Soluções dos Exercícios Propostos no Livro

Introdução aos Fundamentos da Computação: Linguagens e Máquinas

(Ed. Thomson, 2006)

Newton José Vieira
Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais

Belo Horizonte, 30/06/2007

Peço a quem encontrar erros nas soluções a seguir entrar em contato com o autor no endereço nvieira@dcc.ufmg.br. Antecipadamente, agradeço

Capítulo 1

Conceitos Preliminares

1.1 Representação

Nesta seção não há exercícios.

1.2 Prova de Teoremas

1. a) A afirmativa $(\alpha \wedge \beta) \rightarrow \alpha$ só é falsa se $\alpha \wedge \beta$ é verdadeira e α é falsa. Mas a $\alpha \wedge \beta$ não pode ser verdadeira se α é falsa. Assim, $(\alpha \wedge \beta) \rightarrow \alpha$ não pode ser falsa; logo, é válida.

   b) A afirmativa $\alpha \rightarrow (\alpha \vee \beta)$ só é falsa se α é verdadeira e $\alpha \vee \beta$ é falsa. Mas, sendo α verdadeira, $\alpha \vee \beta$ não pode ser falsa. Assim, $\alpha \rightarrow (\alpha \vee \beta)$ não pode ser falsa; logo, é válida.

   c) A afirmativa $(\alpha \wedge \neg\alpha) \rightarrow \beta$ só pode ser falsa se $\alpha \wedge \neg\alpha$ é verdadeira e β é falsa. Mas a afirmativa $\alpha \wedge \neg\alpha$ não pode ser verdadeira, pois é uma contradição. Assim, $(\alpha \wedge \neg\alpha) \rightarrow \beta$ não pode ser falsa; logo, é válida.

   d) A afirmativa $\alpha \rightarrow (\beta \vee \neg\beta)$ só é falsa se α é verdadeira e $\beta \vee \neg\beta$ é falsa. Mas, $\beta \vee \neg\beta$ não pode ser falsa, pois β e $\neg\beta$ não podem ser ambas falsas. Assim, $\alpha \rightarrow (\beta \vee \neg\beta)$ não pode ser falsa; logo, é válida.

   e) A afirmativa $(\alpha \rightarrow \beta) \vee \alpha$ só pode ser falsa se ambas, $\alpha \rightarrow \beta$ e α são falsas. Mas, sendo α falsa, $\alpha \rightarrow \beta$ é verdadeira. Assim, $(\alpha \rightarrow \beta) \vee \alpha$ não pode ser falsa; logo, é válida.

   f) A afirmativa $(\alpha \rightarrow \beta) \vee (\beta \rightarrow \alpha)$ só pode ser falsa se $\alpha \rightarrow \beta$ e $\beta \rightarrow \alpha$ são falsas. Para $\alpha \rightarrow \beta$ ser falsa, α deve ser verdadeira (e β falsa), e para $\beta \rightarrow \alpha$ ser falsa, α deve ser falsa (e β verdadeira); contradição! Assim, a afirmativa original não pode ser falsa; logo, é válida.

2. a) Suponha que $\alpha \rightarrow \beta$ é verdadeira. Segue-se que α é falsa ou β é verdadeira. No primeiro caso, $\neg\alpha$ é verdadeira e, portanto, $\neg\beta \rightarrow \neg\alpha$ é verdadeira. No segundo caso, $\neg\beta$ é falsa e, portanto, $\neg\beta \rightarrow \neg\alpha$ é também verdadeira. Logo, $(\alpha \rightarrow \beta) \Rightarrow (\neg\beta \rightarrow \neg\alpha)$. Suponha, por outro lado, que $\alpha \rightarrow \beta$ é falsa. Segue-se que α é verdadeira e β é falsa. Com isto, $\neg\alpha$ é falsa e $\neg\beta$ é verdadeira e, portanto, $\neg\beta \rightarrow \neg\alpha$ é falsa. Logo, $(\neg\alpha \rightarrow \neg\beta) \Rightarrow (\alpha \rightarrow \beta)$.

   b) Primeiro, mostra-se que $(\alpha \vee \beta) \rightarrow \gamma \Rightarrow [(\alpha \rightarrow \gamma) \wedge (\beta \rightarrow \gamma)]$: Suponha que $(\alpha \vee \beta) \rightarrow \gamma$ é verdadeira. Segue-se que $\alpha \vee \beta$ é falsa ou γ é verdadeira. No primeiro caso, α e β são falsas e, portanto, $\alpha \rightarrow \gamma$ e $\beta \rightarrow \gamma$ são verdadeiras. No segundo caso, sendo

γ verdadeira, $\alpha \to \gamma$ e $\beta \to \gamma$ são também verdadeiras. Assim, se $(\alpha \vee \beta) \to \gamma$ é verdadeira, $(\alpha \to \gamma) \wedge (\beta \to \gamma)$ é verdadeira. Logo, $(\alpha \vee \beta) \to \gamma \Rightarrow [(\alpha \to \gamma) \wedge (\beta \to \gamma)]$. Resta mostrar que $[(\alpha \to \gamma) \wedge (\beta \to \gamma)] \Rightarrow (\alpha \vee \beta) \to \gamma$. Para isto, suponha que $(\alpha \to \gamma) \wedge (\beta \to \gamma)$ é verdadeira; segue-se que $\alpha \to \gamma$ e $\beta \to \gamma$ são verdadeiras. Destas duas segue-se que γ é verdadeira ou α e β são falsas, e, neste último caso, $\alpha \vee \beta$ é falsa. Mas, sendo γ verdadeira ou $\alpha \vee \beta$ falsa, $(\alpha \vee \beta) \to \gamma$ é verdadeira. Portanto, $[(\alpha \to \gamma) \wedge (\beta \to \gamma)] \Rightarrow (\alpha \vee \beta) \to \gamma$.

c) Suponha que $\alpha \to (\beta \wedge \gamma)$ é verdadeira. Segue-se que α é falsa ou $\beta \wedge \gamma$ é verdadeira. No primeiro caso, sendo α falsa, $\alpha \to \beta$ e $\alpha \to \gamma$ são verdadeiras. No segundo caso, sendo β verdadeira, $\alpha \to \beta$ é verdadeira, e sendo γ verdadeira, $\alpha \to \gamma$ é verdadeira. Logo, $(\alpha \to (\beta \wedge \gamma)) \Rightarrow ((\alpha \to \beta) \wedge (\alpha \to \gamma))$. Suponha, agora, que $\alpha \to (\beta \wedge \gamma)$ é falsa. Segue-se que α é verdadeira e $(\beta \wedge \gamma)$ é falsa, ou seja, β é falsa ou γ é falsa. Com isto, $\alpha \to \beta$ é falsa ou $\alpha \to \gamma$ é falsa e, portanto, $(\alpha \to \beta) \wedge (\alpha \to \gamma)$ é falsa. Logo, $\alpha \to (\beta \wedge \gamma) \Rightarrow (\alpha \to \beta) \wedge (\alpha \to \gamma)$.

3. a) Não existe atribuição de valor-verdade para α tal que ambas, α e $\neg\alpha$ sejam verdeiras. Assim, *por vacuidade*, sempre α e $\neg\alpha$ são verdadeiras, γ também é.

   b) Se $\alpha \to \gamma$ é verdadeira, então α é falsa ou γ é verdadeira. No primeiro caso, $\alpha \wedge \beta$ é falsa e, portanto, $(\alpha \wedge \beta) \to \gamma$ é verdadeira. No segundo caso, sendo γ é verdadeira, $(\alpha \wedge \beta) \to \gamma$ é também verdadeira. Conclui-se, então que $\{\alpha \to \gamma\} \Rightarrow (\alpha \wedge \beta) \to \gamma$.

   c) Suponha que $\neg\alpha \to \beta$ e $\neg\beta$ sejam verdadeiras. Neste caso, β é falsa e, portanto, $\neg\alpha$ é falsa. Sendo $\neg\alpha$ falsa, α é verdadeira. Logo, se $\neg\alpha \to \beta$ e $\neg\beta$ são verdadeiras, α é verdadeira, ou seja, $\{\neg\alpha \to \beta, \neg\beta\} \Rightarrow \alpha$.

4. Sejam x e y números reais tais que $x > 0$ e $x < y$. Destas duas últimas segue-se que $x^2 < xy$. Das mesmas, segue-se também que $y > 0$. Desta e do fato de que $x < y$, deduz-se que $xy < y^2$. De $x^2 < xy$ e $xy < y^2$, conclui-se que $x^2 < y^2$.

5. Sejam x e y números reais arbitrários. Suponha que $x^2 + y = 13$ e $y \neq 4$. Deve-se mostrar que, neste caso, $x \neq 3$. Suponha o contrário: $x = 3$. Então, substituindo-se x por 3 em $x^2 + y = 13$, obtém-se: $3^2 + y = 13$ e, assim, $y = 13 - 9 = 4$. Contradição. Logo, se $x^2 + y = 13$ e $y \neq 4$, então $x \neq 3$.

6. Seja x um número real tal que $x > 2$. Seja $k = (x + \sqrt{x^2 - 4})/2$; observe que k é um número real, pois $x > 2$. Além disso,

$$k + \frac{1}{k} = \frac{x + \sqrt{x^2 - 4}}{2} + \frac{1}{(x + \sqrt{x^2 - 4})/2} = \frac{x + \sqrt{x^2 - 4}}{2} + \frac{2}{x + \sqrt{x^2 - 4}}.$$

   Fazendo-se as operações e simplificando-se obtém-se:

$$k + \frac{1}{k} == \frac{x^2 + 2x\sqrt{x^2 - 4} + x^2 - 4 + 4}{2x + 2\sqrt{x^2 - 4}} = x.$$

   Conclui-se, então, que para todo número real x, se $x > 2$, existe um número y tal que $y + (1/y) = x$.

7. Seja x um número natural. A prova será feita pela contrapositiva. Suponha, assim, que $\sqrt{x}$ é um número racional. Neste caso, $\sqrt{x} = p/q$, onde p e q são números naturais primos entre si. Segue-se que $x = p^2/q^2$. Como p e q são primos entre si, p^2 e q^2 são primos entre si, e como x é um número natural q^2 só pode ser 1 e $x = p^2$. Portanto, para todo número natural x, se x não é é um quadrado perfeito, $\sqrt{x}$ não é um número racional.

8. Seja n um número inteiro não divisível por 3. Então $n = 3q + r$, onde q é um inteiro e $r \in \{1, 2\}$. Caso 1: $r = 1$. Tem-se que $n^2 = (3q + 1)^2 = 9q^2 + 6q + 1 = 3(3q^2 + 2q) + 1$. Fazendo-se $k = 3q^2 + 2q$ tem-se que k é um inteiro tal que $n^2 = 3k + 1$, como requerido. Caso 2: $r = 2$. Tem-se que $n^2 = (3q+2)^2 = 9q^2+12q+4 = 3(3q^2+4q+1)+1$. Fazendo-se $k = 3q^2 + 4q + 1$ tem-se que k é um inteiro tal que $n^2 = 3k + 1$.

1.3 Conjuntos

1. a) $A \cap B = \{0, 1, 2, 3, 4, 5\}$.
   b) $C = \{-4, -2, 0, 2, 4, 6, 8\}$.
   c) $D = \{-5, -4, -3, -2, -1, 6, 7, 8\}$.
   d) $\{(0, 7), (2, 7), (4, 7)\}$.

2. Resposta para a primeira pergunta: $A - B = B - A$ se, e somente se, $A = B$. Prova:

   ($\leftarrow$) Suponha que $A = B$. Então $A - B = B - A = \emptyset$.

   ($\rightarrow$) Suponha que $A - B = B - A$. Para provar, primeiramente, que $A \subseteq B$, seja $x \in A$. Suponha que $x \notin B$. Neste caso, $x \in A - B$, e, como $A - B = B - A$, $x \in B - A$. Mas, neste caso, $x \in B$ e $x \notin A$. Contradição. Assim, se $x \in A$, então $x \in B$, ou seja, $A \subseteq B$. Prova-se que $B \subseteq A$ de forma análoga.

   Resposta para a segunda pergunta: $A \cup B = A \cap B$ se, e somente se, $A = B$. Prova:

   ($\leftarrow$) Suponha que $A = B$. Então $A \cup B = A \cap B = A = B$.

   ($\rightarrow$) Suponha que $A \cup B = A \cap B$. Para provar, primeiramente, que $A \subseteq B$, seja $x \in A$. Com isto, $x \in A \cup B$. Neste caso, como $A \cup B = A \cap B$, $x \in A \cap B$. Assim $x \in B$. Portanto, se $x \in A$, então $x \in B$, ou seja, $A \subseteq B$. Prova-se que $B \subseteq A$ de forma análoga.

3. Será mostrado que $A \cup B = A \cup C$ se, e somente se, $(B - C) \cup (C - B) \subseteq A$. (Ou seja, a condição procurada é: a diferença simétrica entre B e C deve estar contida em A, mesmo que $B \neq C$.)

   ($\rightarrow$) Suponha que $A \cup B = A \cup C$. Seja x um elemento arbitrário de $(B - C) \cup (C - B)$. Caso 1: $x \in B - C$, ou seja $x \in B$ e $x \notin C$. Como $x \in B$ e $A \cup B = A \cup C$, segue-se que $x \in A \cup C$. Desta e de $x \notin C$, conclui-se que $x \in A$. Caso 2: análogo. Portanto, se $x \in (B - C) \cup (C - B)$, então $x \in A$. Logo, $(B - C) \cup (C - B) \subseteq A$.

   ($\rightarrow$) Suponha que $(B-C)\cup(C-B) \subseteq A$. Para provar, primeiramente, que $A\cup B \subseteq A\cup C$, seja $x \in A \cup B$. Caso 1: $x \in A$. Então $x \in A \cup C$. Caso 2: $x \in B$. Neste caso, se $x \in C$, $x \in A \cup C$, e se $x \notin C$, $x \in B - C$; e como $B - C \subseteq A$, $x \in A$ e, portanto, $x \in A \cup C$. Assim, se $x \in A \cup B$, então $x \in A \cup C$, ou seja $A \cup B \subseteq A \cup C$. De forma análoga, $A \cup C \subseteq A \cup B$. Portanto, $A \cup C = A \cup B$.

4. a) Basta provar que $\forall x[x \in \overline{A \cup B} \leftrightarrow x \in \overline{A} \cap \overline{B}]$. Seja, então, um elemento x arbitrário. Tem-se:

$$
\begin{aligned}
x \in \overline{A \cup B} &\leftrightarrow x \notin A \cup B && \text{pela definição de complemento}\\
&\leftrightarrow \neg(x \in A \text{ ou } x \in B) && \text{pela definição de união}\\
&\leftrightarrow x \notin A \text{ e } x \notin B && \text{por DeMorgan}\\
&\leftrightarrow x \in \overline{A} \text{ e } x \in \overline{B} && \text{pela definição de complemento}\\
&\leftrightarrow x \in \overline{A} \cap \overline{B} && \text{pela definição de interseção.}
\end{aligned}
$$

b) Suponha que $A \cap B = A \cup B$. Inicialmente, prova-se que $A \subseteq B$. Seja x um elemento arbitrário de A. Neste caso, $x \in A \cup B$; e como $A \cap B = A \cup B$, $x \in A \cap B$. Logo, $x \in B$ e, portanto, $A \subseteq B$. Prova-se que $B \subseteq A$ de forma similar. Portanto, $A = B$. Conclui-se que se $A \cap B = A \cup B$ então $A = B$.

c) Basta provar que $\forall x[x \in (A - B) \cup (B - A) \leftrightarrow x \in (A \cup B) - (A \cap B)]$. Assim, seja um elemento x arbitrário. Tem-se:
($\rightarrow$) Suponha que $x \in (A - B) \cup (B - A)$. Caso 1: $x \in A - B$. Então $x \in A$ e $x \notin B$. Como $x \in A$, $x \in A \cup B$, e como $x \notin B$, $x \notin A \cap B$. Portanto, $x \in (A \cup B) - (A \cap B)$. O caso 2, $x \in B - A$, é similar.
($\rightarrow$) Suponha que $x \in (A \cup B) - (A \cap B)$. Então $x \in A \cup B$ e $x \notin A \cap B$. Segue-se, então, que $x \in A$ ou $x \in B$ e também que $x \notin A$ ou $x \notin B$. Assim, só se pode ter dois casos: o caso em que $x \in A$ e $x \notin B$ e caso em que $x \in B$ e $x \notin A$. Em outras palavras, $x \in A - B$ ou $x \in B - A$. Portanto, $x \in (A - B) \cup (B - A)$.

d) Basta provar que $\forall x[x \in A - (A - B) \leftrightarrow x \in A \cap B]$. Seja, então, um elemento x arbitrário. Tem-se:

$$\begin{aligned}
x \in A - (A - B) &\leftrightarrow x \in A \text{ e } x \notin A - B && \text{pela def. de diferença} \\
&\leftrightarrow x \in A \text{ e } (x \notin A \text{ ou } x \in B) && \text{pela def. de diferença} \\
&\leftrightarrow (x \in A \text{ e } x \notin A) \text{ ou } (x \in A \text{ e } x \in B) && \text{pela distrib. e/ou} \\
&\leftrightarrow x \in A \text{ e } x \in B \\
&\leftrightarrow x \in A \cap B && \text{pela def. de interseção.}
\end{aligned}$$

e) Basta provar que $\forall x[x \in (A - B) - C \leftrightarrow x \in A - (B \cup C]$. Assim, seja um elemento x arbitrário. Tem-se:

$$\begin{aligned}
x \in (A - B) - C &\leftrightarrow x \in A - B \text{ e } x \notin C && \text{pela def. de diferença} \\
&\leftrightarrow x \in A \text{ e } x \notin B \text{ e } x \notin C && \text{pela def. de diferença} \\
&\leftrightarrow x \in A \text{ e } \neg(x \in B \text{ ou } x \in C) && \text{pela lei de De Morgan} \\
&\leftrightarrow x \in A \text{ e } x \notin B \cup C && \text{pela def. de união} \\
&\leftrightarrow x \in A - (B \cup C) && \text{pela def. de diferença.}
\end{aligned}$$

f) Basta provar que $\forall x[x \in (A - B) - C \leftrightarrow x \in (A - C) - (B - C]$. Assim, seja um elemento x arbitrário. Tem-se:

$$\begin{aligned}
x \in (A - B) - (B - C) &\leftrightarrow (x \in A \text{ e } x \notin C) \text{ e } x \notin B - C \\
&\qquad \text{pela def. de diferença} \\
&\leftrightarrow (x \in A \text{ e } x \notin C) \text{ e } (x \notin B \text{ ou } x \in C) \\
&\qquad \text{pela def. de diferença} \\
&\leftrightarrow (x \in A \text{ e } x \notin C \text{ e } x \notin B) \text{ ou } (x \in A \text{ e } x \notin C \text{ e } x \in C) \\
&\qquad \text{pela distr. e/ou} \\
&\leftrightarrow x \in A \text{ e } x \notin C \text{ e } x \notin B \\
&\leftrightarrow x \in A - B \text{ e } x \notin C && \text{pela def. de diferença} \\
&\leftrightarrow x \in (A - B) - C && \text{pela def. de diferença}
\end{aligned}$$

g) Seja um par (x, y) arbitrário. Tem-se:

$$\begin{aligned}
(x, y) \in A \times (B \cap C) &\leftrightarrow x \in A \wedge y \in B \cap C && \text{pela def. de produto} \\
&\leftrightarrow x \in A \wedge y \in B \wedge y \in C && \text{pela def. de interseção} \\
&\leftrightarrow (x \in A \wedge y \in B) \wedge (x \in A \wedge y \in C) \\
&\leftrightarrow x \in A \times B \wedge x \in A \times C && \text{pela def. de produto} \\
&\leftrightarrow x \in (A \times B) \cap (A \times C) && \text{pela def. de interseção}
\end{aligned}$$

h) Seja um par (x, y) arbitrário. Tem-se:

$$\begin{aligned}
(x,y) \in (A \cap B) \times (C \cap D) &\leftrightarrow x \in (A \cap B) \wedge y \in (C \cap D) \\
&\qquad \text{pela def. de produto} \\
&\leftrightarrow x \in A \wedge x \in B \wedge y \in C \wedge y \in D \\
&\qquad \text{pela def. de interseção} \\
&\leftrightarrow (x \in A \wedge y \in C) \wedge (x \in B \wedge y \in D) \\
&\qquad \text{por associatividade} \\
&\leftrightarrow x \in A \times C \wedge x \in B \times D \\
&\qquad \text{pela def. de produto} \\
&\leftrightarrow x \in (A \times C) \cap (B \times D) \\
&\qquad \text{pela def. de interseção}
\end{aligned}$$

5. As partições de $\{1,2\}$ são: $\{\{1\},\{2\}\}$ e $\{\{1,2\}\}$. As partições de $\{1,2,3\}$ são: $\{\{1\},\{2\},\{3\}\}$, $\{\{1\},\{2,3\}\}$, $\{\{2\},\{1,3\}\}$, $\{3\},\{1,2\}\}$ e $\{\{1,2,3\}\}$. E as partições de $\{1,2,3,4\}$ são: $\{\{1\},\{2\},\{3\},\{4\}\}$, $\{\{1\},\{2\},\{3,4\}\}$, $\{\{1\},\{3\},\{2,4\}\}$, $\{\{1\},\{4\},\{2,3\}\}$, $\{\{2\},\{3\},\{1,4\}$, $\{\{2\},\{4\},\{1,3\}\}$, $\{\{3\},\{4\},\{1,2\}\}$, $\{\{1\},\{2,3,4\}\}$, $\{\{2\},\{1,3,4\}\}$, $\{\{3\},\{1,2,4\}\}$, $\{\{4\},\{1,2,3\}\}$, $\{\{1,2\},\{3,4\}\}$, $\{\{1,3\},\{2,4\}\}$, $\{\{1,4\},\{2,3\}\}$ e $\{\{1,2,3,4\}\}$.

