EXEMPLO 1.77

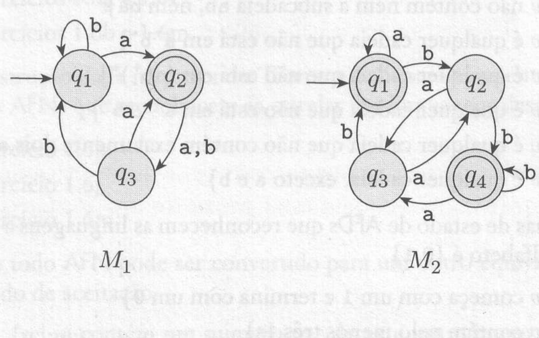
Algumas vezes, "bombear para baixo" é útil quando aplicamos o lema do bombeamento. Usamos o lema do bombeamento para mostrar que $E = \{0^i 1^j | i > j\}$ não é regular. A prova é por contradição.

Suponha que E seja regular. Seja p o comprimento de bombeamento para E dado pelo lema do bombeamento. Seja $s = 0^{p+1}1^p$. Então s pode ser dividida em xyz, satisfazendo as condições do lema do bombeamento. Pela condição 3, y contém somente 0s. Vamos examinar a cadeia $xyyz$ para ver se ela pode estar em E. Adicionar uma cópia extra de y aumenta o número de 0s. Mas, como E contém todas as cadeias em 0^*1^* que têm mais 0s que 1s, aumentando-se o número de 0s obtém-se ainda uma cadeia em E. Não ocorre contradição. Precisamos tentar alguma outra coisa.

O lema do bombeamento afirma que $xy^iz \in E$ mesmo quando $i = 0$; assim, vamos considerar a cadeia $xy^0z = xz$. Removendo-se a cadeia y, o número de 0s em s diminui. Lembre-se que s tem apenas um 0 a mais que 1s. Portanto, xz não pode ter mais 0s que 1s e, logo, não pode ser um membro de E. Dessa forma, obtemos uma contradição. ■

EXERCÍCIOS

R1.1 A seguir estão os diagramas de estado de dois AFDs, M_1 e M_2. Responda às seguintes questões sobre cada uma dessas máquinas.



a. Qual é o estado inicial?

b. Qual é o conjunto de estados de aceitação?

c. Por qual seqüência de estados a máquina passa para a entrada aabb?

d. A máquina aceita a cadeia aabb?

e. A máquina aceita a cadeia ε?

1.4 f, g 1.5 c, g

R1.2 Dê a descrição formal das máquinas M_1 e M_2 desenhadas no Exercício 1.1.

1.3 A descrição formal de um AFD M é $(\{q_1, q_2, q_3, q_4, q_5\}, \{\text{u}, \text{d}\}, \delta, q_3, \{q_3\})$, onde δ é dada pela tabela a seguir. Dê o diagrama de estados dessa máquina.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

1.4 Cada uma das linguagens a seguir é a interseção de duas linguagens mais simples. Em cada caso, construa AFDs para as linguagens mais simples, e depois combine-os usando a construção discutida na nota de rodapé 3 (página 47) para obter o diagrama de estados de um AFD para a linguagem dada. Em todos as casos, $\Sigma = \{\text{a}, \text{b}\}$.

- **a.** $\{w |$ w tem pelo menos três as e pelo menos dois bs$\}$
- R**b.** $\{w |$ w tem exatamente dois as e pelo menos dois bs$\}$
- **c.** $\{w |$ w tem um número par de as e um ou dois bs$\}$
- R**d.** $\{w |$ w tem um número par de as e cada a é seguido por pelo menos um b$\}$
- **e.** $\{w |$ w tem um número par de as e um ou dois bs$\}$
- **f.** $\{w |$ w tem um número ímpar de as e termina com um b$\}$
- **g.** $\{w |$ w tem comprimento par e um número ímpar de as$\}$

1.5 Cada uma das linguagens a seguir é o complemento de uma linguagem mais simples. Em cada caso, construa um AFD para a linguagem mais simples, e use-o para obter o diagrama de estados de um AFD para a linguagem dada. Em todos os casos $\Sigma = \{\text{a}, \text{b}\}$.

