# Ficha (7) de Exercícios sobre Propriedades das Linguagens Regulares

Resoluções/soluções para os exercícios selecionados: 1, 3, 4

**Lema da Bombagem para Linguagens Regulares:**
Seja L uma linguagem regular. Então existe uma constante n (dependente de L) tal que para todas as cadeias w em L com $|w| \geq n$ se pode partir w em 3 subcadeias w=xyz tais que:

- $y \neq \varepsilon$
- $|xy| \leq n$
- Para todo o $k \geq 0$, a cadeia $xy^kz$ também está em L.

---

1. Lema da Bombagem

a) Mostre que a linguagem sobre o alfabeto {0,1} $L=\{0^n1^{2n} \mid n\geq1\}$ não é uma linguagem regular, usando o lema da bombagem.

Vamos assumir que L é uma linguagem regular. Se é uma linguagem regular tem de satisfazer o lema da bombagem para linguagens regulares.

Escolhemos $w=0^n1^{2n}$, cujo comprimento, $|w|=3n$, é $\geq n$

w=xyz

Como $|xy| \leq n$ e $|y| \geq 1$, então y só pode ter 0's

e o comprimento de y: $1 \leq |y| \leq n$

Dadas estas restrições, as decomposições, xyz, possíveis de concretizar podem ser representadas por:

$0^j0^t0^{n-t-j}1^{2n}$, com t (|y|), $1 \leq t \leq n$, e $0 \leq j \leq n-1$

Como para todo o $k \geq 0$, a cadeia $xy^kz$ também está em L, se fizermos k=0 temos:

$0^j0^{n-t-j}1^{2n}$, ou seja 2n 1's e n-t (j+n-t-j = n-t) 0's, e como $1 \leq t \leq n$, o número de 1's será sempre maior do que o dobro dos 0's e por isso essas cadeias (dadas por $0^j0^{n-t-j}1^{2n}$) não pertencem a L.

Uma outra opção para a caixa de texto:

e o comprimento de y: $1 \leq |y| \leq n$

Dadas estas restrições, as decomposições, xyz, possíveis de concretizar podem ser representadas por:

$0^i0^t0^j1^{2n}$, com $1 \leq t \leq n$, $j,i \geq 0$, e $i+j+t = n$

Como para todo o $k \geq 0$, a cadeia $xy^kz$ também está em L, se fizermos k=0 temos:

$0^i0^j1^{2n}$, ou seja 2n 1's e n-t 0's, e como $1 \leq t \leq n$, o número de 1's será sempre maior do que o dobro dos 0's e por isso essas cadeias (dadas por $0^i0^j1^{2n}$) não pertencem a L.

Em vez do que está na caixa de texto, seria aceitável como prova apenas dizer:

xz (k=0), que deveria estar em L, tem menos de n 0's seguidos de 2n 1's e por isso não está em L.

A linguagem não pode por isso ser uma linguagem regular. *qed*

b) Mostre, usando o lema da bombagem, que a linguagem das cadeias da forma $v1^m$, em que v é uma cadeia arbitrária de 0 e 1 de comprimento m, não é regular.

$L = (0+1)^m1^m$

Pode usar-se $w=0^n1^n$ e reduzir a prova à prova que $0^n1^n$ não é uma linguagem regular.

c) Se tentarmos provar que a linguagem formada pela expressão regular (00+11)* não obedece ao lema da bombagem para linguagens regulares não conseguimos.

Por exemplo, se escolhermos $w=(00+11)^n$ e $x=\varepsilon$, $y=(00+11)$, e $z = (00+11)^{n-1}$ repeitamos todas as condições do lema, incluindo $\forall k \geq 0, xy^kz \in L$

e por isso não conseguimos provar que é uma linguagem não regular.

d) Se tentarmos provar que a linguagem formada pela expressão regular 01*0*1 não obedece ao lema da bombagem para linguagens regulares não conseguimos.

Por exemplo, se escolhermos $w=01^n0^m1$ e $x=0$, $y=1$, e $z = 1^{n-1}0^m1$ repeitamos todas as condições do lema, incluindo $\forall k \geq 0, xy^kz \in L$

e por isso não conseguimos provar que é uma linguagem não regular.

e) Idem para a linguagem $\{0^n \mid n$ é um quadrado perfeito$\}$.

Vamos assumir que L é uma linguagem regular. Se é uma linguagem regular tem de satisfazer o lema da bombagem para linguagens regulares.

Escolhemos $w=0^{n^2}$

$w=xyz$, e $|w| = |xyz| = n^2$ (logo $|w| \geq n$)

Como $|xy| \leq n$, y contém entre 1 a n 0's ($1 \leq |y| \leq n$)

Para k=2 temos:

$n^2 + 1 \leq |xyyz| \leq n^2 + n$

O quadrado perfeito a seguir a $n^2$ é $(n+1)^2 = n^2 + 2n + 1 > n^2 + n$

Logo, |xyyz| não pode ser um quadrado perfeito! Mas xyyz deveria fazer parte da linguagem L e não faz!

L não é uma linguagem regular. *qed*

---

3. As propriedades de fecho também podem ser usadas para mostrar que certas linguagens não são regulares. Partindo do conhecimento de que $L_{0^n1^n} = \{0^n1^n \mid n \geq 0\}$ não é regular, mostre que $\{0^i1^j \mid i \neq j\}$ não é regular, usando operações que preservam a regularidade.

Seja $L_1 = \{0^i1^j \mid i \neq j\}$

Podemos obter $L_{0^n1^n}$ usando as seguintes operações fechadas sobre as ling. regulares:

$L_{0^n1^n} = \overline{L_1} \cap L(0^*1^*)$

Como $L(0^*1^*)$ é uma linguagem regular,

Se $L_1$ fosse regular então o complemento de $L_1$ também seria uma ling. regular, e como $L(0^*1^*)$ é uma linguagem regular, a intersecção das duas linguagens ( $\overline{L_1} \cap L(0^*1^*)$) também originaria uma ling. regular.

Como sabemos que $L_{0^n1^n}$ não é uma linguagem regular então $L_1$ não pode ser uma linguagem regular.

---

4. Suponha uma linguagem (regular) L definida sobre um alfabeto $\Sigma$.

a) Apresente um algoritmo para determinar se a linguagem é infinita. Sugestão: use o lema da bombagem e o comprimento das cadeias.

Testar todas as strings de comprimento entre n e 2n-1 (n é o número de estados do DFA). Se alguma string pertencer a L então L é uma linguagem infinita.

Se não existir nenhuma string de comprimento entre n e 2n-1 aceite também não há strings aceites de tamanho $\geq$ 2n e portanto a linguagem L é finita (todas as strings aceites são de tamnho < n).

[Nota: uma outra opção seria identificar ciclos no grafo que representa o DFA, mas teria de se ter cuidado de forma a que os ciclos identificados fossem ciclos que contribuissem para a aceitação de strings, ou seja que estivessem em caminhos do estado inicial para um dos estados de aceitação. Após a minimização de estados a identificação de ciclos é mais fácil?]

b) Apresente um algoritmo para determinar se $L=\Sigma^*$, isto é, todas as cadeias do alfabeto.

Pode-se fazer o complemento do DFA da linguagem e verificar que não aceita nenhuma string.

Testar o DFA complemento com todas as strings de tamanho entre n e 2n-1 (n é um número de estados do DFA). Se nenhuma for aceite então o complemento de L é uma linguagem vazia a L é a linguagem que aceita todas as strings no alfabeto.

[Verificar se o complemento de um DFA não aceita strings pode ser feito analisando se existem caminhos no DFA do estado de entrada para estados de aceitação.]

---