   Segue a função $part(A)$, que retorna o conjunto das partições do conjunto (finito) A:

   Entrada: um conjunto finito A.
   Saída: o conjunto das partições de A.
   se $A = \emptyset$ **então** retorne $\emptyset$ **fimse**;
   $R \leftarrow \emptyset$;
   selecione um elemento $a \in A$;
   para cada $B \subset A$ tal que $a \in B$ **faça**
       para cada $X \in part(A - B)$ **faça**
           $R \leftarrow R \cup \{\{B\} \cup X\}$;
       fimpara
   fimpara;
   retorne $R \cup \{\{A\}\}$.

1.4 Relações

1. a) A relação $\subset$ é transitiva, não é reflexiva nem simétrica.
   b) A relação não é reflexiva, nem simétrica nem transitiva.
   c) A relação é reflexiva e transitiva, e não é simétrica.
   d) A relação é reflexiva e transitiva, e não é simétrica.
   e) A relação é reflexiva, simétrica e transitiva.

2. Sejam $R_1 \subseteq A^2$ e $R_2 \subseteq B^2$

   a) Se R_1 e R_2 são reflexivas, $R_1 \cup R_2$ é reflexiva sobre $A \cup B$: Seja $x \in A \cup B$. Se $x \in A$, $(x,x) \in R_1$, pois R_1 é reflexiva, e se $x \in B$, $(x,x) \in R_2$, pois R_2 é reflexiva. Assim, $(x,x) \in R_1 \cup R_2$.

Se R_1 e R_2 são reflexivas, $R_1 \cap R_2$ é reflexiva sobre $A \cap B$: Seja $x \in A \cap B$. Como $x \in A$ e $x \in B$, $(x, x) \in R_1$ e $(x, x) \in R_2$ pois R_1 e R_2 são reflexivas. Assim, $(x, x) \in R_1 \cap R_2$.

b) Se R_1 e R_2 são simétricas, $R_1 \cup R_2$ é simétrica sobre $A \cup B$: Sejam $x, y \in A \cup B$. Se $x, y \in A$ e $(x, y) \in R_1$, então $(y, x) \in R_1$ e, analogamente, se $x, y \in B$ e $(x, y) \in R_2$, então $(y, x) \in R_2$, pois R_1 e R_2 são simétricas. Por outro lado, se $x \in A - B$ e $y \in B - A$, ou vice-versa, ou seja, x e y não pertencem ao mesmo conjunto, não se pode ter $(x, y) \in R_1$ nem $(x, y) \in R_2$. Assim, em qualquer caso, se $x, y \in A \cup B$ e $(x, y) \in R_1 \cup R_2$, $(y, x) \in R_1 \cup R_2$.

Se R_1 e R_2 são simétricas, $R_1 \cap R_2$ é simétrica sobre $A \cap B$: Sejam $x, y \in A \cap B$. Como $x, y \in A$ e $x, y \in B$ $(y, x) \in R_1$ e $(y, x) \in R_2$, pois R_1 e R_2 são simétricas. Portanto, se $x, y \in A \cap B$ e $(x, y) \in R_1 \cap R_2$, $(y, x) \in R_1 \cap R_2$.

c) Se R_1 e R_2 são transitivas, $R_1 \cup R_2$ pode não ser transitiva. Por exemplo, $R_1 = \{(a, b)\}$ sobre $\{a, b\}$ e $R_2 = \{(b, c)\}$ sobre $\{b, c\}$; tem-se que $R_1 \cup R_2 = \{(a, b), (b, c)\}$ sobre $\{a, b, c\}$, que não é transitiva, pois $(a, c) \notin R_1 \cup R_2$.

A relação $R_1 \cap R_2$ é transitiva sobre $A \cap B$: Dado que $(x, y) \in R_1 \cap R_2$ e $(y, z) \in R_1 \cap R_2$, segue-se que $(x, y) \in R_1$ e $(y, z) \in R_1$, e também que $(x, y) \in R_2$ e $(y, z) \in R_2$. Como R_1 e R_2 são transitivas, deduz-se que $(x, z) \in R_1$ e $(x, z) \in R_2$. Logo, $(x, z) \in R_1 \cap R_2$.

3. $(\rightarrow)$ Suponha que R é simétrica. Sejam x e y elementos arbitrários tais que $(x, y) \in R$. Como R é simétrica, segue-se que $(y, x) \in R$. Desta última, segue-se que $(x, y) \in R^{-1}$. Logo, se $(x, y) \in R$, então $(x, y) \in R^{-1}$, ou seja, $R \subseteq R^{-1}$. De forma similar, mostra-se que $R^{-1} \subseteq R$. Portanto, $R = R^{-1}$.

   $(\leftarrow)$ Suponha que $R = R^{-1}$ e seja $(x, y) \in R$. Segue-se que $(x, y) \in R^{-1}$ e, portanto, $(y, x) \in R$. Logo, se $(x, y) \in R$ então $(y, x) \in R$. Como (x, y) é um elemento arbitrário de R, conclui-se que R é simétrica.

4. $(\rightarrow)$ Suponha que R é reflexiva sobre um conjunto A. Seja x um elemento arbitrário de A. Como R é reflexiva, segue-se que $(x, x) \in R$. Segue-se, pela definição de inversa, que $(x, x) \in R^{-1}$. Logo, R^{-1} é reflexiva sobre A.

   $(\leftarrow)$ Suponha que R^{-1} é reflexiva sobre um conjunto A. Seja x um elemento arbitrário de A. Como R^{-1} é reflexiva, segue-se que $(x, x) \in R^{-1}$. Segue-se, pela definição de inversa, que $(x, x) \in R$. Logo, R é reflexiva sobre A.

5. $(\rightarrow)$ Suponha que R é anti-simétrica. Sejam x e y elementos arbitrários de A tais que $(x, y) \in R \cap R^{-1}$, ou seja, $(x, y) \in R$ e $(x, y) \in R^{-1}$. Desta última, segue-se que $(y, x) \in R$. Como R é anti-simétrica, $(x, y) \in R$ e $(y, x) \in R$, segue-se que $x = y$ e, portanto, $(x, y) \in \iota_A$. Assim, $R \cap R^{-1} \subseteq \iota_A$.

   $(\leftarrow)$ Suponha que $R \cap R^{-1} \subseteq \iota_A$. Sejam x e y tais que $(x, y) \in R$ e $(y, x) \in R$. Como $(y, x) \in R$, $(x, y) \in R^{-1}$, e, portanto, $(x, y) \in R \cap R^{-1}$. Como $R \cap R^{-1} \subseteq \iota_A$, segue-se que $(x, y) \in \iota_A$ e, portanto, $x = y$. Conclui-se que se $(x, y) \in R$ e $(y, x) \in R$, então $x = y$. Portanto, R é anti-simétrica.

6. Nesta questão, considere que se o denominador nas operações de divisão for 0, o resultado é sempre o mesmo ("indefinido") e diferente de qualquer outro resultado.

   Como $x_1/x_2 = x_1/x_2$ para quaisquer $x_1, x_2 \in \mathbf{N}$, R é reflexiva. Suponha que $x_1/x_2 = y_1/y_2$, para $x_1, x_2, y_1, y_2 \in \mathbf{N}$. Segue-se que $y_1/y_2 = x_1/x_2$. Logo, R é simétrica. Supondo-se que $x_1/x_2 = y_1/y_2$ e $y_1/y_2 = z_1/z_2$, conclui-se que $x_1/x_2 = z_1/z_2$. Portanto, R é transitiva. Como R é reflexiva, simétrica e transitiva, R é uma relação de equivalência.

Para cada par $(x_1, x_2) \in \mathbf{N}^2$ há a classe de equivalência $\{(y_1, y_2) \in \mathbf{N}^2 \mid y_1/y_2 = x_1/x_2\}$, ou seja a cada número racional, r, corresponde a classe de equivalência constituída dos pares de naturais (x_1, x_2) tais que $r = x_1/x_2$.

7. a) Fecho reflexivo: $\{(a,a),(c,c),(d,d),(e,e)\} \cup R$.
   b) Fecho simétrico: $\{(a,d),(b,a)\} \cup R$.
   c) Fecho transitivo: $\{(c,a),(c,b),(c,c),(d,b),(d,d)\} \cup R$.
   d) Fecho reflexivo e simétrico: $\{(a,a),(c,c),(d,d),(e,e),(a,d),(b,a)\} \cup R$.
   e) Fecho reflexivo e transitivo: $\{(a,a),(c,c),(d,d),(e,e),(c,a),(c,b),(d,b)\} \cup R$.
   f) Fecho simétrico e transitivo: $\{(a,d),(b,a),(c,a),(c,b),(c,c),(d,b),(d,d)\} \cup R$.
   g) Fecho reflexivo, simétrico e transitivo:
   $\{(a,a),(c,c),(d,d),(e,e),(a,d),(b,a),(c,a),(c,b),(d,b)\} \cup R$.

1.5 Funções

1. O número máximo para $R \subseteq A \times B$ ser uma função é $|R| = |A|$, já que não se pode ter $(x,y) \in f$ e $(x,z) \in f$ se $y \neq z$.

2. Sejam $f : A \to B$, $g : C \to D$ e $h = f \cup g$. Então:

$$h : A \cup C \to B \cup D \text{ se, e somente se, } \forall x \in A \cap C\, f(x) = g(x).$$

   Prova:

   $(\to)$ Suponha que $h : A \cup C \to B \cup D$. Seja um elemento arbitrário x de $A \cap C$. Como $h = f \cup g$ e $h : A \cup C \to B \cup D$, segue-se que $h(x) = f(x) = g(x)$.

   $(\leftarrow)$ Suponha que $\forall x \in A \cap C\, f(x) = g(x)$. Seja y um elemento arbitrário de $A \cup C$. Para mostrar existe um único $z \in B \cup D$ tal que $(y,z) \in h$, serão considerados 3 casos:
   Caso 1: $y \in A - C$. Como $f : A \to B$, $h = f \cup g$ e $y \in A$, um z tal que $(y,z) \in h$ é $z = f(y)$. E como $g : C \to D$, $h = f \cup g$ e $y \notin C$, tal z é único.
   Caso 2: $y \in C - A$. Como $g : C \to D$, $h = f \cup g$ e $y \in C$, um z tal que $(y,z) \in h$ é $z = g(y)$. E como $f : A \to B$, $h = f \cup g$ e $y \notin A$, tal z é único.
   Caso 3: $y \in A \cap C$. Como $f : A \to B$, $g : C \to D$, $h = f \cup g$ e $\forall x \in A \cap C\, f(x) = g(x)$, o único z tal que $(y,z) \in h$ é $z = f(y) = g(y)$.

3. a) f não é injetora, pois $f(0) = f(1) = 0$. É sobrejetora, pois $f(2n) = n$ para todo $n \in \mathbf{N}$.
   b) g é injetora, pois se $g(n) = g(k)$, então $k = n$. Ela não é sobrejetora, pois não existe $n \in \mathbf{N}$ tal que $g(n) = 2$: $g(1) = 1$ e $g(2) = 3$ e g é crescente.
   c) h é injetora: Suponha que $h(n) = h(k)$. Se n e k forem ambos pares ou ambos ímpares, para que $n-1 = k-1$ ou $n+1 = k+1$, deve-se ter que $n = k$. Por outro lado se um é par e o outro é ímpar, para que $n+1 = k-1$, deve-se ter que $n+2 = k$; mas, para que isto seja verdade, n e k não podem ser um par e o outro ímpar!
   h é sobrejetora, pois $h(n+1) = n$ para todo número n par a partir de 0, e $h(n-1) = n$ para todo número n ímpar a partir de 1.
   Portanto, h é bijetora. Da última sentença acima, vê-se que a $h^{-1} = h$.

4. Sejam $|A| = n$ e $|B| = k$.

a) Para cada $a \in A$, $f(a)$ pode ser qualquer elemento de B. Assim, existem k^n funções totais possíveis.

b) Para cada $a \in A$, $f(a)$ pode ser indefinido ou qualquer elemento de B. Assim, existem $(k+1)^n$ funções parciais possíveis.

c) Se $n > k$, não há função injetora. Suponha que $n \leq k$. Escolhido um elemento de B para ser $f(a)$, ele não pode mais ser $f(b)$ para $b \neq a$. Assim, existem $P(k,n) = k.(k-1)\dots(k-n+1) = k!/(k-n)!$ funções injetoras possíveis, se $n \leq k$.

d) Se $n < k$, não há função sobretora. Suponha que $n \geq k$. Deve ser escolhido um elemento de eA tal que $f(a)$ seja um elemento de B, para cada elemento de B. Para isto existem $P(n,k) = n.(n-1)\dots(n-k+1) = n!/(n-k)!$ possibilidades. Para cada uma delas, os $n-k$ elementos restantes de A podem, cada um deles, ser mapeado para qualquer um dos elementos de B: são k^{n-k} possibilidades. Assim, existem $(n!.k^{n-k})/(n-k)!$ funções sobrejetoras possíveis, se $n \geq k$.

e) Quando $n = k$, existe uma única função bijetora. Se $n \neq k$, não existe função bijetora.

5. a) Sejam x e y elementos arbitrários de A tais que $x \neq y$. Como f é injetora, $f(x) \neq f(y)$. E como g é injetora, $g(f(x)) \neq g(f(y))$. Portanto, se $x \neq y$, então $g(f(x)) \neq g(f(y))$. Conclui-se que $g \circ f$ é injetora.

   b) Seja $y \in C$ arbitrário. Como g é sobrejetora, existe $z \in B$ tal que $g(z) = y$. E como f é sobrejetora, existe $x \in A$ tal que $f(x) = z$, ou ainda, $g(f(x)) = y$. Logo, para todo $y \in C$ existe $x \in A$ tal que $g(f(x)) = y$. Portanto, $g \circ f$ é sobrejetora.

   c) Dos dois resultados acima segue-se que $g \circ f$ é bijetora.

6. Seja $f : A \to B$. Seja $C = \{b \in B \,|\, f(a) = b$ para algum $a \in A\}$. Seja a função sobrejetora $g : A \to C$ tal que $g(a) = f(a)$ para todo $a \in A$. Seja a função injetora $h : C \to B$ tal que $h(c) = c$ para todo $c \in C$. Então $f = h \circ g$.

7. Como $g \circ f = f \circ g$, $g(f(n)) = f(g(n))$ para todo $n \in \mathbf{R}$. Logo, $c(an+b)+d = a(cn+d)+b$, ou seja, $acn + bc + d = acn + ad + b$, ou ainda, $bc + d = ad + b$. Tem-se, então, que $b(c-1) = d(a-1)$.

1.6 Conjuntos Enumeráveis

1. a) O conjunto $X = \{n \in \mathbf{N} \,|\, n \,mod\, 10 = 0\}$ é enumerável, pois é um subconjunto de $\mathbf{N}$. Uma função bijetora seria $g : \mathbf{N} \to X$, tal que $g(n) = 10n$.

   b) O conjunto $\mathbf{N}^3 = \{(n_1, n_2, n_3) \,|\, n_1, n_2, n_3 \in \mathbf{N}\}$ é enumerável. Uma função bijetora seria $h : \mathbf{N}^3 \to \mathbf{N}$, tal que $h(n_1, n_2, n_3) = f(f(n_1, n_2), n_3)$, onde $f : \mathbf{N}^2 \to \mathbf{N}$ é tal que $f(i,j) = (i+j)(i+j+1)/2 + i$.

   c) O conjunto $W = \{n \in \mathbf{R} \,|\, 0 < n < 1\}$ não é enumerável, como será mostrado por contradição. Suponha que W é enumerável. Seja, então, r_0, r_1, r_2, ... uma enumeração dos elementos de W. Será denotado por $d_i(r)$ o i-ésimo dígito após a vírgula da expansão decimal do número r de W. Por exemplo, $d_1(0,53) = 5$, $d_7(0,53) = 0$. Seja o número k entre 0 e 1 tal que $d_i(k) = d_i(r_i) + 5 \,mod\, 10$ para $i \in \mathbf{N}$. Tal número difere de qualquer r_i no i-ésimo dígito após a vírgula. Logo $k \neq r_i$ para $i \in \mathbf{N}$. Contradição; W não é enumerável.

2. Suponha que tal conjunto seja enumerável. Então existe uma enumeração $f_0, f_1, f_2, \ldots$ de todas as funções de $\mathbf{N}$ para $\{0, 1\}$. Seja a função $g\colon \mathbf{N} \to \{0, 1\}$ tal que $g(n) = 1 - f_n(n)$. Ora, $g(n) \neq f_n(n)$ para todo $n \in \mathbf{N}$, e, portanto, $g \neq f_n$ para todo $n \in \mathbf{N}$. Logo, a suposição da enumerabilidade das funções não se sustenta.

3. Suponha que tal conjunto seja enumerável. Então existe uma enumeração $f_0, f_1, f_2, \ldots$ de todas as funções de $\mathbf{N}$ para $\mathbf{N}$ monotônicas crescentes. Seja a função $h\colon \mathbf{N} \to \mathbf{N}$ tal que $h(n) = 1 + \sum_{k=0}^{n} f_k(k)$. Veja que, como as funções f_k são monotônicas crescentes, então $f_k(k) > 0$ para $k > 0$. Assim, h, além de diferente de f_n para todo $n \in \mathbf{N}$. é monotônica crescente. Logo, a suposição da enumerabilidade das funções monotônicas crescentes é incorreta.

4. Seja um conjunto enumerável arbitrário A e suponha que $B \subseteq A$. Se B é finito, é contável, por definição. Suponha que B é infinito. Será mostrado que B é enumerável construindo-se uma função bijetora $g : \mathbf{N} \to B$. Como A é enumerável, existe uma função bijetora $f : \mathbf{N} \to A$; assim sendo, pode-se dizer que $A = \{f(k) \,|\, k \in \mathbf{N}\}$. A função g pode ser construída da seguinte forma:

   - $g(0) = f(m)$, onde m é o *menor* número natural tal que $f(k) \in B$;
   - para $i > 0$, $g(i) = f(k)$, onde k é o *menor* número natural tal que $f(k) \in B$ e $k > j$, sendo j tal que $g(i - 1) = f(j)$.

5. Como todo subconjunto de conjunto enumerável é contável (questão 4), sabe-se que $A - B$ é contável. Como $(A - B) \cup B = A \cup B$, basta mostrar que $(A - B) \cup B$ é contável, sendo que $(A - B) \cap B = \emptyset$. No caso em ambos $A - B$ e B são finitos, $(A - B) \cup B$ é finito e, portanto, contável. No caso em que um deles é finito e o outro não, seja $\{a_0, a_1, \ldots, a_m\}$ o conjunto finito e suponha que existe uma função bijetora f de $\mathbf{N}$ para o conjunto infinito. Então, pode-se construir uma função bijetora $h : \mathbf{N} \to A \cup B$, onde $h(k) = a_k$ para $0 \leq k \leq m$ e $h(k) = f(k - m - 1)$ para $k \geq m + 1$. Agora, se ambos $A - B$ e B são infinitos, sejam $f : \mathbf{N} \to A - B$ e $g : \mathbf{N} \to B$ bijetoras. Então pode-se construir uma função bijetora $h : \mathbf{N} \to A \cup B$, onde $h(2k) = f(k)$ e $h(2k - 1) = g(k)$ para $k \in \mathbf{N}$.