- R**a.** $\{w |$ w não contém a subcadeia ab$\}$
- R**b.** $\{w |$ w não contém a subcadeia baba$\}$
- **c.** $\{w |$ w não contém nem a subcadeia ab, nem ba$\}$
- **d.** $\{w |$ w é qualquer cadeia que não está em $\text{a}^*\text{b}^*\}$
- **e.** $\{w |$ w é qualquer cadeia que não está em $(\text{ab}^+)^*\}$
- **f.** $\{w |$ w é qualquer cadeia que não está em $\text{a}^* \cup \text{b}^*\}$
- **g.** $\{w |$ w é qualquer cadeia que não contém exatamente dois as$\}$
- **h.** $\{w |$ w é qualquer cadeia, exceto a e b$\}$

1.6 Dê diagramas de estado de AFDs que reconhecem as linguagens a seguir. Em todos os casos o alfabeto é $\{0,1\}$

- **a.** $\{w |$ w começa com um 1 e termina com um 0$\}$
- **b.** $\{w |$ w contém pelo menos três 1s$\}$
- **c.** $\{w |$ w contém a subcadeia 0101, isto é, $w = x0101y$ para algum x e algum $y\}$
- **d.** $\{w |$ w tem comprimento pelo menos 3 e seu terceiro símbolo é um 0$\}$
- **e.** $\{w |$ w começa com 0 e tem comprimento ímpar, ou começa com 1 e tem comprimento par$\}$

f. $\{w \mid w$ não contém a subcadeia $110\}$
g. $\{w \mid$ o comprimento de w é no máximo $5\}$
h. $\{w \mid w$ é qualquer subcadeia exceto 11 e $111\}$
i. $\{w \mid$ toda posição ímpar de w é um $1\}$
j. $\{w \mid w$ contém pelo menos dois 0s e no máximo um $1\}$
k. $\{\varepsilon, 0\}$
l. $\{w \mid w$ contém um número par de 0s, ou contém exatamente dois 1s$\}$
m. O conjunto vazio
n. Todas as cadeias exceto a cadeia vazia

1.7 Dê diagramas de estado de AFNs com o número especificado de estados reconhecendo cada uma das linguagens a seguir. Em todos os casos o alfabeto é $\{0,1\}$.

[R]**a.** A linguagem $\{w \mid w$ termina com $00\}$ com três estados
b. A linguagem do Exercício 1.6c com cinco estados
c. A linguagem do Exercício 1.6l com seis estados
d. A linguagem $\{0\}$ com dois estados
e. A linguagem $0^*1^*0^+$ com três estados
[R]**f.** A linguagem $1^*(001^+)^*$ com três estados
g. A linguagem $\{\varepsilon\}$ com um estado
h. A linguagem 0^* com um estado

1.8 Use a construção dada na prova do Teorema 1.45 para apresentar os diagramas de estados dos AFNs que reconhecem as uniões das linguagens descritas em

a. Exercícios 1.6a e 1.6b.
b. Exercícios 1.6c e 1.6f.

1.9 Use a construção fornecida na prova do Teorema 1.47 para dar os diagramas de estados dos AFNs que reconhecem as concatenações das linguagens descritas em

a. Exercícios 1.6g e 1.6i.
b. Exercícios 1.6b e 1.6m.

1.10 Use a construção dada na prova do Teorema 1.49 para apresentar os diagramas de estado dos AFNs que reconhecem as estrelas das linguagens descritas em

a. Exercício 1.6b.
b. Exercício 1.6j.
c. Exercício 1.6m.

[R]**1.11** Prove que todo AFN pode ser convertido para um outro equivalente que tem um único estado de aceitação.

1.12 Seja $D = \{w \mid w$ contém um número par de as e um número ímpar de bs e não contém a subcadeia ab$\}$. Apresente um AFD com cinco estados que reconheça D e uma expressão regular que gere D. (Sugestão: descreva D com mais simplicidade.)

1.13 Seja F a linguagem de todas as cadeias sobre $\{0,1\}$ que não contenham um par de 1s que estejam separados por um número ímpar de símbolos. Apresente o diagrama de estados de um AFD com cinco estados que reconheça F. (Você pode achar útil encontrar primeiro um AFN de quatro estados para o complemento de F.)

