo, toda palavra derivada de S tem o mesmo número de 0's e 1's.

2 – Toda palavra que contém o mesmo número de 0's e 1's é gerada por G.

Seja P a propriedade de uma palavra ter o mesmo número de 0's e 1's.

Tome $w = w_1w_2...w_n$. Sem perda de generalidade, vamos supor que w começa com 0. Então, deve existir uma posição i ocupada por um caracter em w tal que $w_i = 1$ e $w = 0x1y$, onde x e y têm a propriedade P.

CASO BASE: $n = 1 \Rightarrow w = \varepsilon$ (é gerada por G. Logo, *OK*)

HIPÓTESE DE INDUÇÃO: para todo w, $|w| < n$, G gera w

PASSO DA INDUÇÃO: Como $w = 0x1y$, x e y têm a propriedade P e $|x|, |y| < n$, pela hipótese de indução, G gera x e y.

Logo, x e y são derivdos de S e w pode ser derivada de 0S1S.

■

.................................................................................................................................................