   A interseção é subconjunto dos dois conjuntos. Logo é contável (questão 4).

   O produto de conjuntos finitos é finito. Se um dos conjuntos é finito e o outro não, seja $\{a_0, a_1, \ldots, a_m\}$ o conjunto finito e suponha que existe uma função bijetora f de $\mathbf{N}$ para o conjunto infinito. Sem perda de generalidade, seja A o conjunto finito. Então, pode-se construir uma função bijetora $h : A \times B \to \mathbf{N}$, onde $h(i, j) = j(m + 1) + i$ para $0 \leq i \leq m$ e $j \in \mathbf{N}$. Se os dois conjuntos forem infinitos, sejam $f : A \to \mathbf{N}$ e $g : B \to \mathbf{N}$ bijetoras. A função $h : A \times B \to \mathbf{N}$, onde $h(f(a), g(b)) = (f(a) + g(b))(f(a) + g(b) + 1)/2 + f(a)$ é bijetora.

6. Seja $|F| = n$. A cardinalidade do conjunto das funções totais $f : F \to E$ é igual à do conjunto E^n: existe uma função bijetora que, a cada função f faz corresponder a n-upla $(f(a_1), f(a_2), \ldots, f(a_n))$, onde a_1, a_2, $\ldots$, a_n são os elementos de F. Como E^n é enumerável, segue-se que o conjunto em questão é enumerável.

1.7 Definições Recursivas

1. Definição recursiva de $f : \mathbf{N} \to \mathbf{N}$, tal que $f(n) = \sum_{k=1}^{n} k$:

   a) $f(1) = 1$;

b) $f(n) = f(n-1) + n$ para $n > 1$.

2. Seja um universo U. O número de elementos do conjunto potência de um conjunto finito de elementos de U , pode ser definido recursivamente assim:

   a) $|\mathcal{P}(\emptyset)| = 1$;

   b) $|\mathcal{P}(X \cup \{a\})| = 2|\mathcal{P}(X)|$ para todo $a \in U$.

3. Definição recursiva da representação de números binários sem zeros à esquerda, $r : \mathbf{N} \to B$, onde B é o conjunto das seqüências de dígitos binários:

   a) $r(0) = 0$, $r(1) = 1$;

   b) $r(n) = r(\lfloor n/2 \rfloor)(n \textit{ mod } 2)$ para $n > 1$.

   Assim, por exemplo, $r(5) = r(2)1 = r(1)01 = 101$.

4. Definição recursiva da seqüência de Fibonacci:

   a) $a_0 = 0$ e $a_1 = 1$;

   b) $a_n = a_{n-2} + a_{n-1}$ para $n \geq 2$.

5. Definição recursiva de multiplicação sobre $\mathbf{N}$:

   a) $m \times 0 = 0$ para todo $m \in \mathbf{N}$;

   b) $m \times s(n) = m + (m \times n)$ para $m, n \in \mathbf{N}$.

6. Definição recursiva de LP′, semelhante a LP, porém com omissão e/ou excesso de parênteses:

   a) cada variável proposicional pertence a LP′;

   b) se $\alpha, \beta \in$ LP′, então pertencem a LP′: $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \to \beta$, $\alpha \leftrightarrow \beta$ e (α).

1.8 Indução Matemática

1. Só existe um conjunto de 0 elementos: $\emptyset$; e $2^0 = 1 = |\{\emptyset\}| = |\mathcal{P}(\emptyset)|$. Seja $n \geq 0$. Suponha, como hipótese de indução, que $|\mathcal{P}(A)| = 2^n$ para conjuntos A de n elementos. Seja um conjunto B de $n+1$ elementos e a um de seus elementos. Tem-se: $\mathcal{P}(B) = \mathcal{P}(B - \{a\}) \cup \{X \cup \{a\} \,|\, X \in \mathcal{P}(B - \{a\})\}$. Pela hipótese de indução, $|\mathcal{P}(B - \{a\})| = 2^n$, e como $\mathcal{P}(B - \{a\})$ e $\{X \cup \{a\} \,|\, X \in \mathcal{P}(B - \{a\})\}$ têm o mesmo número de elementos, $|\mathcal{P}(B)| = 2^n + 2^n = 2^{n+1}$.

2. a) $\sum_{k=0}^{0} k^2 = 0^2 = 0 = 0(0+1)(2 \times 0 + 1)/6$. Seja n um número natural arbitrário. Suponha, como hipótese de indução, que $\sum_{k=0}^{n} k^2 = n(n+1)(2n+1)/6$. Basta provar, então, que $\sum_{k=0}^{n+1} k^2 = (n+1)(n+2)(2n+3)/6$. De fato:

   $$\begin{aligned}\textstyle\sum_{k=0}^{n+1} k^2 &= [\textstyle\sum_{k=0}^{n} k^2] + (n+1)^2 \\ &= [n(n+1)(2n+1)/6] + (n+1)^2 \quad \text{pela hipótese de indução} \\ &= [n(n+1)(2n+1) + 6(n+1)^2]/6 \\ &= (n+1)[n(2n+1) + 6(n+1)]/6 \\ &= (n+1)(2n^2 + 7n + 6)/6 \\ &= (n+1)(n+2)(2n+3)/6.\end{aligned}$$

   b) $\sum_{k=0}^{0} k^3 = 0^3 = 0 = 0^2 = [0(0+1)/2]^2$. Seja $n \geq 0$ arbitrário. Suponha, como hipótese de indução, que $\sum_{k=0}^{n} k^3 = [n(n+1)/2]^2$. Basta provar, então, que $\sum_{k=0}^{n+1} k^3 = [(n+1)(n+2)/2]^2$. De fato:

$$
\begin{aligned}
\textstyle\sum_{k=0}^{n+1} k^3 &= [\textstyle\sum_{k=0}^{n} k^3] + (n+1)^3 \\
&= [n(n+1)/2]^2 + (n+1)^3 \qquad \text{pela hipótese de indução} \\
&= [n^2(n+1)^2 + 4(n+1)^3]/4 \\
&= [(n+1)^2(n^2+4n+4)]/4 \\
&= [(n+1)^2(n+2)^2]/4 \\
&= [(n+1)(n+2)/2]^2.
\end{aligned}
$$

c) Inicialmente, obsereve que: $2^{2\times 0} - 1 = 2^0 - 1 = 1 - 1 = 0$, que é divisível por 3. Seja $n \geq 0$. Suponha, como hipótese de indução, que $2^{2n} - 1$ é divisível por 3. Então: $2^{2(n+1)} - 1 = 2^{2n+2} - 1 = 2^{2n} \times 2^2 - 1 = 4 \times 2^{2n} - 1 = 3 \times 2^{2n} + 2^{2n} - 1$. Como 3×2^{2n} é divisível por 3 e, pela hipótese de indução, $2^{2n} - 1$ também é divisível por 3, segue-se que $2^{2(n+1)} - 1$ é divisível por 3.

d) $0^3 - 0 = 0$, que é divisível por 6. Seja $n \geq 0$. Hipótese de indução: $n^3 - n$ é divisível por 6. Deve-se provar que $(n+1)^3 - (n+1)$ é divisível por 6. Tem-se: $(n+1)^3-(n+1) = n^3+3n^2+3n+1-n-1 = (n^3-n)+(3n^2+3n) = (n^3-n)+3n(n+1)$. Como $n(n+1)$ é sempre um número par, $3n(n+1)$ é divisível por 6. E como, pela hipótese de indução, $n^3 - n$ também é divisível por 6, segue-se que $(n+1)^3 - (n+1)$ é divisível por 6.

e) $7^0 - 1 = 1 - 1 = 0$, que é divisível por 6. Seja $n \geq 0$. Suponha, como hipótese de indução, que $7^n - 1$ é divisível por 6. Tem-se: $7^{n+1} - 1 = 7^n - 1 + 6 \times 7^n$. Como 6×7^n é divisível por 6 e, pela hipótese de indução, $7^n - 1$ é divisível por 6, conclui-se que $7^{n+1} - 1$ é divisível por 6.

3. a) Tem-se: $\sum_{k=1}^{1}[k(k+1)] = 1(1+1) = 2 = 1(2)(3)/3 = 1(1+1)(1+2)/3$. Seja $n \geq 1$. Suponha, como hipótese de indução, que $\sum_{k=1}^{n}[k(k+1)] = n(n+1)(n+2)/3$. Basta, então, provar que $\sum_{k=1}^{n+1}[k(k+1)] = (n+1)(n+2)(n+3)/3$. De fato:

$$
\begin{aligned}
\textstyle\sum_{k=1}^{n+1}[k(k+1)] &= [\textstyle\sum_{k=1}^{n} k(k+1)] + (n+1)(n+2) \\
&= [n(n+1)(n+2)/3] + (n+1)(n+2) \quad \text{pela hipótese de indução} \\
&= [n(n+1)(n+2) + 3(n+1)(n+2)]/3 \\
&= (n+1)(n+2)(n+3)/3.
\end{aligned}
$$

b) $\sum_{k=1}^{1} 2^k = 2^1 = 2 = 2\times 1 = 2(2-1) = 2(2^1-1)$. Seja $n \geq 1$. Suponha, como hipótese de indução, que $\sum_{k=1}^{n} 2^k = 2(2^n - 1)$. Tem-se: $\sum_{k=1}^{n+1} 2^k = [\sum_{k=1}^{n} 2^k] + 2^{n+1} = 2(2^n - 1) + 2^{n+1}$, pela hipótese de indução. Logo, $\sum_{k=1}^{n+1} 2^k = 2^{n+1} - 2 + 2^{n+1} = 2(2^{n+1} - 1)$, como requerido.

c) Inicialmente, veja que $\sum_{k=1}^{1}[1/k(k+1)] = 1/(1+1)$. Seja $n \geq 1$ arbitrário, e suponha, como hipótese de indução, que $\sum_{k=1}^{n}[1/k(k+1)] = n/(n+1)$. Então:

$$
\begin{aligned}
\textstyle\sum_{k=1}^{n+1}[1/k(k+1)] &= [\textstyle\sum_{k=1}^{n}[1/k(k+1)]] + 1/(n+1)(n+2) \\
&= [n/(n+1)] + 1/(n+1)(n+2) \quad \text{pela hipótese de indução} \\
&= [n(n+2)+1]/[(n+1)(n+2)] \\
&= [n^2+2n+1]/[(n+1)(n+2)] \\
&= (n+1)^2/[(n+1)(n+2)] \\
&= (n+1)/(n+2).
\end{aligned}
$$

Logo, pelo princípio de indução, $\sum_{k=1}^{n}[1/k(k+1)] = n/(n+1)$ para todo $n \geq 1$.

d) $1^3+(1+1)^3+(1+2)^3 = 1+8+27 = 36$, que é divisível por 9. Seja um número natural n arbitrário maior ou igual a 1. Suponha, como hipótese de indução, que $n^3+(n+1)^3+(n+2)^3$ é divisível por 9. Deve-se mostrar que $(n+1)^3+(n+2)^3+(n+3)^3$ é divisível por 9. Tem-se que $(n+1)^3+(n+2)^3+(n+3)^3 = (n+1)^3+(n+2)^3+n^3+9n^2+27n+27$, ou seja, $(n+1)^3+(n+2)^3+(n+3)^3 = n^3+(n+1)^3+(n+2)^3+9(n^2+3n+3)$. Deduz-se, então, aplicando-se a hipótese de indução, que $(n+1)^3+(n+2)^3+(n+3)^3$ é divisível por 9.

4. Seja n um número natural arbitrário. Suponha, como hipótese, que

$$F(k) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right]$$

para $k < n$. Serão considerados 3 casos:

Caso 1: $n = 0$. Neste caso, $F(0) = 0$ por definição, e $(1/\sqrt{5})[((1+\sqrt{5})/2)^0 - ((1-\sqrt{5})/2)^0] = 1-1 = 0$.

Caso 2: $n = 1$. Neste caso, $F(1) = 1$ por definição, e $(1/\sqrt{5})[((1+\sqrt{5})/2)^1 - ((1-\sqrt{5})/2)^1] = (1+\sqrt{5}-1+\sqrt{5})/(2\sqrt{5}) = 1$.

Caso 3: $n \geq 2$. Neste caso, $F(n) = F(n-1)+F(n-2)$ por definição. Aplicando-se a hipótese de indução, obtém-se:

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\right] + \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\right].$$

Assim,

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2}+1\right) - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\left(\frac{1-\sqrt{5}}{2}+1\right)\right].$$

Verifica-se facilmente que:

$$\frac{1+\sqrt{5}}{2}+1 = \left(\frac{1+\sqrt{5}}{2}\right)^2 \text{ e } \frac{1-\sqrt{5}}{2}+1 = \left(\frac{1-\sqrt{5}}{2}\right)^2.$$

Portanto,

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

5. Será usada indução forte sobre o número de conectivos. Assim, seja $n \geq 0$, e suponha que o resultado valha para sentenças com menos de n conectivos. Deve-se provar, então, que o resultado vale para sentenças com n conectivos. Considera-se dois casos:

   (a) $n = 0$. Uma sentença sem conectivos é uma variável proposicional, e esta tem apenas dois prefixos: λ e ela mesma. Em ambos o número de abre e fecha parênteses é zero.

   (b) $n > 0$. Uma sentença com conectivos é da forma $\neg\alpha$ ou $(\alpha \oplus \beta)$, onde $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

   i) $\neg\alpha$. Como α tem $n-1$ conectivos, o resultado vale para α, pela hipótese de indução. Segue-se que vale também para $\neg\alpha$, que não tem outros parênteses que os de α.

   ii) $(\alpha \oplus \beta)$. Como α e β têm menos de n conectivos, o resultado vale para ambos, pela hipótese de indução. Segue-se que o resultado vale também para $\alpha \oplus \beta$. E o resultado continua valendo ao se colocar os parênteses externos.

1.9 Grafos

1. Isto segue do fato de que a soma dos graus dos vértices é par: em um grafo sem arestas tal soma é zero, e cada aresta acrescenta 2 unidades à soma.

2. Será feita uma prova por indução sobre o número de vértices. A menor árvore, que é da forma $(\{v\}, \emptyset, v)$, tem um vértice e nenhuma aresta. Seja um número $n \geq 1$ e suponha, como hipótese de indução, que a proposição é verdadeira para árvores com n vértices. Uma árvore com $n+1$ vértices é da forma $(V \cup \{v\}, A \cup \{\{v, v'\}\}, r)$, onde $v \in V$, $v' \notin V$ e (V, A, r) é uma árvore de n vértices. Pela hipótese de indução, $|A| = n - 1$, ou ainda, $|A| + 1 = (n + 1) - 1$, o que mostra que a proposição vale para árvores de $n + 1$ vértices, já que estas têm $|A| + 1$ arestas.

3. Isto segue do fato de que a cada aresta introduzida no grafo, a soma dos graus dos vértices aumenta de 2 unidades.

4. O número de arestas de K_n é $C(n, 2) = n(n + 1)/2$.

5. a) Uma árvore binária de altura 0 tem apenas 1 vértice e ele é folha; e $2^0 = 1$. Seja $n \geq 0$. Suponha, como hipótese de indução, que árvores binárias de altura n possuem, no máximo, 2^n folhas. Uma árvore binária B de altura $n + 1$ possue um máximo de folhas quando as duas subárvores da raiz possuem um máximo de folhas. Isto acontece quando ambas têm altura n. Assim, pela hipótese de indução, elas possuem 2^n folhas. Logo a árvore B tem $2^n + 2^n = 2^{n+1}$ folhas.

   b) Uma árvore binária de altura 0 tem apenas 1 vértice; e $2^{0+1} - 1 = 1$. Seja $n \geq 0$. Suponha, como hipótese de indução, que árvores binárias de altura n possuem, no máximo, $2^{n+1} - 1$ vértices. Uma árvore binária B de altura $n+1$ possue um máximo de vértices quando as duas subárvores da raiz possuem um máximo de vértices. Isto acontece quando ambas têm altura n. Assim, pela hipótese de indução, elas possuem $2^{n+1} - 1$ vértices. Logo a árvore B tem os $2^{n+1} - 1$ vértices de cada uma mais a raiz: $2^{n+1} - 1 + 2^{n+1} - 1 + 1 = 2 \times 2^{n+1} + 1 = 2^{n+2} + 1$.

6. Será usada a definição de árvore do livro.

   (a)→(b) Por indução sobre o número de vértices. Se a árvore tem 1 vértice não tem arestas. Logo, satisfaz (b). Seja $n \geq 1$, e suponha que (b) vale para árvores de n vértices. Seja uma árvore A de $n + 1$ vértices. Pela definição, tal árvore é formada de uma árvore de n vértices pela inclusão de um vértice *novo* e uma nova aresta. Como, pela hipótese de indução, (b) vale para árvore de n vértices, então continua valendo para A.

   (b)→(c) Sendo o grafo acíclico, se existir um caminho simples de um vértice a outro, ele é único. Assim, basta mostrar a existência de caminho simples. Isto será feito por indução sobre o número de vértices. Se o grafo tem 1 vértice, existe caminho simples dele até ele mesmo. Seja $n \geq 1$, e suponha que (c) vale para grafos (que satisfazem (b)) de n vértices ou menos. Seja Seja um grafo G de $n + 1$ vértices que satisfaça (b). Seja v um vértice arbitrário de G. Suponha que existam k vértices adjacentes a v, v_1, v_2,…,v_k. Retirando-se as arestas $\{v, v_1\}$, …, $\{v, v_k\}$, como o grafo é acíclico, obtém-se $k+1$ grafos (componentes) sem vértices em comum, k grafos em que estão os vértices v_1, v_2,…,v_k, e um contendo apenas v. Pela hipótese de indução, (c) vale para cada um deles. Com a reinclusão de v e arestas incidentes, haverá também um caminho simples de cada vértice de cada um dos $k + 1$ componentes a qualquer outro de outro componente, passando por v.

(c)→(a) A prova será feita por indução sobre o número de vértices. Se o grafo tem 1 vértice, ele não tem aresta, pois existe um único caminho simples do vértice a ele mesmo. Seja $n \geq 1$, e suponha que (a) vale para grafos (que satisfazem (c)) de n vértices ou menos. Seja Seja um grafo G de $n+1$ vértices que satisfaça (c). Seja v um vértice arbitrário de G de grau 1; v existe, pois o conjunto de vértices é finito e , assim, partindo-se de qualquer vértice, um caminhamento qualquer deve terminar antes repetir um vértice; caso contrário, existiria mais de um caminho simples, contrariando (c). Removendo-se v e a aresta incidente, obtém-se um grafo que, pela hipótese de indução, é uma árvore. Adicionando-se v e a aresta incidente, obtém-se então uma árvore.

1.10 Linguagens Formais

1. O número de prefixos e o de sufixos é $n+1$. O número máximo de subpalavras é $1 + C(n-1+2,2) = 1 + C(n+1,2) = 1 + (n+1)n/2$; assim, o número de subpalavras vai de $n+1$ a $1+(n+1)n/2$.

2. a) $\{\mathtt{1}\}^*\{\mathtt{0}\}\{\mathtt{0},\mathtt{1}\}^*$.
   b) $\{\mathtt{0},\mathtt{1}\}(\{\mathtt{0},\mathtt{1}\}^2)^*$.
   c) $\{\mathtt{0}\}^*\{\mathtt{0}\}\{\mathtt{1}\}^*$.
   d) $\{\lambda\} \cup \{\mathtt{0}\}\{\mathtt{10}\}^* \cup \{\mathtt{1}\}\{\mathtt{01}\}^*$.
   e) $\{xx \mid x \in \{\mathtt{0},\mathtt{1}\}^*\}$.