1.14 **a.** Mostre que, se M é um AFD que reconhece a linguagem B, tornando-se estados de aceitação os estados de M que não são de aceitação, e vice-versa, obtém-se um novo AFD que reconhece o complemento de B. Conclua que a classe das linguagens regulares é fechada sob complemento.

b. Mostre por meio de um exemplo que, se M é um AFN que reconhece a linguagem C, tornando-se estados de aceitação os estados de M que não são de aceitação, e vice-versa, não necessariamente se obtém um novo AFN que reconhece o complemento de C. A classe das linguagens reconhecidas por AFNs é fechada sob complemento? Explique sua resposta.

1.15 Dê um contra-exemplo para mostrar que a seguinte construção falha em provar o Teorema 1.49, o fecho da classe de linguagens regulares sob a operação estrela.[8] Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1. Construa $N = (Q_1, \Sigma, \delta, q_1, F)$ como segue. N supostamente reconhece A_1^*.

a. Os estados de N são os estados de N_1.

b. O estado inicial de N é o mesmo que o estado incial de N_1.

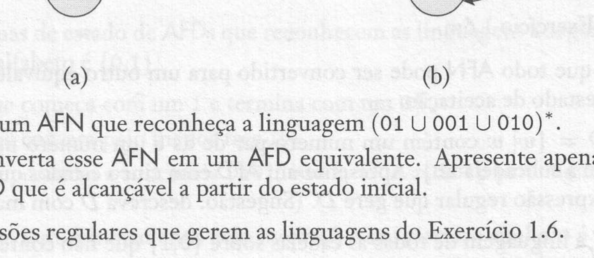
c. $F = \{q_1\} \cup F_1$.
Os estados de aceitação F são os estados de aceitação antigos mais seu estado inicial.

d. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \notin F_1 \text{ ou } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ e } a = \varepsilon. \end{cases}$$

(Sugestão: Mostre essa construção graficamente, como na Figura 1.50.)

1.16 Use a construção dada no Teorema 1.39 para converter os dois autômatos finitos não-determinísticos apresentados a seguir em autômatos finitos determinísticos equivalentes.



(a) (b)

1.17 **a.** Dê um AFN que reconheça a linguagem $(01 \cup 001 \cup 010)^*$.

b. Converta esse AFN em um AFD equivalente. Apresente apenas a parte do AFD que é alcançável a partir do estado inicial.

1.18 Dê expressões regulares que gerem as linguagens do Exercício 1.6.

---

[8] Em outras palavras, você deve apresentar um autômato finito, N_1, para o qual o autômato construído, N, não reconhece a estrela da linguagem de N_1.

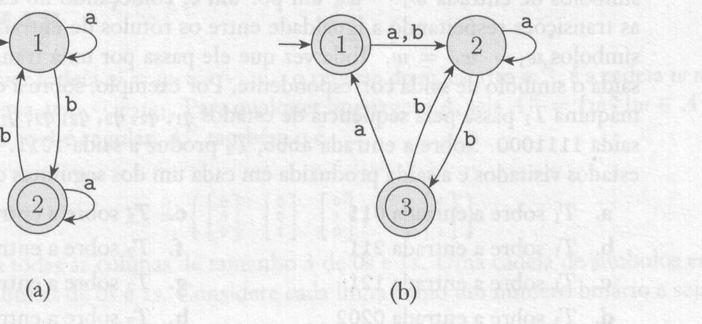
1.19 Use o procedimento descrito no Lema 1.55 para converter as seguintes expressões regulares em autômatos finitos não-determinísticos.

a. $(0 \cup 1)^*000(0 \cup 1)^*$
b. $(((00)^*(11)) \cup 01)^*$
c. $\emptyset^*$

1.20 Para cada uma das linguagens a seguir, apresente duas cadeias que sejam membros e duas que *não* sejam membros — um total de quatro cadeias para cada caso. Presuponha o alfabeto $\Sigma = \{a,b\}$ em todos os casos.

a. a^*b^*
b. $a(ba)^*b$
c. $a^* \cup b^*$
d. $(aaa)^*$
e. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$
f. $aba \cup bab$
g. $(\varepsilon \cup a)b$
h. $(a \cup ba \cup bb)\Sigma^*$

1.21 Use o procedimento descrito no Lema 1.60 para converter os seguintes autômatos finitos em expressões regulares.



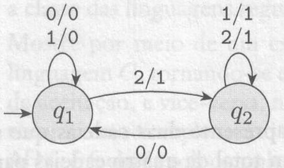
(a) (b)

1.22 Em certas linguagens de programação, comentários ocorrem entre delimitadores como /# e #/. Seja C a linguagem de todas as cadeias de comentários delimitados. Um membro de C deve começar com /# e terminar com #/, mas não pode ter qualquer outra ocorrência de #/. Por simplicidade, vamos dizer que os comentários propriamente ditos são escritos com apenas os símbolos a e b; assim, o alfabeto de C é $\Sigma = \{a, b, /, \#\}$.