3. a) $\{\mathtt{0},\mathtt{1}\}^{10}$.
   b) $\{\mathtt{0},\mathtt{1}\}\{\lambda,\mathtt{0},\mathtt{1}\}^{199}$.
   c) $\{\mathtt{01},\mathtt{1}\}\{\mathtt{0},\mathtt{1}\}^*\{\mathtt{00}\}$.
   d) $\{\mathtt{1},\mathtt{011}\}^*$.
   e) $\{\mathtt{1}\}^*(\{\mathtt{0}\}\{\mathtt{1}\}^*\{\mathtt{0}\}\{\mathtt{1}\}^*)^* \cup \{\mathtt{0}\}^*\{\mathtt{1}\}\{\mathtt{0}\}^*(\{\mathtt{1}\}\{\mathtt{0}\}^*\{\mathtt{1}\}\{\mathtt{0}\}^*)^*$.
   f) $\{\mathtt{0}\}^*\{\mathtt{1}\}\{\mathtt{0}\}^*\{\lambda,\mathtt{1}\}\{\mathtt{0}\}^* \cap (\{\mathtt{0},\mathtt{1}\}^3)^*$

4. a) $A(B \cup C) \subseteq (AB) \cup (AC)$
   Seja $w \in \Sigma^*$ tal que $w \in A(B \cup C)$. Então existem x e y tais que $w = xy$, $x \in A$ e $y \in B \cup C$. No caso em que $y \in B$, segue-se que $xy \in AB$ e no caso em que $y \in C$, segue-se que $xy \in AC$. Logo, $xy = w \in (AB) \cup (AC)$.
   $(AB) \cup (AC) \subseteq A(B \cup C)$
   Seja $w \in \Sigma^*$ tal que $w \in (AB) \cup (AC)$. Então $w \in AB$ ou $w \in AC$. No primeiro caso, existem x e y tais que $w = xy$, $x \in A$ e $y \in B$. Como $y \in B \cup D$ para qualquer linguagem D, segue-se que $xy = w \in A(B \cup C)$. No caso em que $w \in AC$, existem x e y tais que $w = xy$, $x \in A$ e $y \in C$. Como $y \in D \cup C$ para qualquer linguagem D, segue-se que $xy = w \in A(B \cup C)$. Conclui-se, portanto, que $w \in A(B \cup C)$.
   b) Contra-exemplo: $A = \{\lambda, a\}$, $B = \{b\}$, $C = \{ab\}$.

5. Suponha que $\lambda \in L$. Neste caso, $\lambda \in L^+$ e, portanto, $L^+ \cup \{\lambda\} = L^+$. Como $L^* = L^+ \cup \{\lambda\}$, segue-se que $L^* = L^+$. Agora suponha que $\lambda \notin L$. Então $\lambda \notin L^+$. Como $L^* = L^+ \cup \{\lambda\}$, segue-se que $L^+ = L^* - \{\lambda\}$.

6. L^* é finita se, e somente se, $L = \emptyset$ ou $L = \{\lambda\}$.

7. Uma condição necessária e suficiente para $L = L^R$ é: $w \in L$ se, e somente se, $w^R \in L$, para toda palavra w.

   Prova:

   ($\rightarrow$) Suponha que $L = L^R$. Seja w uma palavra arbitrária. Se $w \in L$, então, pela definição de reverso, $w^R \in L^R$; e como $L^R = L$, $w^R \in L$. Por outro lado, se $w^R \in L$, então, pela definição de reverso, $(w^R)^R = w \in L^R$; e como $L^R = L$, $w \in L$. Assim, para toda palavra w, $w \in L$ se, e somente se, $w^R \in L$.

   ($\leftarrow$) Suponha que $w \in L$ se, e somente se, $w^R \in L$, para toda palavra w. $L \subseteq L^R$, pois: se $x \in L$, então, pela suposição acima, $x^R \in L$; e pela definição de reverso, $(x^R)^R = x \in L^R$. Por outro lado, $L^R \subseteq L$, pois: se $x \in L^R$, da definição de reverso, tem-se que $x^R \in L$ (pois $x = (x^R)^R$); e pela suposição acima, $x \in L$. Conclui-se então que $L = L^R$.

8. A prova será por indução sobre n. Para $n = 0$, tem-se: $(w^R)^0 = \lambda = \lambda^R = (w^0)^R$. Seja $n \geq 0$ e suponha que $(w^R)^n = (w^n)^R$. Segue-se que: $(w^R)^{n+1} = (w^R)^n w^R = (w^n)^R w^R$, pela hipótese de indução. E como $(w^n)^R w^R = (ww^n)^R = (w^{n+1})^R$, segue-se que $(w^R)^{n+1} = (w^{n+1})^R$.

9. $L^* \subseteq \bigcup_{n \in \mathbf{N}} L^n$.

   Por indução forte sobre $|w|$. Suponha, como hipótese de indução, que se $w \in L^*$ então $w \in \bigcup_{n \in \mathbf{N}} L^n$ para palavras de tamanho menor que um certo k arbitrário. Para mostrar que isto vale também para palavras de tamanho k, considera-se dois casos:

   $k = 0$. A única palavra de tamanho 0 é λ, e $\lambda \in \bigcup_{n \in \mathbf{N}} L^n$, pois $L^0 = \{\lambda\}$.

   $k > 0$. Seja w de tamanho k tal que $w \in L^*$. Pela definição de fecho de Kleene, pode-se dizer que $w = xy$, onde $x \in L^*$, $y \in L$ e $y \neq \lambda$. Pela hipótese de indução, $x \in \bigcup_{n \in \mathbf{N}} L^n$. Segue-se que $x \in L^i$ para algum $i \in \mathbf{N}$. Como $y \in L$, tem-se que $xy \in L^{i+1}$. Portanto, $w \in \bigcup_{n \in \mathbf{N}} L^n$.

   $\bigcup_{n \in \mathbf{N}} L^n \subseteq L^*$.

   Este resultado segue do fato de que para todo $n \geq 0$, $L^n \subseteq L^*$, fato este que será provado por indução sobre n. Inicialmente, note que $L^0 = \{\lambda\}$, e $\lambda \in L^*$ por definição. Seja n arbitrário e suponha que $L^n \subseteq L^*$. Como $L^{n+1} = L^n L$, e, pela hipótese de indução, $L^n \subseteq L^*$, segue-se, pela definição de fecho de Kleene, que $L^{n+1} \subseteq L^*$.

10. a) Seja w uma palavra arbitrária. Se $w \in L^*$, como $\lambda \in L^*$, segue-se que $w\lambda = w \in L^*L^*$. Para completar, será provado, por indução no tamanho de y que para quaisquer palavras $x, y \in L^*$, se $xy \in L^*L^*$ então $xy \in L^*$. Inicialmente, veja que se $x \in L^*$, $x\lambda = x \in L^*$. Seja $n \geq 0$ e suponha, como hipótese de indução, que se $xy \in L^*L^*$ então $xy \in L^*$ para palavras y de n ou menos símbolos. Seja y uma palavra arbitrária de L^* de $n+1$ símbolos e $x \in L^*$. Então, $y = y_1y_2$, onde $y_1 \in L^*$ e $y_2 \in L$, $y_2 \neq \lambda$. Assim, dado que $xy_1 \in L^*L^*$, pela hipótes de indução $xy_1 \in L^*$. Como $y_2 \in L$, segue-se que $(xy_1)y_2 = xy \in L^*$.

    b) Seja w uma palavra arbitrária. Se $w \in L^*$, como $\lambda \in (L^*)^*$, segue-se da definição de fecho de Kleene que $\lambda w \in (L^*)^*L^*$, ou seja, $w \in (L^*)^*$. Será provado, por indução no tamanho de w que para quaisquer palavras $w \in (L^*)^*$, $w \in L^*$. Inicialmente, veja que $\lambda \in L^*$. Seja $n \geq 0$ e suponha, como hipótese de indução, que se $x \in (L^*)^*$ então $x \in L^*$ para palavras x de n ou menos símbolos. Seja w uma palavra de $n+1$ símbolos. Suponha que $w \in (L^*)^*$. Então $w = xy$, onde $x \in (L^*)^*$ e $y \in L^*$, $y \neq \lambda$. Como $|x| \leq n$, segue-se pela hipótese de indução que $x \in L^*$. Assim, $xy \in L^*L^*$ e, do resultado do item (a), segue-se que $xy \in L^*$.

c) Seja w uma palavra arbitrária. Se $w \in (L_1 \cup L_2)^*$, como $\lambda \in L_1^*$, segue-se que $w\lambda = w \in (L_1 \cup L_2)^* L_1^*$. Será mostrado por indução no tamanho de y que para quaisquer palavras $x \in (L_1 \cup L_2)^*$ e $y \in L_1^*$, se $xy \in (L_1 \cup L_2)^* L_1^*$ então $xy \in (L_1 \cup L_2)^*$. Inicialmente, observe que $\lambda \in (L_1 \cup L_2)^*$. Seja $n \geq 0$ e suponha, como hipótese de indução, que se $xy \in (L_1 \cup L_2)^* L_1^*$ então $xy \in (L_1 \cup L_2)^*$ para palavras $y \in L_1^*$ de n ou menos símbolos. Seja $y \in L_1^*$ uma palavra de $n+1$ símbolos; assim, $y = y_1 y_2$, onde $y_1 \in L_1^*$ e $y_2 \in L_1$, $y_2 \neq \lambda$, e suponha que $xy \in (L_1 \cup L_2)^* L_1^*$. Então $xy_1 \in (L_1 \cup L_2)^* L_1^*$. Como $|y_1| \leq n$, segue-se pela hipótese de indução que $xy_1 \in (L_1 \cup L_2)^*$. Como $y_2 \in L_1$, $y_2 \in L_1 \cup L_2$. Assim, $(xy_1)y_2 = w \in (L_1 \cup L_2)^*$.

d) $(L_1 \cup L_2)^* \subseteq (L_1^* L_2^*)^*$.

Será provado, por indução no tamanho de w que para quaisquer palavras $w \in (L_1 \cup L_2)^*$, $w \in (L_1^* L_2^*)^*$. Tem-se: $\lambda \in (L_1^* L_2^*)^*$. Seja $n \geq 0$ e suponha, como hipótese de indução, que $z \in (L_1^* L_2^*)^*$ para palavras $z \in (L_1 \cup L_2)^*$ de n ou menos símbolos. Seja $w \in (L_1 \cup L_2)^*$ de $n+1$ símbolos. Então, $w = xy$, $y \neq \lambda$, $x \in (L_1 \cup L_2)^*$ e $y \in (L_1 \cup L_2)$. Pela hipótese de indução, $x \in (L_1^* L_2^*)^*$. Se $y \in L_1$, então $y \in L_1^*$ e, como $\lambda \in L_2^*$, $y\lambda = y \in L_1^* L_2^*$; logo, $xy = w \in (L_1^* L_2^*)^*$. Por outro lado, se $y \in L_2$, raciocínio análogo mostra que $xy = w \in (L_1^* L_2^*)^*$.

$(L_1^* L_2^*)^* \subseteq (L_1 \cup L_2)^*$.

Será provado, por indução no tamanho de w que para quaisquer palavras $w \in (L_1^* L_2^*)^*$, $w \in (L_1 \cup L_2)^*$. Tem-se: $\lambda \in (L_1 \cup L_2)^*$. Seja $n \geq 0$ e suponha, como hipótese de indução, que $z \in (L_1 \cup L_2)^*$ para palavras $z \in (L_1^* L_2^*)^*$ de n ou menos símbolos. Seja $w \in (L_1^* L_2^*)^*$ de $n+1$ símbolos. Então, $w = xy$, $y \neq \lambda$, $x \in (L_1^* L_2^*)^*$ e $y \in L_1^* L_2^*$. Pela hipótese de indução, $x \in (L_1 \cup L_2)^*$. Sejam $y_1 \in L_1^*$ e $y_2 \in L_2^*$ tais que $y = y_1 y_2$. Então $w = xy_1 y_2 \in (L_1 \cup L_2)^* L_1^* L_2^*$. Aplicando-se o resultado do item (c) duas vezes, conclui-se que $w \in (L_1 \cup L_2)^*$.

11. $L_1 = \{\lambda, \mathtt{a}\}$, $L_2 = \{\mathtt{a}, \mathtt{aa}\}$.

12. a) Definição recursiva de $X = \{\mathtt{0}\}^* \{\mathtt{1}\}^*$:
    - $\lambda \in X$;
    - se $x \in X$ então $\mathtt{0}x \in X$ e $x\mathtt{1} \in X$.

    b) Definição recursiva de $Y = \{\mathtt{0}^n \mathtt{1}^n \mid n \in \mathbf{N}\}$:
    - $\lambda \in Y$;
    - se $x \in Y$ então $\mathtt{0}x\mathtt{1} \in Y$.

    c) Definição recursiva de $Z = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w \text{ contém } \mathtt{00}\}$:
    - $\mathtt{00} \in Z$;
    - se $x \in Z$ então $\mathtt{0}x, \mathtt{1}x, x\mathtt{0}, x\mathtt{1} \in Z$.

    d) Definição recursiva de $W = \{\mathtt{0}^0\mathtt{1}\mathtt{0}^1\mathtt{1}\mathtt{0}^2\mathtt{1}\ldots\mathtt{0}^n\mathtt{1} \mid n \in \mathbf{N}\}$:
    - $\mathtt{1}, \mathtt{101} \in W$;
    - se $x\mathtt{1}\mathtt{0}^n\mathtt{1} \in W$ então $x\mathtt{1}\mathtt{0}^n\mathtt{1}\mathtt{0}^{n+1}\mathtt{1} \in W$.

1.11 Gramáticas

1. Nas derivações abaixo estão grifadas as variáveis expandidas.

a) $A \Rightarrow \underline{B}B \Rightarrow \underline{B} \Rightarrow \lambda$.
$A \Rightarrow B\underline{B} \Rightarrow \underline{B} \Rightarrow \lambda$.

b) $A \Rightarrow \underline{B}B \Rightarrow 0\underline{B}1B \Rightarrow 01\underline{B} \Rightarrow 01.$
$A \Rightarrow \underline{B}B \Rightarrow 0B1\underline{B} \Rightarrow 0\underline{B}1 \Rightarrow 01.$
$A \Rightarrow B\underline{B} \Rightarrow \underline{B}0B1 \Rightarrow 0\underline{B}1 \Rightarrow 01.$
$A \Rightarrow B\underline{B} \Rightarrow B0\underline{B}1 \Rightarrow \underline{B}01 \Rightarrow 01.$

c) $A \Rightarrow \underline{B}B \Rightarrow 0\underline{B}1B \Rightarrow 01\underline{B} \Rightarrow 010\underline{B}1 \Rightarrow 0101.$
$A \Rightarrow \underline{B}B \Rightarrow 0B1\underline{B} \Rightarrow 0\underline{B}10B1 \Rightarrow 010\underline{B}1 \Rightarrow 0101.$
$A \Rightarrow \underline{B}B \Rightarrow 0B1\underline{B} \Rightarrow 0B10\underline{B}1 \Rightarrow 0\underline{B}101 \Rightarrow 0101.$
$A \Rightarrow B\underline{B} \Rightarrow \underline{B}0B1 \Rightarrow 0\underline{B}10B1 \Rightarrow 010\underline{B}1 \Rightarrow 0101.$
$A \Rightarrow B\underline{B} \Rightarrow \underline{B}0B1 \Rightarrow 0B10\underline{B}1 \Rightarrow 0\underline{B}101 \Rightarrow 0101.$
$A \Rightarrow B\underline{B} \Rightarrow B0\underline{B}1 \Rightarrow B01 \Rightarrow 0\underline{B}101 \Rightarrow 0101.$

d) $A \Rightarrow \underline{B}B \Rightarrow \underline{B} \Rightarrow 0\underline{B}1 \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow \underline{B}B \Rightarrow 0\underline{B}1B \Rightarrow 00\underline{B}11B \Rightarrow 0011\underline{B} \Rightarrow 0011.$
$A \Rightarrow \underline{B}B \Rightarrow 0\underline{B}1B \Rightarrow 00B11\underline{B} \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow \underline{B}B \Rightarrow 0B1\underline{B} \Rightarrow 0\underline{B}1 \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow B\underline{B} \Rightarrow \underline{B} \Rightarrow 0\underline{B}1 \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow B\underline{B} \Rightarrow \underline{B}0B1 \Rightarrow 0\underline{B}1 \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow B\underline{B} \Rightarrow B0\underline{B}1 \Rightarrow \underline{B}00B11 \Rightarrow 00\underline{B}11 \Rightarrow 0011.$
$A \Rightarrow B\underline{B} \Rightarrow B0\underline{B}1 \Rightarrow B00\underline{B}11 \Rightarrow \underline{B}0011 \Rightarrow 0011.$

A linguagem gerada é $\{0^n1^n \mid n \in \mathbf{N}\}^2$.

2. $L(G') = L(G)$. Cada palavra gerada por G, a gramática do Exemplo 49, pode ser gerada por uma derivação da forma:

$$
\begin{array}{ll}
P \Rightarrow \mathtt{a}A\mathtt{bc} & \text{(regra 1)} \\
\overset{k}{\Rightarrow} \mathtt{aa}^k A(\mathtt{b}C)^k\mathtt{bc} & \text{(regra 2, } k \text{ vezes; } k \geq 0) \\
\Rightarrow \mathtt{aa}^k(\mathtt{b}C)^k\mathtt{bc} & \text{(regra 3)} \\
\Rightarrow \mathtt{a}^{k+1}(\mathtt{b}C)^{k-1}\mathtt{b}^2\mathtt{Cc} & \text{(regra 4, 1 vez)} \\
\overset{2}{\Rightarrow} \mathtt{a}^{k+1}(\mathtt{b}C)^{k-2}\mathtt{b}^3\mathtt{C}^2\mathtt{c} & \text{(regra 4, 2 vezes)} \\
\vdots & \\
\overset{k}{\Rightarrow} \mathtt{a}^{k+1}\mathtt{b}^{k+1}\mathtt{C}^k\mathtt{c} & \text{(regra 4, } k \text{ vezes)} \\
\overset{k}{\Rightarrow} \mathtt{a}^{k+1}\mathtt{b}^{k+1}c^{k+1} & \text{(regra 5, } k \text{ vezes)}
\end{array}
$$

Cada uma destas palavras pode ser gerada, por meio de G', como mostrado abaixo. O que está diferente está grifado.

$$
\begin{array}{ll}
P \Rightarrow \mathtt{a}A\mathtt{b}\underline{D} & \text{(regra 1)} \\
\overset{k}{\Rightarrow} \mathtt{aa}^k A(\mathtt{b}C)^k\mathtt{b}\underline{D} & \text{(regra 2, } k \text{ vezes; } k \geq 0) \\
\Rightarrow \mathtt{aa}^k(\mathtt{b}C)^k\mathtt{b}\underline{D} & \text{(regra 3)} \\
\Rightarrow \mathtt{a}^{k+1}(\mathtt{b}C)^{k-1}\mathtt{b}^2\mathtt{C}\underline{D} & \text{(regra 4, 1 vez)} \\
\overset{2}{\Rightarrow} \mathtt{a}^{k+1}(\mathtt{b}C)^{k-2}\mathtt{b}^3\mathtt{C}^2\underline{D} & \text{(regra 4, 2 vezes)} \\
\vdots & \\
\overset{k}{\Rightarrow} \mathtt{a}^{k+1}\mathtt{b}^{k+1}\mathtt{C}^k\underline{D} & \text{(regra 4, } k \text{ vezes)} \\
\overset{k}{\Rightarrow} \mathtt{a}^{k+1}\mathtt{b}^{k+1}\underline{D}c^k & \text{(regra 5, } k \text{ vezes)} \\
\overset{k}{\Rightarrow} \mathtt{a}^{k+1}\mathtt{b}^{k+1}c^{k+1} & \text{(regra 6, 1 vez)}
\end{array}
$$

Note-se que uma derivação de uma palavra da linguagem gasta um passo a mais na gramática G': o último, que é usado para produzir o último c. Uma derivação em G' que, usando a regra 6, substitua D por c antes da subsitituição de todos os Cs por cs, não leva a nenhuma palavra da linguagem.