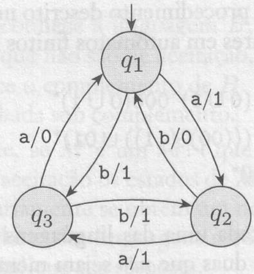
a. Dê um AFD que reconheça C.
b. Dê uma expressão regular que gere C.

R**1.23** Seja B qualquer linguagem sobre o alfabeto Σ. Prove que $B = B^+$ sse $BB \subseteq B$.

1.24 Um ***transdutor de estado finito*** (TEF) é um tipo de autômato finito determinístico cuja saída é uma cadeia, e não simplesmente *aceite* ou *rejeite*. Os diagramas de estado a seguir são dos transdutores de estado finito T_1 e T_2.



T_1



T_2

Cada transição de um TEF é rotulada com dois símbolos, um designando o símbolo de entrada para aquela transição, e outro designando o símbolo de saída. Os dois símbolos são escritos com uma barra, /, separando-os. Em T_1, a transição de q_1 para q_2 tem símbolos de entrada 2 e símbolo de saída 1. Algumas transições podem ter múltiplos pares de entrada–saída, como a transição em T_1 de q_1 para si próprio. Quando um TEF computa sobre uma cadeia de entrada w, ele toma os símbolos de entrada $w_1 \cdots w_n$ um por um e, começando no estado inicial, segue as transições respeitando a igualdade entre os rótulos de entrada e a seqüência de símbolos $w_1 \cdots w_n = w$. Toda vez que ele passa por uma transição, ele dá como saída o símbolo de saída correspondente. Por exemplo, sobre a entrada 2212011, a máquina T_1 passa pela seqüência de estados $q_1, q_2, q_2, q_2, q_2, q_1, q_1, q_1$ e produz a saída 1111000. Sobre a entrada abbb, T_2 produz a saída 1011. Dê a seqüência de estados visitados e a saída produzida em cada um dos seguintes casos.

- **a.** T_1 sobre a entrada 011
- **b.** T_1 sobre a entrada 211
- **c.** T_1 sobre a entrada 121
- **d.** T_1 sobre a entrada 0202
- **e.** T_2 sobre a entrada b
- **f.** T_2 sobre a entrada bbab
- **g.** T_2 sobre a entrada bbbbbb
- **h.** T_2 sobre a entrada ε

1.25 Leia a definição informal do transdutor de estado finito dada no Exercício 1.24. Dê uma definição formal desse modelo, seguindo o padrão da Definição 1.5 (página 36). Suponha que um TEF tem um alfabeto de entrada Σ e um alfabeto de saída Γ mas não um conjunto de estados de aceitação. Inclua uma definição formal da computação de um TEF. (Dica: Um TEF é uma 5-upla. Sua função de transição é da forma $\delta\colon Q \times \Sigma \longrightarrow Q \times \Gamma$.)

1.26 Usando a solução que você forneceu para o Exercício 1.25, dê uma descrição formal das máquinas T_1 e T_2 apresentadas no Exercício 1.24.

1.27 Leia a definição informal do transdutor de estado finito dada no Exercício 1.24. Dê o diagrama de estados de um TEF com o seguinte comportamento. Seus alfabetos de entrada e de saída são $\{0,1\}$. Sua cadeia de saída é idêntica à de entrada nas posições pares, mas invertida nas posições ímpares. Por exemplo, sobre a entrada 0000111 ele deve emitir a saída 1010010.

1.28 Converta as seguintes expressões regulares em AFNs usando o procedimento dado no Teorema 1.54. Em todos os casos $\Sigma = \{a, b\}$.

- **a.** $\mathtt{a(abb)^* \cup b}$

b. $a^+ \cup (ab)^+$
c. $(a \cup b^+)a^+b^+$

1.29 Use o lema do bombeamento para mostrar que as linguagens a seguir não são regulares.

[R]**a.** $A_1 = \{0^n 1^n 2^n \mid n \geq 0\}$
b. $A_2 = \{www \mid w \in \{a, b\}^*\}$
[R]**c.** $A_3 = \{a^{2^n} \mid n \geq 0\}$ (Aqui, a^{2^n} significa uma cadeia de 2^n as.)

1.30 Descreva o erro na seguinte "prova" de que 0^*1^* não é uma linguagem regular. (Deve existir um erro, pois 0^*1^* *é* regular.) A prova é por contradição. Suponha que 0^*1^* seja regular. Seja p o comprimento de bombeamento para 0^*1^* dado pelo lema do bombeamento. Escolha s como a cadeia 0^p1^p. Você sabe que s é um membro de 0^*1^*, mas o Exemplo 1.73 mostra que s não pode ser bombeada. Assim, você tem uma contradição. Portanto, 0^*1^* não é regular.

PROBLEMAS

1.31 Para qualquer cadeia $w = w_1w_2 \cdots w_n$, o ***reverso*** de w, escrito $w^{\mathcal{R}}$, é a cadeia w na ordem reversa, $w_n \cdots w_2w_1$. Para qualquer linguagem A, seja $A^{\mathcal{R}} = \{w^{\mathcal{R}} \mid w \in A\}$. Mostre que se A é regular, $A^{\mathcal{R}}$ também o é.

1.32 Seja

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

Σ_3 contém todas as colunas de tamanho 3 de 0s e 1s. Uma cadeia de símbolos em Σ_3 dá três linhas de 0s e 1s. Considere cada linha como um número binário e seja

$$B = \{w \in \Sigma_3^* \mid \text{a última linha de } w \text{ é a soma das duas primeiras}\}.$$

Por exemplo,

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in B, \quad \text{mas} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \notin B.$$

Mostre que B é regular. (Dica: trabalhar com $B^{\mathcal{R}}$ é mais fácil. Você pode supor o resultado afirmado no Problema 1.31.)

1.33 Seja

$$\Sigma_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

Aqui, Σ_2 possui todas as colunas de 0s e 1s de altura dois. Uma cadeia de símbolos em Σ_2 dá duas linhas de 0s e 1s. Considere cada linha como um número binário, e seja

$$C = \{w \in \Sigma_2^* \mid \text{a última linha de } w \text{ é três vezes a primeira}\}.$$

Por exemplo, $\begin{bmatrix} 0 \\ 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in C$, mas $\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \notin C$. Mostre que C é regular. (Você pode supor o resultado afirmado no Problema 1.31.)

1.34 Seja Σ_2 o mesmo que no Problema 1.33. Considere cada linha como um número binário, e seja

$$D = \{w \in \Sigma_2^* | \text{ o número na primeira linha de } w \text{ é maior que o número na última linha}\}.$$

Por exemplo, $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \in D$, mas $\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} \notin D$. Mostre que D é regular.

1.35 Seja Σ_2 o mesmo que no Problema 1.33. Considere a primeira e a última linhas como cadeias de 0s e 1s, e seja

$$E = \{w \in \Sigma_2^* | \text{ a última linha de } w \text{ é o reverso da primeira linha de } w\}.$$

Mostre que E não é regular.

1.36 Seja $B_n = \{\mathtt{a}^k | \ k \text{ é um múltiplo de } n\}$. Mostre que para cada $n \geq 1$, a linguagem B_n é regular.

1.37 Seja $C_n = \{x | \ x \text{ é um número binário múltiplo de } n\}$. Mostre que para cada $n \geq 1$, a linguagem C_n é regular.

1.38 Um AFN-***de-todos-os-caminhos*** M é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ que aceita $x \in \Sigma^*$ se *todo* estado em que M pode estar após ler a entrada x é um estado de F. Note que, em contraste, um AFN ordinário aceita uma cadeia se *algum* estado entre os alcançáveis por M é um estado de aceitação. Prove que os AFNs-de-todos-os-caminhos reconhecem a classe das linguagens regulares.