3. a) $P \to \mathtt{a}I \mid \mathtt{b}P \mid \lambda$
   $I \to \mathtt{a}P \mid \mathtt{b}I$

   b) $X \to \mathtt{a}X\mathtt{b} \mid \lambda$

   c) $X \to \mathtt{a}X\mathtt{a} \mid \mathtt{b}X\mathtt{b} \mid \lambda \mid \mathtt{a} \mid \mathtt{b}$

   d) $P \to A \mid B \mid \lambda$
   $A \to \mathtt{a}B\mathtt{a} \mid \mathtt{a}$
   $B \to \mathtt{b}A\mathtt{b} \mid \mathtt{b}$

   e) $P \to \mathtt{a}BPC\mathtt{d} \mid \lambda$
   $BC \to \mathtt{bc}$
   $B\mathtt{a} \to \mathtt{a}B$
   $B\mathtt{b} \to \mathtt{bb}$
   $\mathtt{d}C \to C\mathtt{d}$
   $\mathtt{c}C \to \mathtt{cc}$

4. Para a gramática do item (b), observando-se o esquema:

   $$X \overset{k}{\Rightarrow} \mathtt{a}^k X \mathtt{b}^k \Rightarrow \mathtt{a}^k\mathtt{b}^k.$$

   vê-se que são gastos $k + 1 = (n/2) + 1$ passos.

   Para a gramática do item (c), se $n = 0$, é gasto 1 passo:

   $$X \Rightarrow \lambda$$

   e se $n > 0$ são gastos $k = \lceil n/2 \rceil + 1$ passos, pois

   - Uma derivação começa assim: $X \overset{k}{\Rightarrow} xXx^R$ (regras 1 e 2, k vezes, $k \geq 0$, $|x| = |x^R| = k$).
   - Em seguida, é aplicada uma das 3 últimas regras. Se for a regra λ, o número de passos é $k = (n/2) + 1$, onde $n = 2k$. Se for uma das outras duas regras, o número de passos é $k = [(n-1)/2] + 1$, onde $n = 2k + 1$.

5. $L(G) = \{\mathtt{a}\}^*\{\mathtt{b}\}^*$.

   Prova:

   $\{\mathtt{a}\}^*\{\mathtt{b}\}^* \subseteq L(G)$. O seguinte esquema de derivação mostra que toda palavra da forma $\mathtt{a}^i\mathtt{b}^j$, para $i, j \geq 0$, é gerada por G:

   $$\begin{array}{lll} A & \overset{i}{\Rightarrow} \mathtt{a}^i A & (\text{regra } A \to \mathtt{a}A,\ i \text{ vezes, } i \geq 0) \\ & \Rightarrow \mathtt{a}^i B & (\text{regra } A \to B) \\ & \overset{j}{\Rightarrow} \mathtt{a}^i\mathtt{b}^j B & (\text{regra } B \to \mathtt{b}B,\ j \text{ vezes, } j \geq 0) \\ & \Rightarrow \mathtt{a}^i\mathtt{b}^j & (\text{regra } B \to \lambda) \end{array}$$

   Como *qualquer palavra* gerada por G segue necessariamente este mesmo esquema de derivação, segue-se que $L(G) \subseteq \{\mathtt{a}\}^*\{\mathtt{b}\}^*$.

1.12 Problemas de Decisão

1. a) Nada se pode dizer. X pode ser decidível ou não.

   b) X é decidível: para qualquer entrada x (para X), o algoritmo $\mathcal{R}$ produz uma saída y, a partir da qual um algoritmo para D obtém a resposta para x.

   c) X é indecidível: pelo item anteriror, se X fosse decidível, I seria decidível.

   d) Nada se pode dizer. X pode ser decidível ou não.

1.13 Exercícios

1. Segue uma definição recursiva de $\hat{v} : \text{LP} \to \{V, F\}$:

   a) $\hat{v}(\alpha) = v(\alpha)$ para $\alpha \in \text{VP}$;

   b) $\hat{v}(\neg\alpha) = V$ se, e somente se, $\hat{v}(\alpha) = F$;
   $\hat{v}((\alpha \wedge \beta)) = V$ se, e somente se, $\hat{v}(\alpha) = V$ e $\hat{v}(\beta) = V$;
   $\hat{v}((\alpha \vee \beta)) = F$ se, e somente se, $\hat{v}(\alpha) = F$ e $\hat{v}(\beta) = F$;
   $\hat{v}((\alpha \to \beta)) = F$ se, e somente se, $\hat{v}(\alpha) = V$ e $\hat{v}(\beta) = F$;
   $\hat{v}((\alpha \leftrightarrow \beta)) = V$ se, e somente se, $\hat{v}(\alpha) = \hat{v}(\beta)$.

2. Basta substituir, indutivamente:

   - $\alpha \wedge \beta$ por $\neg(\neg\alpha \vee \neg\beta)$;
   - $\alpha \to \beta$ por $\neg\alpha \vee \beta$;
   - $\alpha \leftrightarrow \beta$ por $(\neg\alpha \vee \beta) \wedge (\alpha \vee \neg\beta)$;
   - $\exists x\alpha$ por $\neg\forall x\neg\alpha$.

3. Por indução sobre n. Inicialmente, veja que $(0+1)^2 - 0^2 = 1 > 0$. Seja n um número natural arbitrário. Suponha, como hipótese de indução, que existe $k \in \mathbf{N}$ tal que $(k+1)^2 - k^2 > n$. Seja k_0 tal número. Como $(k_0+1)^2 - k_0^2 = k_0^2 + 2k_0 + 1 - k_0^2 = 2k_0 + 1$, pela hipótese de indução, tem-se que $2k_0 + 1 > n$. Logo, $2k_0 + 2 > n + 1$. E como $(k_0+2)^2 - (k_0+1)^2 = k_0^2 + 4k_0 + 4 - k_0^2 - 2k_0 - 1 = 3k_0 + 3$, segue-se que $(k_0+2)^2 - (k_0+1)^2 > n+1$. Portanto, existe um número natural k tal que $(k+1)^2 - k^2 > n+1$: $k_0 + 1$ seria um tal número.

4. Deve-se provar que existem $k, n_0 \in \mathbf{N}$ tais que para todo $n \geq n_0$, $10n^2 + 100n + 1000 \leq kn^2$. Sejam $k = 1110$ e $n_0 = 1$. Será provado, por indução, que para todo $n \geq n_0$, $10n^2 + 100n + 1000 \leq kn^2 = 1110n^2$. Inicialmente, veja que $10 \times 1^2 + 100 \times 1 + 1000 = 1110 = 1110 \times 1^2$. Seja n um número natural arbitrário maior ou igual a 1, e suponha, como hipótese de indução, que $10n^2 + 100n + 1000 \leq 1110n^2$. Segue-se que $10(n+1)^2 + 100(n+1) + 1000 = 10n^2 + 20n + 10 + 100n + 100 + 1000 = (10n^2 + 100n + 1000) + 20n + 110$. Pela hipótese de indução, $(10n^2 + 100n + 1000) + 20n + 110 \leq 1110n^2 + 20n + 110$. Como esta última é menor do que $(1110n^2 + 20n + 110) + 2200n + 1000$ e esta, por sua vez é igual a $1110n^2 + 2220n + 1110 = 1110(n+1)^2$, conclui-se que $10(n+1)^2 + 100(n+1) + 1000 < 1110(n+1)^2$. Portanto, pelo princípio de indução, para todo $n \geq 1$, $10n^2 + 100n + 1000 \leq 1110n^2$. Já que existem $k, n_0 \in \mathbf{N}$ tais que para todo $n \geq n_0$, $10n^2 + 100n + 1000 \leq kn^2$, conclui-se que $10n^2 + 100n + 1000$ é $O(n^2)$.

   Técnicas de provas utilizadas: construtiva para a existencial 2 vezes (quando se exibiu $k = 1110$ e $n_0 = 1$) e indução (para provar que para todo $n \geq n_0 \ldots$).

5. $|A \cup B| = |A| + |B| - |A \cap B|$. *Prova*: Cada elemento de $A - B$ e cada elemento de $A \cap B$ é contado uma vez em $|A|$, e cada elemento de $B - A$ e cada elemento de $A \cap B$ é contado uma vez em $|B|$. Assim, os elementos de $A \cap B$ são contados duas vezes em $|A| + |B|$, razão da subtração de $|A \cap B|$.

   Generalizando: $| \cup_{i=1}^{n} A_i| = S_1 + S_2 + \cdots + S_n$, onde:

$$S_k = (-1)^{k+1} \times \sum_{j_1 \neq j_2 \neq \cdots \neq j_k} |A_{j_1} \cap A_{j_2} \cap \ldots \cap A_{j_k}|, \text{ para } 1 \leq k \leq n.$$

   Para provar este resultado por indução, note inicialmente que $| \cup_{i=1}^{1} A_i| = |A_1| = S_1$. Seja $n \geq 1$ arbitrário. Hipótese de indução: $| \cup_{i=1}^{n} A_i| = S_1 + S_2 + \cdots + S_n$, sendo S_k definido como acima. Basta, agora, provar que $| \cup_{i=1}^{n+1} A_i| = S'_1 + S'_2 + \cdots + S'_{n+1}$, onde cada S'_k é como S_k, só que cada j_k pode ser também $k+1$. Como $\cup_{i=1}^{n+1} A_i = (\cup_{i=1}^{n} A_i) \cup A_{n+1}$, tem-se, pelo resultado acima, que:

$$| \cup_{i=1}^{n+1} A_i| = | \cup_{i=1}^{n} A_i| + |A_{n+1}| - |(\cup_{i=1}^{n} A_i) \cap A_{n+1}|.$$

   Pela hipótese de indução, segue-se que:

$$| \cup_{i=1}^{n+1} A_i| = S_1 + S_2 + \cdots + S_n + |A_{n+1}| - |(\cup_{i=1}^{n} A_i) \cap A_{n+1}|.$$

   Como $S'_1 = S_1 + |A_{n+1}|$ e a interseção distribui sobre união:

$$| \cup_{i=1}^{n+1} A_i| = S'_1 + S_2 + \cdots + S_n - | \cup_{i=1}^{n} (A_i \cap A_{n+1})|.$$

   Pela hipótese de indução, tem-se que:

$$| \cup_{i=1}^{n+1} (A_i \cap A_{n+1})| = S''_1 + S''_2 + \cdots + S''_n, \text{ onde:}$$

$$S''_k = (-1)^{k+1} \times \sum_{j_1 \neq j_2 \neq \cdots \neq j_k} |(A_{j_1} \cap A_{n+1}) \cap (A_{j_2} \cap A_{n+1}) \cap \ldots \cap (A_{j_k} \cap A_{n+1})|$$

   para $1 \leq k \leq n$. Ou ainda:

$$S''_k = (-1)^{k+1} \times \sum_{j_1 \neq j_2 \neq \cdots \neq j_k} |A_{j_1} \cap A_{j_2} \cap \ldots \cap A_{j_k} \cap A_{n+1}|.$$

   Assim, como

$$| \cup_{i=1}^{n+1} A_i| = S'_1 + S_2 + \cdots + S_n - (S''_1 + S''_2 + \cdots + S''_n).$$

   e como $S_2 - S''_1 = S'_2$, $S_3 - S''_2 = S'_3$, $\ldots$, $S_n - S''_{n-1} = S'_n$, e $-S''_n = S'_{n+1}$, conclui-se que $| \cup_{i=1}^{n+1} A_i| = S'_1 + S'_2 + \cdots + S'_n + S'_{n+1}$.

6. a) Seja X um elemento arbitrário de $\mathcal{P}(A) \cup \mathcal{P}(B)$. Então $X \in \mathcal{P}(A)$ ou $X \in \mathcal{P}(B)$. No primeiro caso, $X \subseteq A$, e no segundo, $X \subseteq B$. Em qualquer caso, $X \subseteq A \cup B$. Portanto, $X \in \mathcal{P}(A \cup B)$. Conclui-se que $\mathcal{P}(A) \cup \mathcal{P}(B) \subseteq \mathcal{P}(A \cup B)$.

   b) $\mathcal{P}(A) \cap \mathcal{P}(B) \subseteq \mathcal{P}(A \cap B)$.
   Seja X um elemento arbitrário de $\mathcal{P}(A) \cap \mathcal{P}(B)$. Então $X \in \mathcal{P}(A)$ e $X \in \mathcal{P}(B)$; logo, $X \subseteq A$ e $X \subseteq B$. Segue-se que $X \subseteq A \cap B$. Portanto, $X \in \mathcal{P}(A \cap B)$. Conclui-se que $\mathcal{P}(A) \cap \mathcal{P}(B) \subseteq \mathcal{P}(A \cap B)$.
   $\mathcal{P}(A \cap B) \subseteq \mathcal{P}(A) \cap \mathcal{P}(B)$.
   Seja X um elemento arbitrário de $\mathcal{P}(A \cap B)$. Então $X \subseteq A \cap B$; logo, $X \subseteq A$ e $X \subseteq B$. Segue-se que $X \in \mathcal{P}(A)$ e $X \in \mathcal{P}(B)$. Portanto, $X \in \mathcal{P}(A) \cap \mathcal{P}(B)$. Conclui-se que $\mathcal{P}(A \cap B) \subseteq \mathcal{P}(A) \cap \mathcal{P}(B)$.

7. a) $A \triangle B \subseteq (A \cup B) - (A \cap B)$.

Seja $x \in A \triangle B$. Então, por definição, $x \in A - B$ ou $x \in B - A$. No primeiro caso, $x \in A$ e $x \notin B$; como $x \in A$, $x \in A \cup B$, e como $x \notin B$, $x \notin A \cap B$. Assim, $x \in (A \cup B) - (A \cap B)$. No segundo caso, procede-se de forma análoga para mostrar que também $x \in (A \cup B) - (A \cap B)$.

$(A \cup B) - (A \cap B) \subseteq A \triangle B$.

Seja $x \in (A \cup B) - (A \cap B)$. Segue-se que $x \in A \cup B$ e $x \notin A \cap B$. *Caso 1:* $x \in A$. Como $x \notin A \cap B$, $x \notin B$. Assim, $x \in A - B$. *Caso 2:* $x \in B$. Como $x \notin A \cap B$, $x \notin A$. Assim, $x \in B - A$. Portanto, $x \in A - B$ ou $x \in B - A$, ou ainda, $x \in (A - B) \cup (B - A) = A \triangle B$.

b) $A \triangle (B \triangle C) \subseteq (A \triangle B) \triangle C$.

Seja $x \in A \triangle (B \triangle C)$. Então, pela definição, $x \in A - (B \triangle C)$ ou $x \in (B \triangle C) - A$.

Caso 1: $x \in A - (B \triangle C)$.

Então $x \in A$ e $x \notin B \triangle C$. Desta última, segue-se que $x \notin B - C$ e $x \notin C - B$, ou seja, (I) $x \notin B$ ou $x \in C$ e (II) $x \notin C$ ou $x \in B$. Por um lado, se $x \in C$, de (II) segue-se que $x \in B$; com isto, $x \notin A - B$ e $x \notin B - A$; logo, $x \notin A \triangle B$ e, assim, $x \in C - A \triangle B$. Por outro lado, se $x \notin C$, de (I) segue-se $x \notin B$; com isto, e como $x \in A$, $x \in A - B$ e, portanto, $x \in A \triangle B$; segue-se que $x \in (A \triangle B) - C$. Como $x \in C - A \triangle B$ se $x \in C$, e $x \in (A \triangle B) - C$ se $x \notin C$, conclui-se que $x \in (A \triangle B) \triangle C$.

Caso 2: $x \in (B \triangle C) - A$

Então $x \in B \triangle C$ e $x \notin A$. Da primeira, Segue-se que $x \in B$ e $x \notin C$ ou $x \in C$ e $x \notin B$. No caso em que $x \in B$ e $x \notin C$, $x \in B - A$ e, portanto, $x \in A \triangle B$; segue-se que $x \in (A \triangle B) - C$ e que $x \in (A \triangle B) \triangle C$. E no caso em que $x \in C$ e $x \notin B$, $x \notin A - B$ e $x \notin B - A$; portanto, $x \notin A \triangle B$; e assim, $x \in C - (A \triangle B)$, ou ainda, $x \in (A \triangle B) \triangle C$.

$(A \triangle B) \triangle C \subseteq A \triangle (B \triangle C)$.

Procede-se forma similar à acima.

c) $(A - B) \triangle (B - A) \subseteq A \triangle B$.

Seja $x \in (A - B) \triangle (B - A)$. Então $x \in (A - B) - (B - A)$ ou $x \in (B - A) - (A - B)$. No caso em que $x \in (A - B) - (B - A)$, $x \in A - B$ e, portanto, $x \in A \triangle B$. E no caso em que $x \in (B - A) - (A - B)$, $x \in A - B$ e, portanto, $x \in A \triangle B$.

$A \triangle B \subseteq (A - B) \triangle (B - A)$.

Seja $x \in A \triangle B$. Então $x \in A - B$ ou $x \in B - A$. No primeiro caso, segue-se que $x \in (A - B) - (B - A)$, e no segundo segue-se que $x \in (B - A) - (A - B)$. Assim, em qualquer caso, $x \in (A - B) \triangle (B - A)$.

d) $(\rightarrow)$

Suponha que $A \triangle B = A$ e suponha que $B \neq \emptyset$. Seja, então $b \in B$. Se $b \in A$, então, como $A \triangle B = A$, $b \in A \triangle B$; mas isto é impossível, pois se $b \in A$, $b \notin B - A$ e se $b \in B$, $b \notin A - B$. Assim, $b \notin A$. Neste caso, $b \in B - A$ e, logo, $b \in A \triangle B$. Mas isto também nõ pode ser, já que $A \triangle B = A$. Conclui-se, portanto, que b não pode existir, e, assim, $B = \emptyset$.

$(\leftarrow)$

Suponha que $B = \emptyset$. Prova-se, primeiro, que $A \triangle B \subseteq A$. Seja $x \in A \triangle B$. Com isto, $x \in A - B$ ou $x \in B - A$. Como $B = \emptyset$, este último caso é impossível. Assim, $x \in A - B$. E como $B = \emptyset$, $x \in A$. Portanto, $A \triangle B \subseteq A$. Agora, prova-se que $A \subseteq A \triangle B$. Seja $x \in A$. Como $B = \emptyset$, $x \in A - B$ e, portanto, $x \in A \triangle B$. Logo, $A \subseteq A \triangle B$.

e) $A - B = A \triangle B$ se, e somente se, $B \subseteq A$, como provado abaixo.

($\rightarrow$)

Suponha que $A - B = A \triangle B$. Seja $b \in B$. Com isto, $b \notin A - B$. Supondo que $b \notin A$, segue-se que $b \in B - A$ e, portanto, $b \in A \triangle B$. Isto não pode ser, visto que $A - B = A \triangle B$. Assim, se $b \in B$, então $b \in A$. Portanto, $B \subseteq A$.

($\leftarrow$)

Suponha que $B \subseteq A$. Se $x \in A - B$, então $x \in A \triangle B$; logo $A - B \subseteq A \triangle B$. Assim, basta mostrar que $A \triangle B \subseteq A - B$. Seja $x \in A \triangle B$. Como $B \subseteq A$, tem-se $x \notin B - A$; portanto, $x \in A - B$. Logo, $A \triangle B \subseteq A - B$.

8. Seja uma relação R reflexiva e transitiva arbitrária. Será mostrado por indução que $R^n = R$ para todo $n \geq 1$. $R^1 = R$, por definição. Seja $n \geq 1$. Suponha, como hipótese de indução, que $R^n = R$. Por definição, $R^{n+1} = R^n \circ R$, pois $n + 1 \geq 2$. Pela hipótese de indução, $R^n = R$. Assim, basta mostrar que $R \circ R = R$. Mostra-se primeiro que $R \subseteq R \circ R$. Para isto, seja $(x, y) \in R$. Como R é reflexiva, $(x, x) \in R$. Logo, $(x, y) \in R \circ R$. Portanto, $R \subseteq R \circ R$. Agora, mostra-se que $R \circ R \subseteq R$. Seja $(x, y) \in R \circ R$; então existe z tal que $(x, z) \in R$ e $(z, y) \in R$. Como R é transitiva, $(x, y) \in R$. Portanto, $R \circ R \subseteq R$.

9. Seja $F : A \rightarrow A$ tal que $f(a) = a_C$ para todo $a \in A$, onde a_C é um certo elemento da classe de equivalência a que pertence a. Tal elemento existe, visto que todo $a \in A$ pertence a alguma classe de equivalência e classes de equivalência não são vazias. Qualquer elemento da classe de equivalência de a serve para ser a_C, mas escolhido um, ele deve ser a única imagem para todos os componentes da classe. Uma função assim satisfaz os requisitos, como mostrado abaixo.

   ($\rightarrow$)

   Suponha que xRy. Então x e y pertencem à mesma classe de equivalência C e, portanto, $f(x) = x_C = y_C = f(y)$.

   ($\leftarrow$)

   Suponha que $f(x) = f(y)$. Então $f(x)$ e $f(y)$ pertencem à mesma classe de equivalência. Portanto, xRy.