1.39 A construção no Teorema 1.54 mostra que todo AFNG é equivalente a um AFNG com apenas dois estados. Podemos mostrar que um fenômeno oposto ocorre para AFDs. Prove que para todo $k > 1$ existe uma linguagem $A_k \subseteq \{0,1\}^*$ que é reconhecida por um AFD com k estados, mas não por um com apenas $k-1$ estados.

1.40 Digamos que uma cadeia x é um ***prefixo*** de uma cadeia y se existe uma cadeia z tal que $xz = y$, e que x é um ***prefixo próprio*** de y se, adicionalmente, $x \neq y$. Em cada um dos itens a seguir, definimos uma operação sobre uma linguagem A. Mostre que a classe das linguagens regulares é fechada sob aquela operação.

^{R}**a.** NÃOPREFIXO$(A) = \{w \in A |$ nenhum prefixo próprio de w é um membro de $A\}$.

b. NÃOESTENDE$(A) = \{w \in A | w$ não é prefixo próprio de nenhuma cadeia em $A\}$.

1.41 Para as linguagens A e B, seja o ***embaralhamento perfeito*** de A e B a linguagem

$$\{w | \ w = a_1 b_1 \cdots a_k b_k, \text{ onde } a_1 \cdots a_k \in A \text{ e } b_1 \cdots b_k \in B, \text{ cada } a_i, b_i \in \Sigma\}.$$

Mostre que a classe das linguagens regulares é fechada sob embaralhamento perfeito.

1.42 Para as linguagens A e B, seja o ***embaralhamento*** de A e B a linguagem

$$\{w | \ w = a_1 b_1 \cdots a_k b_k, \text{ onde } a_1 \cdots a_k \in A \text{ e } b_1 \cdots b_k \in B, \text{ cada } a_i, b_i \in \Sigma^*\}.$$

Mostre que a classe das linguagens regulares é fechada sob embaralhamento.

1.43 Seja A uma linguagem qualquer. Defina $REM(A)$ como a linguagem contendo todas as cadeias que podem ser obtidas pela remoção de um símbolo de uma cadeia em A. Assim, $REM(A) = \{xz \mid xyz \in A \text{ onde } x, z \in \Sigma^*, y \in \Sigma\}$. Mostre que a classe das linguagens regulares é fechada sob a operação REM. Apresente ambas, uma prova por meio de figuras e uma prova mais formal por construção, como no Teorema 1.47.

R**1.44** Sejam B e C linguagens sobre $\Sigma = \{0, 1\}$. Defina

$$B \overset{1}{\leftarrow} C = \{w \in B \mid \text{para algum } y \in C, \text{ as cadeias } w \text{ e } y \text{ contêm número igual de 1s}\}.$$

Mostre que a classe das linguagens regulares é fechada sob a operação $\overset{1}{\leftarrow}$.

$\star$**1.45** Seja $A/B = \{w \mid wx \in A \text{ para algum } x \in B\}$. Mostre que se A é regular e B é qualquer linguagem, então A/B é regular.

1.46 Prove que as seguintes linguagens não são regulares. Você pode usar o lema do bombeamento e o fechamento da classe das linguagens regulares sob união, interseção e complemento.

a. $\{0^n 1^m 0^n \mid m, n \geq 0\}$

R**b.** $\{0^m 1^n \mid m \neq n\}$

c. $\{w \mid w \in \{0,1\}^* \text{ não é um palíndromo}\}$[9]

d. $\{wtw \mid w, t \in \{0,1\}^+\}$

1.47 Sejam $\Sigma = \{1, \#\}$ e

$$Y = \{w \mid w = x_1\#x_2\#\cdots\#x_k \text{ para } k \geq 0, \text{ cada } x_i \in 1^*, \text{ e } x_i \neq x_j \text{ para } i \neq j\}.$$

Prove que Y não é regular.

1.48 Sejam $\Sigma = \{0,1\}$ e

$$D = \{w \mid w \text{ contém um número igual de ocorrências das subcadeias 01 e 10}\}.$$

Logo, $101 \in D$ porque 101 contém um único 01 e um único 10, mas $1010 \notin D$ porque 1010 contém dois 10s e um 01. Mostre que D é uma linguagem regular.