10. $[a] \neq \emptyset$, visto que R é reflexiva e, portanto, aRa.

    Agora, mostra-se que as classes de equivalência são disjuntas, ou seja, dados $a, b \in A$, $[a] = [b]$ ou $[a] \cap [b] = \emptyset$. Suponha que $[a] \cap [b] \neq \emptyset$, e seja $x \in A$ tal que $x \in [a]$ e $x \in [b]$. Seja $y \in [a]$. Então, aRy e, por simetria, yRa. Como aRx, por transitividade, yRx. Como xRb, por transitividade, yRb. Por simetria, bRy. Logo, $y \in [b]$. Como y é um elemento arbitrário de $[a]$, $[a] \subseteq [b]$. De form análoga, mostra-se $[b] \subseteq [a]$.

    Falta mostrar que $\cup_{a \in A}[a] = A$. As classes de equivalência $[a]$ só têm elementos de A. Assim, $\cup_{a \in A}[a] \subseteq A$. Por outro lado, se $x \in A$, então existe uma classe de equivalência $[x]$ por definição, o que mostra que $A \subseteq \cup_{a \in A}[a]$.

11. a) $f(A \cup B) \subseteq f(A) \cup f(B)$.

    Seja $x \in f(A \cup B)$. Então, $x = f(y)$ onde $y \in A \cup B$. Se $y \in A$, segue-se que $f(y) \in f(A)$ e se $y \in B$, $f(y) \in f(B)$. Assim, $x = f(y) \in f(A) \cup f(B)$.

    $f(A) \cup f(B) \subseteq f(A \cup B)$.

    Seja $x \in f(A) \cup f(B)$. Então $x \in f(A)$ ou $x \in f(B)$. Se $x \in f(A)$, então $x = f(y)$, onde $y \in A$; se $y \in A$, $y \in A \cup B$, e portanto $x = f(y) \in f(A \cup B)$. Da mesma forma, mostra-se que se $x \in f(B)$, $x \in f(A \cup B)$.

b) $f(A \cap B) \subseteq f(A) \cap f(B)$.

Seja $x \in f(A \cap B)$. Então, $x = f(y)$ onde $y \in A \cap B$. Assim, $y \in A$ e $y \in B$, de onde se segue que $f(y) \in f(A)$ e $f(y) \in f(B)$. Logo, $x = f(y) \in f(A) \cap f(B)$.

12. a) Tem-se:

$$\begin{aligned} f_A(x) = 1 &\leftrightarrow x \in A && \text{por definição} \\ &\leftrightarrow x \notin \overline{A} \\ &\leftrightarrow f_{\overline{A}}(x) \neq 1 && \text{por definição} \\ &\leftrightarrow f_{\overline{A}}(x) = 0 \\ &\leftrightarrow 1 - f_{\overline{A}}(x) = 1. \end{aligned}$$

Portanto, $f_A(x) = 1 - f_{\overline{A}}(x)$.

b) Tem-se:

$$\begin{aligned} f_{A\cup B}(x) = 1 &\leftrightarrow x \in A \cup B && \text{por definição} \\ &\leftrightarrow x \in A \text{ ou } x \in B \\ &\leftrightarrow f_A(x) = 1 \text{ ou } f_B(x) = 1 && \text{por definição} \\ &\leftrightarrow f_A(x) + f_B(x) - f_A(x)f_B(x) = 1. \end{aligned}$$

Portanto, $f_{A\cup B}(x) = f_A(x) + f_B(x) - f_A(x)f_B(x)$.

c) Tem-se:

$$\begin{aligned} f_{A\cap B}(x) = 1 &\leftrightarrow x \in A \cap B && \text{por definição} \\ &\leftrightarrow x \in A \text{ e } x \in B \\ &\leftrightarrow f_A(x) = 1 \text{ e } f_B(x) = 1 && \text{por definição} \\ &\leftrightarrow f_A(x)f_B(x) = 1. \end{aligned}$$

Portanto, $f_{A\cap B}(x) = f_A(x)f_B(x)$.

d) Tem-se:

$$\begin{aligned} f_{A-B}(x) = 1 &\leftrightarrow x \in A - B && \text{por definição} \\ &\leftrightarrow x \in A \text{ e } x \notin B \\ &\leftrightarrow f_A(x) = 1 \text{ e } f_B(x) \neq 1 && \text{por definição} \\ &\leftrightarrow f_A(x) = 1 \text{ e } f_B(x) = 0 \\ &\leftrightarrow f_A(x) = 1 \text{ e } 1 - f_B(x) = 1 \\ &\leftrightarrow f_A(x)(1 - f_B(x)) = 1. \end{aligned}$$

Portanto, $f_{A\cap B}(x) = f_A(x)f_B(x)$.

13. $(\rightarrow)$

Seja uma função injetora $f : A \rightarrow B$. Seja C a imagem de A. Então $g : C \rightarrow A$ tal que $g(c)$ é o elemento a tal que $f(a) = c$ é uma função sobretora. O elemento a existe e é único visto que f é uma função injetora.

$(\leftarrow)$

Seja uma função sobrejetora $f : B \rightarrow A$. Então $g : A \rightarrow B$ tal que $g(a)$ é algum elemento de $\{b \,|\, f(b) = a\}$ é uma função injetora. Como f é uma função sobrejetora, para todo a tal conjunto não é vazio; e os conjuntos são disjuntos, pois f é função.

14. Seja $\Sigma = \{a_1, a_2, \ldots, a_n\}$. Então:

Σ^* **é enumerável.** Uma enumeração para Σ é dada pela função $\eta : \Sigma^* \rightarrow \mathbf{N}$ tal que $\eta(\lambda) = 0$ e $\eta(a_{p_m}a_{p_{m-1}}\ldots a_{p_0}) = \sum_{k=0}^{m}(p_k \times n^k)$.

$\mathcal{P}(\Sigma^*)$ **não é enumerável.** Suponha que $\mathcal{P}(\Sigma^*)$ é enumerável. Então existe uma função bijetora de $\mathcal{P}(\Sigma^*)$ para $\mathbf{N}$, de forma que os elementos de $\mathcal{P}(\Sigma^*)$ podem ser enumerados: $A_0, A_1, A_2, \ldots$. Seja o conjunto $B = \{w \in \Sigma^* \,|\, w \notin A_{\eta(w)}\}$, onde η é a função de enumeração vista acima (ou qualquer outra). Mas, como $\{\eta(w) \,|\, w \in \Sigma^*\} = \mathbf{N}$, segue-se que $B \neq A_i$ para todo $i \in \mathbf{N}$. Logo, a suposição de que existe a enumeração $A_0, A_1, A_2, \ldots$ não é correta, ou seja, $\mathcal{P}(\Sigma^*)$ não é enumerável.

15. Seja uma função bijetora $f : A \to \mathcal{P}(A)$. Agora, considere o conjunto $D \subseteq A$ assim definido:

$$\text{para cada } a \in A, a \in D \text{ se, e somente se, } a \notin f(a).$$

Como f é uma função bijetora, para cada $X \subseteq A$ deve existir $x \in A$ tal que $f(x) = X$. Em particular, para D deve então existir $d \in A$ tal que $f(d) = D$. Mas, pela definição de D, segue-se que $d \in D = f(d)$ se, e somente se, $d \notin f(d)$. Tal contradição leva à conclusão que não existe uma função bijetora de um conjunto A para um conjunto $\mathcal{P}(A)$.

Este resultado implica que para qualquer conjunto existe conjunto com cardinalidade maior. Em particular, considerando-se conjuntos infinitos, existe uma infinidade de "ordens de infinidade": $\mathbf{N}$, $\mathcal{P}(\mathbf{N})$, $\mathcal{P}(\mathcal{P}(\mathbf{N}))$, etc.

16. A representação de um número n na base b gasta $\lfloor \log_b n \rfloor + 1$ símbolos, se $b > 1$. Como $\log_b n = \log_b c \times \log_c n$, onde $b > 1$ e $c > 1$, vê-se que a transição de uma base para outra, ambas maiores ou iguais a 2, é influenciada apenas por um fator constante, $\log_b c$. E dado o exposto no Exemplo 47, as representações em bases maiores ou iguais a 2 são exponencialmente mais concisas do que na base 1.

17. Definição recursiva de $v : \{\mathtt{0}, \mathtt{1}\}^* \to \mathbf{N}$, de forma que $v(w)$ é o número representado por w na base 2:

   a) $v(\mathtt{0}) = 0$ e $v(\mathtt{1}) = 1$;
   b) $v(x\mathtt{0}) = 2v(x)$ e $v(x\mathtt{1}) = 2v(x) + 1$.

18. Subtração:

   a) $m - 0 = m$, para todo $m \in \mathbf{N}$;
   b) $s(m) - s(n) = m - n$, para todo $m, n \in \mathbf{N}$.

   Divisão:

   a) $m/s(0) = m$, para todo $m \in \mathbf{N}$;
   b) para todo $m \in \mathbf{N}$ e todo $n > s(0)$, $m/n = 0$, se $m < n$, e $m/n = s((m - n)/n)$, se $m \geq n$.

   onde a relação $<$ é assim definida:

   a) $n < s(n)$, para todo $n \in \mathbf{N}$;
   b) se $m < n$, então $m < s(n)$, para todo $m, n \in \mathbf{N}$.

   Resto da divisão:

   a) $n \; mod \; n = 0$, para todo $n \in \mathbf{N}$;
   b) $m \; mod \; n = m$, se $m < n$;
   c) $m \; mod \; n = (m - n) \; mod \; n$, se $n < m$.

Máximo divisor comum:

a) $mdc(n, n) = n$, para todo $n \in \mathbf{N}$;

b) $mdc(m, n) = mdc(m - n, n)$, se $n < m$;

c) $mdc(m, n) = mdc(m, n - m)$, se $m < n$.

19. Conjunto $Anc(v)$ de ancestrais de um vértice v de uma árvore (V, A, r):

a) $v \in Anc(v)$;

b) se $v \in Anc(v)$ e $(v', v) \in A$, então $v' \in Anc(v)$.

20. $\psi(n) = \lfloor \log_2 n \rfloor$. Será feita uma demonstração por indução forte. Assim, suponha, como hipótese de indução, que $\psi(k) = \lfloor \log_2 k \rfloor$ para todo $k < n$.
<u>*Caso* $n = 1$</u>. $\psi(1) = 0 = \log_2 1$.
<u>*Caso* $n > 1$</u>. Por definição, $\psi(n) = \psi(\lfloor n/2 \rfloor) + 1$. Como, $\lfloor n/2 \rfloor < n$, segue-se, pela hipótese de indução, que $\psi(n) = \lfloor \log_2 \lfloor n/2 \rfloor \rfloor + 1$. Se $n \ mod\ 2 = 0$, então $\psi(n) = \lfloor \log_2 n/2 \rfloor + 1 = \lfloor (\log_2 n) - 1 \rfloor + 1 = \lfloor (\log_2 n) \rfloor - 1 + 1 = \lfloor \log_2 n \rfloor$. Por outro lado, se $n \ mod\ 2 = 1$, então $\psi(n) = \lfloor \log_2 (n-1)/2 \rfloor + 1 = \lfloor \log_2 (n-1) - 1 \rfloor + 1 = \lfloor \log_2 (n-1) \rfloor - 1 + 1 = \lfloor \log_2 (n-1) \rfloor$. Mas, se $n \ mod\ 2 = 1$ e $n > 0$, $\lfloor \log_2 (n-1) \rfloor = \lfloor \log_2 n \rfloor$. Assim, $\psi(n) = \lfloor \log_2 n \rfloor$ em qualquer caso.

21. a) $\sum_{k=1}^{0} [k(k+1)(k+2)] = 0 \times 1 \times 2 = 0 = (0 \times 1 \times 2 \times 3)/4$. Seja $n \geq 1$. Suponha, cmo hipótese de indução, que $\sum_{k=1}^{n} [k(k+1)(k+2)] = n(n+1)(n+2)(n+3)/4$. Tem-se: $\sum_{k=1}^{n+1} [k(k+1)(k+2)] = [\sum_{k=1}^{n} [k(k+1)(k+2)]] + (n+1)(n+2)(n+3)$. Aplicando-se a hipótese de indução, obtém-se que esta última é igual a $[n(n+1)(n+2)(n+3)/4] + (n+1)(n+2)(n+3)$. Simplificando-se esta, chega-se a $\sum_{k=1}^{n+1} [k(k+1)(k+2)] = (n+1)(n+2)(n+3)(n+4)/4$, o que completa a prova.

b) Inicialmente, veja que $3^0 - 7^0 - 2 = 0$, que é divisível por 8. Seja n um número natural arbitrário. Suponha como hipótese de indução que $3^n + 7^n - 2$ é divisível por 8. Tem-se: $3^{n+1} + 7^{n+1} - 2 = 3 \times 3^n + 7 \times 7^n - 2 = 3(3^n + 7^n - 2) + (4 \times 7^n + 4)$. Pela hipótese de indução, o primeiro termo desta última é divisível por 8. Assim, basta mostrar que o segundo, $4 \times 7^n + 4$ também é divisível por 8, o que será feito também por indução. Para $n = 0$: $4 \times 7^0 + 4 = 8$, que é divisível por 8. Seja $n \geq 0$, e suponha que $4 \times 7^n + 4$ é divisível por 8. Segue-se que $4 \times 7^{n+1} + 4 = 7(4 \times 7^n + 4) - 24$. O primeiro termo desta última é divisível por 8, pela hipótese de indução, e o segundo, 24, também é, o que leva à conclusão que $4 \times 7^{n+1} + 4$ é divisível por 8.

c) Tem-se: $\prod_{k=2}^{0} (1 - 1/k) = 1 - 1/2 = 1/2$. Seja $n \geq 2$, e suponha, como hipótese de indução, que $\prod_{k=2}^{n} (1 - 1/k) = 1/n$. Segue-se que: $\prod_{k=2}^{n+1} (1 - 1/k) = [\prod_{k=2}^{n} (1 - 1/k)] \times [1 - 1/(n+1)] = (1/n) \times [1 - 1/(n+1)]$, pela hipótese de indução. Esta última é igual a $(1/n) \times (n + 1 - 1)/(n+1)] = 1/(n+1)$, o que completa a prova.

d) $\sum_{k=1}^{2} (1/\sqrt{k}) = 1 + 1/\sqrt{2} = (\sqrt{2} + 1)/\sqrt{2} = \sqrt{2}(\sqrt{2} + 1)/2 > \sqrt{2}$, pois $\sqrt{2} + 1 > 2$. Seja $n \geq 2$. Hipótese de indução: $\sum_{k=1}^{n} (1/\sqrt{k}) > \sqrt{n}$. Segue-se que: $\sum_{k=1}^{n+1} (1/\sqrt{k}) = [\sum_{k=1}^{n} (1/\sqrt{k})] + 1/\sqrt{n+1}$. Segue-se, pela hipótese de indução, que $\sum_{k=1}^{n+1} (1/\sqrt{k}) > \sqrt{n} + 1/\sqrt{n+1}$. Mas esta última é igual a $[\sqrt{n}\sqrt{n+1} + 1]/\sqrt{n+1}$, que, por sua vez é igual a $\sqrt{n+1}[\sqrt{n}\sqrt{n+1} + 1]/(n+1)$. Mas esta é maior do que $\sqrt{n+1}$, como requerido para completar a prova, pois $\sqrt{n}\sqrt{n+1} + 1 > n + 1$, já que $\sqrt{n}\sqrt{n+1} > n$.

22. Representando cada pessoa por um vértice e cada relacionamento de amizade por uma aresta, de tal forma que v, v' é uma aresta se, e somente se, as pessoas representadas por v e v' são amigas, deve-se mostrar que o grafo tem 2 vértices com o mesmo grau. Para

isto, pode-se usar o princípio do pombal, segundo o qual para se acomodar n pombos em menos de n casinhas, alguma casinha deverá receber mais de 1 pombo. Seja um grafo de n vértices, representando um grupo de n pessoas. Observando-se que um vértice pode ter de 0 a $n-1$ vértices adjacentes (o grafo não pode conter *loops* nem arestas múltiplas), considera-se 2 casos. Supondo que um certo vértice tem grau 0, então cada vértice só pode ter graus de 0 a $n-2$ (um vértice não pode ser adjacente a si mesmo nem àquele de grau 0). Por outro lado, se nenhum vértice tem grau 0, cada vértice só pode ter graus de 1 a $n-1$. Nos dois casos, pelo princípio do pombal, existem dois vértices com o mesmo grau.

23. Se cada vértice pode ter qualquer número de filhos, a altura é 0, se a árvore contém apenas 1 vértice, e é 2, se ela contém mais de 1 vértice; neste último caso, todos os vértices, com excessão da raiz, são filhos da raiz. Se cada vértice pode ter no máximo r filhos, a altura mínima é a de uma árvore que em cada nível k tem r^k vértices, com excessão, possivelmente, do último. Assim, a altura mínima, neste caso, é $\lfloor \log_r n \rfloor$, onde n é o número de vértices.

24. a) Uma árvore estritamente n-ária com i vértices internos tem $n \times i + 1$ vértices, já que cada um dos i vértice internos tem n filhos; e o termo 1 refere-se à raiz.

    b) Dado o resultado anterior, tem-se que uma árvore estritamente n-ária de k vértices tem $(k-1)/n$ vértices internos. Já o número de folhas é $k - i = k - (k-1)/n$.

    c) Uma árvore estritamente n-ária com k vértices tem altura mínima igual a $\lfloor \log_n k \rfloor$, e altura máxima igual a $\lceil (k-1)/n \rceil$.

25. $L = X$, onde $X = \{\mathtt{0}^m\mathtt{1}^n \mid m, n \geq 0\}$.

    $L \subseteq X$.

    Será provado por indução sobre o tamanho das palavras w que se $w \in L$ emtão $w \in X$. $\lambda = \lambda\lambda = \mathtt{0}^0\mathtt{1}^0 \in X$. Seja $n \geq 0$, e suponha, como hipótese de indução, que $x \in X$ se $x \in L$, para x tal que $|x| = n$. Para $w \in L$ tal que $|w| = n+1$ tem-se, pela definição de L, que $w = \mathtt{0}y$ ou $w = x\mathtt{1}$, onde x e y são palavras de L. Pela hipótese de indução, segue-se que $x \in X$ e $y \in X$, ou seja, ambos, x e y são da forma $\mathtt{0}^m\mathtt{1}^n$. Então $w = \mathtt{0}\mathtt{0}^m\mathtt{1}^n \in X$ ou $w = \mathtt{0}^m\mathtt{1}^n\mathtt{1} \in X$.

    $X \subseteq L$.

    Será provado por indução sobre o tamanho das palavras w que se $w \in X$ emtão $w \in L$. $\mathtt{0}^0\mathtt{1}^0 = \lambda\lambda = \lambda \in L$. Seja $n \geq 0$, e suponha, como hipótese de indução, que $x \in L$ se $x \in X$, para x tal que $|x| = n$. Para $w \in X$ tal que $|w| = n+1$ tem-se, que w é da forma $\mathtt{0}\mathtt{0}^m\mathtt{1}^n$ ou da forma $\mathtt{1}\mathtt{1}^n$, onde $m, n \geq 0$. No primeiro caso, pela hipótese de indução, segue-se que $\mathtt{0}^m\mathtt{1}^n \in L$; então, pela definição de L, $w \in L$. No segundo caso, como $\mathtt{1}\mathtt{1}^n = \mathtt{1}^n\mathtt{1}$, e, pela hipótese de indução, $\mathtt{1}^n \in L$, tem-se também que, pela definição de L, $w \in L$.

26. a) Definição recursiva de $L_1 = \{w \in \{\mathtt{0}, \mathtt{1}\} \mid |w| \text{ é par}\}$:
        - $\lambda \in L_1$;
        - se $y \in L_1$, então $\mathtt{00}y, \mathtt{01}y, \mathtt{10}y, \mathtt{11}y \in L_1$.

    b) Definição recursiva de $L_2 = \{w \in \{\mathtt{0}, \mathtt{1}\} \mid w \text{ é palíndromo}\}$:
        - $\lambda, \mathtt{0}, \mathtt{1} \in L_2$;
        - se $w \in L_2$, então $\mathtt{0}w\mathtt{0}, \mathtt{1}w\mathtt{1} \in L_2$.

    c) Definição recursiva de $L_3 = \{w \in \{\mathtt{0}, \mathtt{1}\} \mid w \text{ contém } \mathtt{00}\}$:

- $\mathtt{00} \in L_3$;
- se $w \in L_3$, então $\mathtt{0}w, \mathtt{1}w, w\mathtt{0}, w\mathtt{1} \in L_3$.

d) Definição recursiva de $L_4 = \{w \in \{\mathtt{0}, \mathtt{1}\} \mid w$ não contém $\mathtt{00}\}$:

- $\lambda, \mathtt{0} \in L_4$;
- se $y \in L_4$, então $y\mathtt{1}, y\mathtt{10} \in L_4$.

e) Definição recursiva de $L_5 = \{\mathtt{0}^{n^2} \mid n \in \mathbf{N}\}$:

- $\lambda \in L_5$;
- se $y \in L_5$, então $y\mathtt{0}^{2\sqrt{|y|}+1} \in L_5$.

f) Definição recursiva de $L_6 = \{w \mid w$ é uma permutação dos dígitos 1, 2 e 3$\}$:

- $123 \in L_6$;
- se $abc \in L_6$, então $acb \in L_6$ e $bca \in L_6$, onde $a, b, c \in \{1, 2, 3\}$.

g) Definição recursiva de $L_7 = \{w \mid w$ é uma permutação dos 10 dígitos decimais$\}$:

- $\mathtt{0123456789} \in L_7$;
- se $xab \in L_7$ ou $axb \in L_7$, então $xba \in L_7$, onde $a, b \in \{\mathtt{1}, \mathtt{2}\}$ e $x \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^*$.

27. Definição recursiva de *concatenação*:

- $x\lambda = x$ para $x \in \Sigma^*$;
- se $x(ya) = (xy)a$ para $x, y \in \Sigma^*$ e $a \in \Sigma$.

Será provado que $(xy)z = x(yz)$ por indução sobre $|z|$. Para $z = \lambda$, tem-se: $(xy)\lambda = xy = x(y\lambda)$. Seja $n \geq 0$. Suponha que $(xy)z = x(yz)$ para toda palavra z tal que $|z| = n$. Seja w uma palavra de $n+1$ símbolos. Então $w = za$ para uma palavra z de n símbolos e $a \in \Sigma$. Segue-se que $(xy)w = (xy)(za) = ((xy)z)a$, pela definição de concatenação. Prosseguindo: $((xy)z)a = (x(yz))a$, pela hipótese de indução. E, aplicando-se duas vezes a definição de concatenação, determina-se que $(x(yz))a = x((yz)a) = x(y(za))$, o que conclui a argumentação, visto que $x(y(za)) = x(yw)$.

Definição recursiva de *reverso*:

- $\lambda^R = \lambda$;
- se $x^R = y$ e $a \in \Sigma$ então $(xa)^R = ax^R$.

Será provado que $(xy)^R = y^R x^R$ por indução sobre $|y|$. Para $y = \lambda$, tem-se: $(xy)^R = (x\lambda)^R = x^R = \lambda x^R = \lambda^R x^R = y^R x^R$. Seja $n \geq 0$, e suponha que $(xy)^R = y^R x^R$ para palavras y tal que $|y| = n$. Seja w uma palavra de $n+1$ símbolos, ou seja $w = za$ para uma palavra z de n símbolos e $a \in \Sigma$. Então $(xw)^R = (x(za))^R = ((xz)a)^R$, pela definição de concatenação. E $((xz)a)^R = a(xz)^R$, pelas definição de reverso. Pela hipótese de indução, segue-se que $a(xz)^R = a(z^R x^R)$. Pelo resultado acima (associatividade da concatenação), $a(z^R x^R) = (az^R)x^R$, e pela definição de reverso, $(az^R)x^R = (za)^R x^R = w^R x^R$.

A seguir, mostra que se w é palíndromo, $w = vv^R$ ou $w = vav^R$ para algum $v \in \Sigma^*$ e $a \in \Sigma$. Seja w um palíndromo. Se $|w|$ é par, então $w = xy$, onde $|x| = |y| \geq 0$. Como w é palíndromo, $w = xy = (xy)^R = y^R x^R$. De $xy = y^R x^R$ e do fato de que $|x| = |y| = |y^R|$, segue-se que $y^R = x$ (e $x^R = y$). Assim, $w = xy = xx^R$. Agora, suponha que $|w|$ é ímpar. Então $w = xay$, onde $|x| = |y| \geq 0$, $a \in \Sigma$. Como w é palíndromo, $w = xay = (xay)^R = y^R(xa)^R = y^R a x^R$. Como $w = xay = y^R a x^R$ e $|x| = |y| = |y^R|$, segue-se que $y^R = x$ (e $x^R = y$) e, portanto, $w = xay = xax^R$.

28. De $x = x^R$, $y = y^R$ e $xy = (xy)^R = y^R x^R$, deduz-se que $xy = yx$. Será mostrado por indução forte sobre $|xy|$ que, neste caso, existem $k, n \in \mathbf{N}$ e z tais que $x = z^k$ e $y = z^n$. Seja $m \geq 0$ e suponha, como hipótese de indução, que o resultado valha para todo x e y tais que $|xy| < m$. Sejam, então duas palavras quaisquer x e y tais que $|xy| = m$. Se $m = 0$, $x = y = \lambda$, e basta fazer $k = n = 0$. Seja, então, x e y tais que $|xy| \geq 1$. São 3 casos a considerar: $|x| = |y|$, $|x| < |y|$ e $|x| > |y|$. No primeiro caso, tem-se que $x = y$ e, portanto, basta fazer $z = x$ e $k = n = 1$. No segundo caso, deve existir uma palavra s tal que $y = xs$. Como $xy = yx$, segue-se que $xxs = xsx$, que implica que $xs = sx$. Desta última, pela hipótese de indução, existem u, i e j tais que $x = u^i$ e $s = u^j$. Assim, $y = xs = u^{i+j}$. Isto mostra que o resultado vale tomando-se $z = u$, $k = i$ e $n = i + j$. O terceiro caso é similar a este último.

29. Seja $X \subseteq L^*$. Por indução sobre o tamanho de uma palavra w, será mostrado que $w \in L^*$ se, e somente se, $w \in (L^* \cup X)^*$. Tem-se: $\lambda \in (L^* \cup X)^*$ e $\lambda \in L^*$, por definição. Seja uma palavra w tal que $|w| > 0$. Se $w \in L^*$, então, por definição, $w = xy$, onde $x \in L^*$ e $y \in L$. Como $x \in L^*$, $x \in L^* \cup X$ e também $x \in (L^* \cup X)^*$. Como $y \in L$, $y \in L^*$ e também $y \in L^* \cup X$. Logo $xy = w \in (L^* \cup X)^*$. Por outro lado, se $w \in (L^* \cup X)^*$, $w = xy$, onde $x \in (L^* \cup X)^*$ e $y \in L^* \cup X$. Como $X \subseteq L^*$, $y \in L^*$. Pela hipótese de indução, $x \in L^*$. Assim, segue-se que $xy = w \in L^*$.

30. Como Σ_1 e Σ_2 são alfabetos, Σ_1^* e Σ_2^* são enumeráveis. Assim, sejam $f_1 : \Sigma_1^* \to \mathbf{N}$ e $f_2 : \Sigma_2^* \to \mathbf{N}$ funções bijetoras. A função $g : \Sigma_1^* \Sigma_2^* \to \mathbf{N}$, onde $g(xy) = (f_1(x) + f_2(y))(f_1(x) + f_2(y) + 1)/2 + f_1(x)$ para $x \in \Sigma_1^*$ e $y \in \Sigma_2^*$, é bijetora. Logo, $\Sigma_1^* \Sigma_2^*$ é enumerável.

31. Por indução sobre n. O domínio de $H_0 : \Sigma^0 \times \Sigma^0 \to \mathbf{N}$ é $\{(\lambda, \lambda)\}$, e tem-se que $H_0(\lambda, \lambda) = 0$, e portanto $H_0(\lambda, \lambda) = H_0(\lambda, \lambda) + H_0(\lambda, \lambda)$. Seja $n \geq 0$ e suponha, como hipótese de indução, que $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$ para qualquer palavra z de Σ^n. Sejam $a, b, c \in \Sigma$. Será mostrado que $H_{n+1}(xa, yb) \leq H_n(xa, zc) + H_n(zc, yb)$. Considera-se 5 casos:

$a = b = c$.
Então $H_{n+1}(xa, yb) = H_n(x, y)$, $H_{n+1}(xa, zc) = H_n(x, z)$ e $H_{n+1}(zc, yb) = H_n(z, y)$. Como, pela hipótese de indução, $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$, $H_{n+1}(xa, yb) \leq H_n(xa, zc) + H_n(zc, yb)$.

$a = b \neq c$.
Então $H_{n+1}(xa, yb) = H_n(x, y)$, $H_{n+1}(xa, zc) = H_n(x, z) + 1$ e $H_{n+1}(zc, yb) = H_n(z, y) + 1$. Como, pela hipótese de indução, $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$, segue-se que $H_n(x, y) < H_n(x, z) + 1 + H_n(z, y) + 1$ e, portanto, $H_{n+1}(xa, yb) < H_n(xa, zc) + H_n(zc, yb)$.

$a = c \neq b$.
Então $H_{n+1}(xa, yb) = H_n(x, y) + 1$, $H_{n+1}(xa, zc) = H_n(x, z)$ e $H_{n+1}(zc, yb) = H_n(z, y) + 1$. Como, pela hipótese de indução, $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$, segue-se que $H_n(x, y) + 1 \leq H_n(x, z) + H_n(z, y) + 1$ e, portanto, $H_{n+1}(xa, yb) \leq H_n(xa, zc) + H_n(zc, yb)$.

$b = c \neq a$.
Então $H_{n+1}(xa, yb) = H_n(x, y) + 1$, $H_{n+1}(xa, zc) = H_n(x, z) + 1$ e $H_{n+1}(zc, yb) = H_n(z, y)$. Como, pela hipótese de indução, $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$, segue-se que $H_n(x, y) + 1 \leq H_n(x, z) + 1 + H_n(z, y)$ e, portanto, $H_{n+1}(xa, yb) \leq H_n(xa, zc) + H_n(zc, yb)$.

$a \neq b \neq c$.
Então $H_{n+1}(xa, yb) = H_n(x, y) + 1$, $H_{n+1}(xa, zc) = H_n(x, z) + 1$ e $H_{n+1}(zc, yb) =$

$H_n(z, y) + 1$. Como, pela hipótese de indução, $H_n(x, y) \leq H_n(x, z) + H_n(z, y)$, segue-se que $H_n(x, y)+1 < H_n(x, z)+1+H_n(z, y)+1$ e, portanto, $H_{n+1}(xa, yb) < H_n(xa, zc)+H_n(zc, yb)$.

Assim, vê-se que, em qualquer caso, $H_{n+1}(xa, yb) \leq H_n(xa, zc) + H_n(zc, yb)$.

32. a) Gramática para $\{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w \text{ não contém } \mathtt{00}\}$:

$$
\begin{aligned}
P &\to \mathtt{0}A \mid \mathtt{1}P \mid \lambda \\
A &\to \mathtt{1}P \mid \lambda
\end{aligned}
$$

b) Gramática para $\{\mathtt{0}^n\mathtt{1}^{2n+1}\mathtt{0}^n \mid n \in \mathbf{N}\}$:

$$
\begin{aligned}
P &\to \mathtt{0}UP\mathtt{0} \mid \mathtt{1} \\
U\mathtt{1} &\to \mathtt{111} \\
U\mathtt{0} &\to \mathtt{0}U
\end{aligned}
$$

c) Gramática para $\{w\mathtt{0}w \mid w \in \{\mathtt{1}, \mathtt{2}\}^*\}$:

$$
\begin{aligned}
P &\to \mathtt{1}PU \mid \mathtt{2}PD \mid \mathtt{0} \\
\mathtt{0}U &\to \mathtt{01} \\
\mathtt{0}D &\to \mathtt{02} \\
\mathtt{1}U &\to U\mathtt{1} \\
\mathtt{2}U &\to U\mathtt{2} \\
\mathtt{1}D &\to D\mathtt{1} \\
\mathtt{2}D &\to D\mathtt{2}
\end{aligned}
$$

d) Gramática para $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^k \mid 0 \leq n < k\}$:

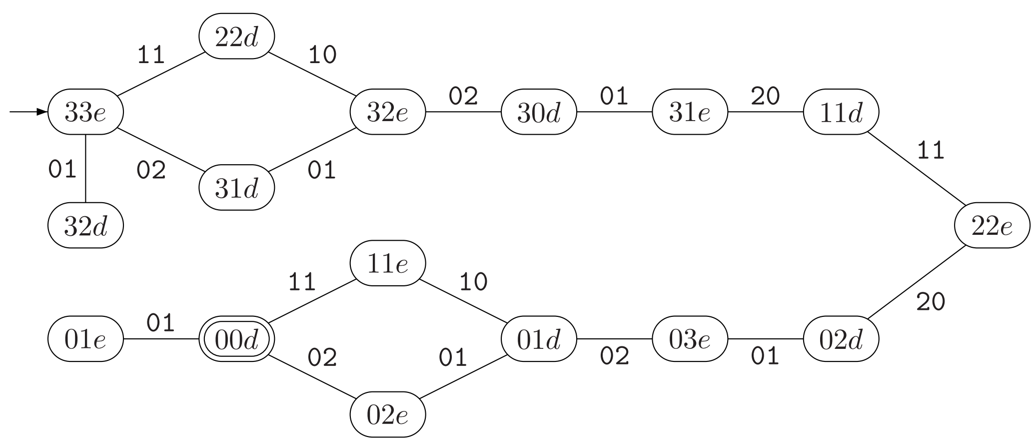
$$
\begin{aligned}
P &\to AS \\
A &\to \mathtt{a}A\mathtt{b}C \mid \lambda \\
C\mathtt{b} &\to \mathtt{b}C \\
C\mathtt{c} &\to \mathtt{cc} \\
S &\to \mathtt{c}S \mid \mathtt{c}
\end{aligned}
$$

Capítulo 2

Máquinas de Estado-Finito

2.1 Alguns Exemplos

1. Cada estado será uma palavra mcl, onde $m \in \{0, 1, 2, 3\}$ é o número de missionários do lado esquerdo, $c \in \{0, 1, 2, 3\}$ é o número de canibais do lado esquerdo, e $l \in \{e, d\}$ é o lado em que está a canoa (e: esquerdo; d: direito). Cada transição terá um um rótulo da forma ij, onde $1 \leq i + j \leq 2$, sendo que i é o número de missionários e j é o número de canibais viajando na canoa. O diagrama de estados está mostrado na figura a seguir. Nele, cada aresta (v_1, r, v_2) representa duas transições: de v_1 para v_2 e de v_2 para v_1, ambas com rótulo r.



2. $M = (\{0, 1, 2, 3\}, \{\mathtt{0}, \mathtt{1}, \mathtt{2}\}, \delta, 0, \{0\})$, onde δ é dada por:

δ	0	1	2
0	0	1	2
1	3	0	1
2	2	3	0
3	1	2	3

3. Observando-se o cruzamento em T, nota-se que são necessários semáforos para controlar o tráfego nos sentidos:

   - de A para B (nome: A_1);

- de A para C (nome: A_2);
- de B para C (nome: B);
- de C para B (nome: C).

Procurando maximizar o número de semáforos abertos (isto é com o farol verde aceso), chega-se a um conjunto de três situações possíveis:

- A_1 e A_2 abertos, B e C fechados;
- A_2 e C abertos, A_1 e B fechados;
- B e C abertos, A_1 e A_2 fechados.

A máquina terá três estados, um para cada uma destas três situações. Para dar nome a um estado serão usados os nomes dos semáforos abertos na situação correspondente. Os estados serão: $\{A_1, A_2\}$, $\{A_2, C\}$ e $\{B, C\}$.

A transição de um estado para outro dependerá das leituras dos sensores. Para cada semáforo há um sensor que verifica se há veículo no sentido controlado por ele. Para os semáforos A_1, A_2, B e C, sejam a_1, a_2, b e c as situações em que o sensores respectivos estão detectando a presença de veículo nos sentidos respectivos. Assim, um subconjunto de $S = \{a_1, a_2, b, c\}$ pode ser usado para indicar os sentidos em que os carros estão se movimentando. Por exemplo, $\emptyset$ indica que nenhum veículo está sendo detectado; $\{a_1, c\}$ indica que só estão sendo detectados veículos nos sentidos de A para B e de C para B. Existirá uma transição de cada estado sob cada subconjunto de S.

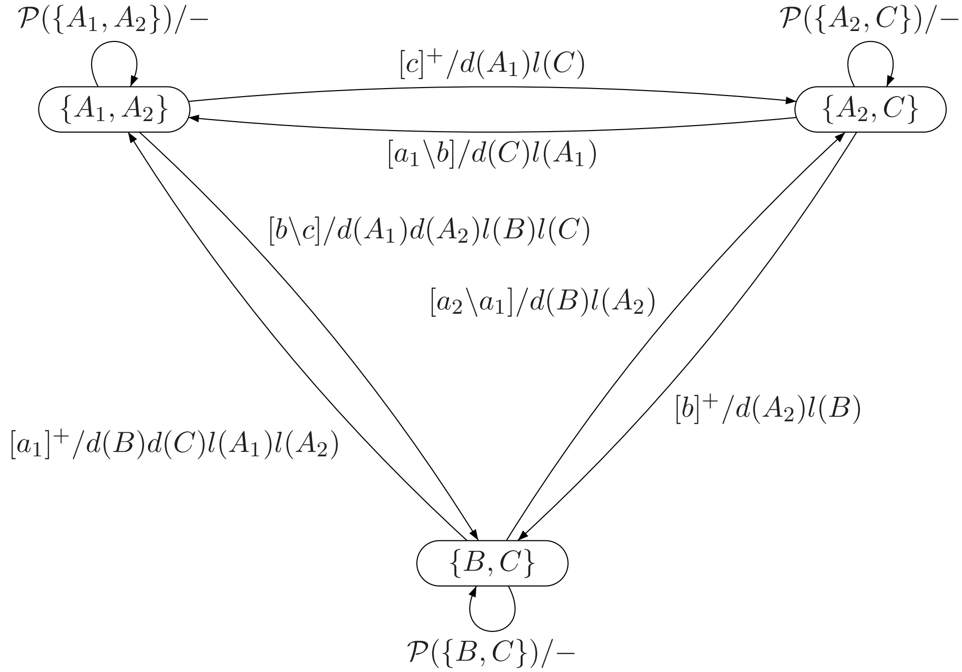
Para cada transição, será especificada como saída uma seqüência de ações do tipo liga/desliga semáforos. A ação "ligar semáforo x", $x \in \{A_1, A_2, B, C\}$, será representada por $l(x)$, e a ação "desligar semáforo x" por $d(x)$. A ação "deixa como está" será representada por $-$.

A figura a seguir mostra um diagrama de estados para o problema. Para simplificar o diagrama, cada aresta representa um conjunto de transições; no diagrama, a saída é a mesma para cada uma de tais transições. Para este fim, os rótulos são da forma X/y, onde X é um conjunto de subconjuntos de S, e y, a saída comum, é uma seqüência de ações liga/desliga.

Alguns valores para X estão indicados através da notação:

- $[s]^+ = \{X \subset S \,|\, s \in X\}$; e
- $[s_1 \backslash s_2] = [s_1]^+ - [s_2]^+$.

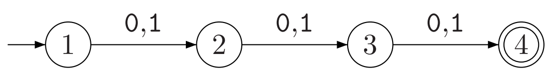
Aqueles da forma $[s]^+$ rotulam transições com maior "prioridade". Por exemplo, no estado $\{A_2, C\}$, dois conjuntos, um com a_1 e outro com b indicam que há veículos nos sentidos dos semáforos A_1 e B. Com isto, deve-se escolher entre a transição para o estado $\{A_1, A_2\}$ e a transição para o estado $\{B, C\}$. Lá está indicado que a transição escolhida é aquela para $\{B, C\}$, pois $[b]^+ = \{X \subset S \,|\, b \in X\}$ e $[a_1 \backslash b] = [a_1]^+ - [b]^+$ (veja a figura).



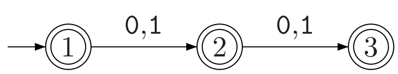
Pode-se mostrar que, nesta solução, não há possibilidade de um veículo ser impedido de se dirigir de algum ponto a outro indefinidamente. Por outro lado, a solução apresentada não é única.