1.49 **a.** Seja $B = \{1^k y \mid y \in \{0,1\}^* \text{ e } y \text{ contém pelo menos } k \text{ 1s, para } k \geq 1\}$. Mostre que B é uma linguagem regular.

b. Seja $C = \{1^k y \mid y \in \{0,1\}^* \text{ e } y \text{ contém no máximo } k \text{ 1s, para } k \geq 1\}$. Mostre que C não é uma linguagem regular.

R**1.50** Leia a definição informal de transdutor de estado finito dada no Exercício 1.24. Prove que nenhum TEF pode dar como saída $w^{\mathcal{R}}$, para toda entrada w, se os alfabetos de entrada e de saída são $\{0,1\}$.

1.51 Sejam x e y cadeias e seja L uma linguagem qualquer. Dizemos que x e y são ***distingüíveis por L*** se existe alguma cadeia z tal que exatamente uma das cadeias, xz ou yz, é um membro de L; caso contrário, para toda cadeia z, temos $xz \in L$ sempre que $yz \in L$ e dizemos que x e y são ***indistingüíveis por L***. Se x e y são indistingüíveis por L escrevemos $x \equiv_L y$. Mostre que $\equiv_L$ é uma relação de equivalência.

---

[9] Um ***palíndromo*** é uma cadeia que tem a mesma leitura da frente para trás e de trás para frente.

R⋆**1.52** **Teorema de Myhill–Nerode.** Olhe para o Problema 1.51. Seja L uma linguagem e suponha que X seja um conjunto de cadeias. Digamos que X é ***distingüível duas-a-duas por L*** se cada duas cadeias distintas em X são distingüíveis por L. Defina o ***índice de L*** como sendo o número máximo de elementos em qualquer conjunto que é distingüível duas-a-duas por L. O índice de L pode ser finito ou infinito.

a. Mostre que, se L é reconhecida por um AFD com k estados, L tem índice no máximo k.

b. Mostre que, se o índice de L é um número finito k, ela é reconhecida por um AFD com k estados.

c. Conclua que L é regular sse ela tem um índice finito. Além disso, seu índice é o tamanho do menor AFD que a reconhece.

1.53 Seja $\Sigma = \{0, 1, +, =\}$ e

$$SOMA = \{x{=}y{+}z \mid x, y, z \text{ são inteiros binários, e } x \text{ é a soma de } y \text{ e } z\}.$$

Mostre que $SOMA$ não é regular.

1.54 Considere a linguagem $F = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0 \text{ e se } i = 1 \text{ então } j = k\}$.

a. Mostre que F não é regular.

b. Mostre que F funciona como uma linguagem regular no lema do bombeamento. Em outras palavras, dê um comprimento de bombeamento p e demonstre que F satisfaz as três condições do lema do bombeamento para esse valor de p.

c. Explique porque os itens (a) e (b) não contradizem o lema do bombeamento.

1.55 O lema do bombeamento diz que toda linguagem regular tem um comprimento de bombeamento p, tal que toda cadeia na linguagem pode ser bombeada se ela tiver comprimento p ou mais. Se p é um comprimento de bombeamento para a linguagem A, o mesmo acontece com qualquer comprimento $p' \geq p$. O ***comprimento mínimo de bombeamento*** para A é o menor p que é um comprimento de bombeamento para A. Por exemplo, se $A = 01^*$, o comprimento mínimo de bombeamento é 2. A razão é que a cadeia $s = 0$ está em A, tem comprimento 1, e s não pode ser bombeada, mas qualquer cadeia em A de comprimento 2 ou mais contém um 1 e, portanto, pode ser bombeada pela divisão da mesma de forma que $x = 0$, $y = 1$, e z seja o resto. Para cada uma das linguagens a seguir, dê o comprimento mínimo de bombeamento e justique sua resposta.

R**a.** 0001^*

R**b.** 0^*1^*

c. $001 \cup 0^*1^*$

R**d.** $0^*1^+0^+1^* \cup 10^*1$

e. $(01)^*$

f. ε

g. $1^*01^*01^*$

h. $10(11^*0)^*0$

i. 1011

j. Σ^*

⋆**1.56** Se A é um conjunto de números naturais e k um número natural maior que 1, seja

$$B_k(A) = \{w \mid w \text{ é a representação na base } k \text{ de algum número em } A\}.$$

Aqui, não permitimos 0s à esquerda na representação de um número. Por exemplo, $B_2(\{3, 5\}) = \{11, 101\}$ e $B_3(\{3, 5\}) = \{10, 12\}$. Dê um exemplo de um conjunto A para o qual $B_2(A)$ seja regular, mas $B_3(A)$ não seja. Prove que seu exemplo funciona.

*1.57 Se A é uma linguagem qualquer, seja $A_{\frac{1}{2}-}$ o conjunto de todas as primeiras metades das cadeias em A, de modo que

$$A_{\frac{1}{2}-} = \{x|\ \text{para alguma } y,\ |x| = |y| \text{ e } xy \in A\}.$$

Mostre que, se A é regular, então $A_{\frac{1}{2}-}$ também o é.

*1.58 Se A é uma linguagem qualquer, seja $A_{\frac{1}{3}-\frac{1}{3}}$ o conjunto de todas as cadeias em A com suas terças partes do meio removidas, de forma que

$$A_{\frac{1}{3}-\frac{1}{3}} = \{xz|\ \text{para alguma } y,\ |x| = |y| = |z| \text{ e } xyz \in A\}.$$

Mostre que, se A é regular, então $A_{\frac{1}{3}-\frac{1}{3}}$ não é necessariamente regular.

*1.59 Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e seja h um estado de M denominado sua "casa". Uma ***seqüência sincronizadora*** para M e h é uma cadeia $s \in \Sigma^*$ onde $\delta(q, s) = h$ para todo $q \in Q$. (Aqui estendemos δ para cadeias, de forma que $\delta(q, s)$ é igual ao estado em que M termina quando M inicia no estado q e lê a entrada s.) Digamos que M é ***sincronizável*** se tem uma seqüência sincronizadora para algum estado h. Prove que, se M é um AFD sincronizável de k estados, então ele tem uma seqüência sincronizadora de comprimento no máximo k^3. Você pode encontrar um limite melhor que esse?

1.60 Seja $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Para cada $k \geq 1$, seja C_k a linguagem constituída de todas as cadeias que contêm um $\mathtt{a}$ exatamente k posições a partir da extremidade direita. Assim, $C_k = \Sigma^*\mathtt{a}\Sigma^{k-1}$. Descreva um AFN com $k + 1$ estados que reconheça C_k, ambos em termos de um diagrama de estados e de uma descrição formal.

1.61 Considere as linguagens C_k definidas no Problema 1.60. Prove que para cada k, nenhum AFD pode reconhecer C_k com menos de 2^k estados.

1.62 Seja $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Para cada $k \geq 1$, seja D_k a linguagem constituída de todas as palavras que têm pelo menos um $\mathtt{a}$ entre os últimos k símbolos. Logo, $D_k = \Sigma^*\mathtt{a}(\Sigma \cup \varepsilon)^{k-1}$. Descreva um AFD com no máximo $k + 1$ estados que reconheça D_k, ambos em termos de um diagrama de estados e de uma descrição formal.

1.63
a. Seja A uma linguagem regular infinita. Prove que A pode ser dividida em dois subconjuntos regulares disjuntos infinitos.

b. Sejam B e D duas linguagens. Escreva $B \Subset D$ se $B \subseteq D$ e D contém infinitamente muitas cadeias que não estão em B. Mostre que, se B e D são duas linguagens regulares e $B \Subset D$, então podemos encontrar uma linguagem regular C tal que $B \Subset C \Subset D$.

1.64 Seja N um AFN com k estados que reconhece certa linguagem A.

a. Mostre que, se A é não-vazia, A contém alguma cadeia de comprimento no máximo k.

b. Mostre, dando um exemplo, que a parte (a) não é necessariamente verdadeira se você subsituir ambos os As por $\overline{A}$.

c. Mostre que, se $\overline{A}$ é não-vazia, $\overline{A}$ contém alguma cadeia de comprimento no máximo 2^k.

d. Mostre que o limite dado na parte (c) é quase exato; isto é, para cada k, mostre a existência de um AFN reconhecendo uma linguagem A_k tal que $\overline{A_k}$ é não-vazia e as menores cadeias membros de $\overline{A_k}$ são de comprimento exponencial em k. Aproxime-se do limite apresentado em (c) tanto quanto puder.