2.2 Autômatos Finitos Determinísticos

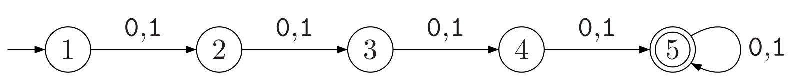
1. a)



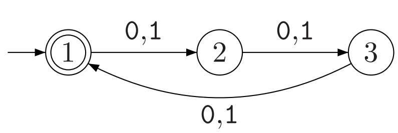
b)



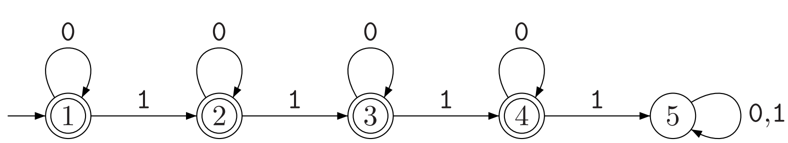
c)



d)



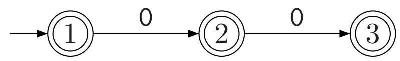
e)



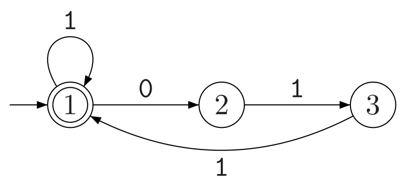
f) Um AFD seria aquele com estados $E = (\{0,1,2\} \times \{0,1,2\}) \cup \{e\}$, sendo e um estado de erro, e cada estado restante, da forma (i, j), atingido quando a palavra de entrada w for tal que $|w| \bmod 3 = i$ e w tiver j ocorrências de `1`. O estado inicial é $(0, 0)$, os estados finais são $(0, 1)$ e $(0, 2)$, e a função de transição é dada por:

- $\delta((i,j),0) = ((i+1) \bmod 3, j)$ para $i,j \in \{0,1,2\}$;
- $\delta((i,j),1) = ((i+1) \bmod 3, j+1)$ para $i \in \{0,1,2\}$ e $j \in \{0,1\}$;
- $\delta((i,2),1) = e$ para $i \in \{0,1,2\}$; e
- $\delta(e,a) = e$ para $a \in \{0,1\}$.

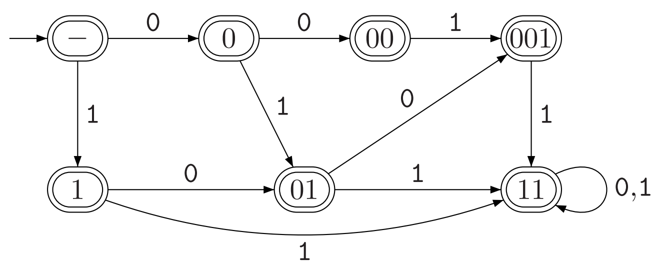
2. a)



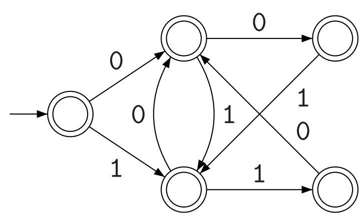
b)



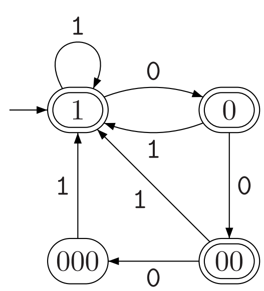
c)



d)



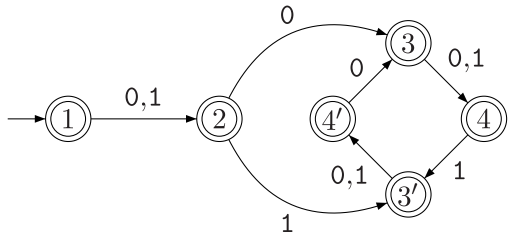
e)



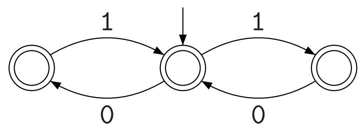
f) Sejam $\Sigma = \{0,1,2\}$ e $E = \{xyz \mid x,y,z \in \{p,i\}\}$. Suponha que $\overline{p} = i$ e $\overline{i} = p$. Então o AFD é $(E, \Sigma, \delta, ppp, \{ppp\})$, onde δ é dada por:

$$\delta(xyz, a) = \begin{cases} \overline{x}yz & \text{se } a = 0 \\ x\overline{y}z & \text{se } a = 1 \\ xy\overline{z} & \text{se } a = 2 \end{cases}$$

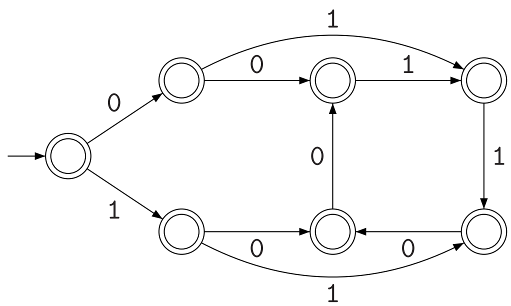
3. a)



b)



c)



d) Após consumido um prefixo x, deve-se saber, para prosseguir-se com o reconhecimento:

- se $|x|$ é par ou ímpar (para se saber se a próxima posição é par ou ímpar);
- se x tem número par ou ímpar de 0s nas posições pares; e
- se x tem número par ou ímpar de 0s nas posições ímpares.

Logo, são necessários 8 estados. Representado-se cada estado como uma tripla $[k_1, k_2, k_3]$, de forma que cada k_i pode ser p (par) ou i (ímpar), e

- $k_1 = p \leftrightarrow |x|$ é par,
- $k_2 = p \leftrightarrow x$ tem número par de 0s nas posições pares, e
- $k_3 = p \leftrightarrow x$ tem número par de 0s nas posições ímpares,

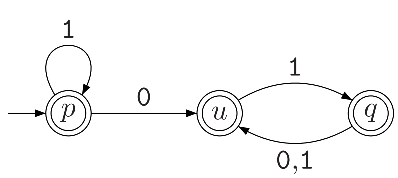
um AFD para a linguagem em questão seria

$$(\{p, i\}^3, \{\mathtt{0}, \mathtt{1}\}, \delta, [p, p, p], \{[p, p, i], [i, p, i]\}\})$$

sendo δ dada por $\delta([k_1, k_2, k_3], \mathtt{0}) = [k'_1, k'_2, k'_3]$ e $\delta([k_1, k_2, k_3], \mathtt{1}) = [k'_1, k_2, k_3]$, onde:

- $k'_1 = p \leftrightarrow k_1 = i$
- $k'_2 = p \leftrightarrow (k_1 = p$ e $k_2 = p)$ ou $(k_1 = i$ e $k_2 = i)$
- $k'_3 = p \leftrightarrow (k_1 = i$ e $k_3 = p)$ ou $(k_1 = p$ e $k_3 = i)$

e)



4. Um AFD equivalente: $M' = (E \cup \{g, f\}, \Sigma, \delta', i, F \cup \{f\})$, onde $g, f \notin E$ e δ' é como segue:

- $\delta'(e, a) = \delta(e, a)$ para todo $a \in \Sigma$ tal que $\delta(e, a)$ é definido;
- $\delta'(e, a) = g$ para todo $a \in \Sigma$ tal que $\delta(e, a)$ é indefinido e $e \notin F$;
- $\delta'(e, a) = f$ para todo $a \in \Sigma$ tal que $\delta(e, a)$ é indefinido e $e \in F$;
- $\delta'(g, a) = g$ para todo $a \in \Sigma$; e
- $\delta'(f, a) = f$ para todo $a \in \Sigma$.

5. Por indução sobre $|x|$. Para $x = \lambda$, $\hat{\delta}(e, \lambda y) = \hat{\delta}(\hat{\delta}(e, \lambda), y)$, pela definição de $\hat{\delta}$. Suponha, como hipótese de indução, que $\hat{\delta}(e, xy) = \hat{\delta}(\hat{\delta}(e, x), y)$ para palavras x de tamanho n, $n \geq 0$. Para palavras de tamanho $n + 1$, az, onde a é um símbolo do alfabeto e z uma palavra de tamanho n, tem-se:

$$\begin{array}{lll} \hat{\delta}(e, azy) & = \hat{\delta}(\delta(e, a), zy) & \text{pela definição de } \hat{\delta} \\ & = \hat{\delta}(\hat{\delta}(\delta(e, a), z), y) & \text{pela hipótese de indução} \\ & = \hat{\delta}(\hat{\delta}(az), y) & \text{pela definição de } \hat{\delta}. \end{array}$$

6. Em Prolog, um AFD poderia ser representado por meio de 3 predicados:

   `inicial(E)`: `E` é um estado inicial;

   `final(E)`: `E` é um estado final;

   `delta(E1,A,E2)`: $\delta(\mathtt{E1}, \mathtt{A}) = \mathtt{E2}$.

   Representando uma palavra por meio de lista, λ sendo representanda pela lista vazia, um programa que retorna *yes* se uma palavra `W` é aceita, e retorna *no* em caso contrário, seria:

```
aceita(W) :-
   inicial(I),
   deltachapeu(I,W,E),
   final(E).
```

   onde `deltachapeu` é assim definida:

```
deltachapeu(E,[],E).
deltachapeu(E,[X|R],ER) :-
   delta(E,X,EX),
   deltachapeu(EX,R,ER).
```

7. Basta substituir o comando

   $e \leftarrow D[e, s]$

   por um comando da forma:

   caso e **seja**
   pp: **se** $s = \mathsf{0}$ **então** $e \leftarrow ip$ **senão** $e \leftarrow pi$ **fimse**
   ip: **se** $s = \mathsf{0}$ **então** $e \leftarrow pp$ **senão** $e \leftarrow ii$ **fimse**
   pi: **se** $s = \mathsf{0}$ **então** $e \leftarrow ii$ **senão** $e \leftarrow pp$ **fimse**
   ii: **se** $s = \mathsf{0}$ **então** $e \leftarrow pi$ **senão** $e \leftarrow ip$ **fimse**
   fimcaso

8. A definição 6 diz que $e \approx e'$ se, e somente se:

$$\hat{\delta}(e, y) \in F \text{ se, e somente se, } \hat{\delta}(e', y) \in F.$$

para toda palavra y. Então, que $\approx$ é uma relação de equivalência, segue diretamente do fato de que $\leftrightarrow$ ("se e somente se") é também uma relação de equivalência.

9. Deve-se provar que $[e]_n = \{e' \mid e \approx_n e'\}$. Será feita uma prova por indução sobre n. Para o caso em que $e \in F$, $[e]_0 = F$, por definição. E, ainda para o caso em que $e \in F$, pela Definição 8 tem-se que para todo e' $e \approx_0 e'$ se, e somente se, $e' \in F$; logo, $\{e' \mid e \approx_n e'\} = F$. Portanto, $[e]_0 = F = \{e' \mid e \approx_0 e'\}$ se $e \in F$. Para o caso em que $e \notin F$, analogamente se mostra que $[e]_0 = E - F = \{e' \mid e \approx_0 e'\}$. Seja $n \geq 0$ e suponha, como hipótese de indução, $[e]_n = \{e' \mid e \approx_n e'\}$. Por definição, $[e]_{n+1} = \{e' \in [e]_n \mid [\delta(e', a)]_n = [\delta(e, a)]_n$ para todo $a \in \Sigma\}$. Assim, basta mostrar que para todo $e' \in E$,

$$e' \in [e]_n \text{ e } \forall\, a \in \Sigma[\delta(e', a)]_n = [\delta(e, a)]_n \text{ se, e somente se, } e \approx_{n+1} e'.$$

Pela hipótese de indução, tem-se que $e' \in [e]_n$ se, e somente se, $e \approx_n e'$. Pela hipótese de indução, tem-se ainda que para qualquer $a \in \Sigma$, $e'' \in [\delta(e, a)]_n$ se, e somente se, $e'' \approx_n \delta(e, a)$. Desta última, segue-se que $[\delta(e', a)]_n = [\delta(e, a)]_n$ se, e somente se, $\delta(e', a) \approx_n \delta(e, a)$. Logo, tem-se que: $e' \in [e]_n$ e $\forall\, a \in \Sigma[\delta(e', a)]_n = [\delta(e, a)]_n$ se, e somente se, $e \approx_n e'$ e $\delta(e, a) \approx_n \delta(e', a)$. Finalmente, usando-se a definição 8, conclui-se que $e' \in [e]_n$ e $\forall\, a \in \Sigma$ $[\delta(e', a)]_n = [\delta(e, a)]_n$ se, e somente se, $e \approx_{n+1} e'$.

10. Evolução das partições do conjunto de estados:

S_0: $\{[\lambda, \lambda], [c1, t0]\}, \{[c0, t0], [c0t1], [c1, t1]\}$
S_1: $\{[\lambda, \lambda]\}, \{[c1, t0]\}, \{[c0, t0], [c0t1]\}, \{[c1, t1]\}$
S_2: $\{[\lambda, \lambda]\}, \{[c1, t0]\}, \{[c0, t0], [c0t1]\}, \{[c1, t1]\}$

11. Cada estado do AFD original é representado pela palavra que leva do estado inicial até ele. O estado de erro é representado por "*". As partições do conjunto de estados evoluem assim (tal sequência foi obtida por meio de um programa feito pelo autor em Prolog):

S_0: {{λ,al,alm,as,b,ba,bar,barc,br,bra,bras,bro,c,cal,calm,cas,d,di,dis,disc,*}, {a,alma,asa,barco,brasa,broa,ca,calma,casa,disco}}.

S_1: {{λ,alm,as,bras,bro,c,calm,cas}, {al,b,ba,bar,br,bra,cal,d,di,dis,*}, {barc,disc}, {a,alma,asa,barco,brasa,broa,ca,calma,casa,disco}}.

S_2: {{λ}, {alm,as,bras,bro,c,calm,cas}, {al,cal}, {b,ba,d,di,*}, {bar,dis}, {br}, {bra}, {barc,disc}, {a,ca}, {alma,asa,barco,brasa,broa,calma,casa,disco}}.

S_3: {{λ}, {alm,as,bras,bro,calm,cas}, {c}, {al,cal}, {b}, {ba}, {d,*}, {di}, {bar,dis}, {br}, {bra}, {barc,disc}, {a,ca}, {alma,asa,barco,brasa,broa,calma,casa,disco}}.

S_4: {{λ}, {alm,as,bras,bro,calm,cas}, {c}, {al,cal}, {b}, {ba}, {d}, {*}, {di}, {bar,dis}, {br}, {bra}, {barc,disc}, {a,ca}, {alma,asa,barco,brasa,broa,calma,casa,disco}}.

S_5: {{λ}, {alm,as,bras,bro,calm,cas}, {c}, {al,cal}, {b}, {ba}, {d}, {*}, {di}, {bar,dis}, {br}, {bra}, {barc,disc}, {a,ca}, {alma,asa,barco,brasa,broa,calma,casa,disco}}.

12. Os AFDs têm 12 estados: aqueles de $\{1, 2, 3\} \times \{1, 2, 3, e\}$. O estado e é um estado de erro, necessário para completar o diagrama de estados simplificado mostrado no Exercício 2(b). O estado inicial é $[1, 1]$. A função de transição é dada por:

δ	0	1
$[1,1]$	$[2,2]$	$[2,1]$
$[1,2]$	$[2,e]$	$[2,3]$
$[1,3]$	$[2,e]$	$[2,1]$
$[1,e]$	$[2,e]$	$[2,e]$

δ	0	1
$[2,1]$	$[3,2]$	$[3,1]$
$[2,2]$	$[3,e]$	$[3,3]$
$[2,3]$	$[3,e]$	$[3,1]$
$[2,e]$	$[3,e]$	$[3,e]$

δ	0	1
$[3,1]$	$[1,2]$	$[1,1]$
$[3,2]$	$[1,e]$	$[1,3]$
$[3,3]$	$[1,e]$	$[1,1]$
$[3,e]$	$[1,e]$	$[1,e]$

Os estados finais são:

a) Para a união: $\{[1,1],[1,2],[1,3],[1,e],[2,1],[3,1]\}$.

b) Para a interseção: $\{[1,1]\}$.

13. Basta aplicar o produto aos AFDs M_1' e M_2' obtidos como abaixo, ao invés de aplicá-lo diretamente sobre M_1 e M_2. Fora isto, o lema e o teorema ficam inalterados. Dados os AFDs $M_1 = (E_1, \Sigma_1, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma_2, \delta_2, i_2, F_2)$, obtém-se M_1' assim:

    - se $\Sigma_2 - \Sigma_1 = \emptyset$, então $M_1' = M_1$;
    - se $\Sigma_2 - \Sigma_1 \neq \emptyset$, então $M_1' = (E_1 \cup \{d\}, \Sigma_1 \cup \Sigma_2, \delta_1', i_1, F_1)$, onde:
        - $d \notin E_1$; e
        - $\delta_1'(e,a) = \delta(e,a)$, se $a \in \Sigma_1$;
          $\delta_1'(e,a) = d$, se $a \in \Sigma_2 - \Sigma_1$;
          $\delta_1'(d,a) = d$, para todo $a \in \Sigma_1 \cup \Sigma_2$.

    M_2' é obtido de forma análoga. Por tal construção, M_1' e M_2' têm o mesmo alfabeto $(\Sigma_1 \cup \Sigma_2)$, $L(M_1') = L(M_1)$ e $L(M_2') = L(M_2)$. Assim, o lema e o teorema ficam como estão, apenas trocando-se M_1 por M_1' e M_2 por M_2'.

14. Uma possibilidade é construir, inicialmente, um AFD em formato de árvore, como mostrado no livro, e depois aplicar o algoritmo de minimização. Um algoritmo de minimização específico para conjuntos finitos de palavras, cujo tempo de execução é linear, é o de Revuz[1]. Um outro para construção incremental, ou seja, para inserir uma palavra em um AFD mínimo de forma a obter um AFD também mínimo, em tempo linear, é o de Sgarbas et al.[2].

15. Suponha que existe um AFD que reconheça a linguagem L, e seja k o número de estados do mesmo. Seja a palavra de L, $z = \mathtt{a}^k\mathtt{c}\mathtt{a}^k$. De acordo com o Teorema, existem u, v e w tais que $z = uvw$, $v \neq \lambda$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, analisando-se os valores possíveis para v, conclui-se que $uv^2w \notin L$:

    - se v contém apenas as, então $uv^2w = \mathtt{a}^{k+|v|}\mathtt{c}\mathtt{a}^k$ ou $uv^2w = \mathtt{a}^k\mathtt{c}\mathtt{a}^{k+|v|}$; e
    - se v contém c, então uv^2w contém mais de um c.

    Contradição. Logo, não existe AFD que reconhece L.

16. a) Suponha que existe um AFD que reconheça a linguagem $\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^n \mid n \in \mathbf{N}\}$, e seja k seu número de estados. Seja a palavra $z = \mathtt{0}^k\mathtt{1}^k\mathtt{0}^k \in L$. De acordo com o Teorema no final da seção, existem u, v e w tais que $z = uvw$, $v \neq \lambda$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, como mostrado abaixo, $uv^2w \notin L$:

[1] Revuz, D. Minimization of acyclic deterministic automa in linear time, *Theoretical Computer Science 92*, 1992, pp. 181–189.

[2] Sgarbas, K.N, Fakotakis, N.D., Kokkinakis, G.K. Optimal insertion in deterministic DAWGs, *Theoretical Computer Science 301*, 2003, pp. 103–117.

- se v contém apenas 0s do prefixo de k 0s, $uv^2w = 0^{k+|v|}1^k0^k$;
- se v contém i 0s do prefixo e j 1s, onde $i > 0$, $j > 0$ e $i + j = |v|$, então $uv^2w = 0^k1^j0^i1^k0^k$;
- se v contém apenas 1s, $uv^2w = 0^k1^{k+|v|}0^k$;
- se v contém i 1s e j 0s do sufixo, onde $i > 0$, $j > 0$ e $i + j = |v|$, então $uv^2w = 0^k1^k0^j1^i0^k$;
- se v contém apenas 0s do sufixo de k 0s, $uv^2w = 0^k1^k0^{k+|v|}$.

Contradição. Logo, não existe AFD que reconhece L.

b) Um AFD que reconhece $\{0^n0^n0^n \mid n \in \mathbf{N}\}$: $(\{1, 2, 3\}, \{0\}, \delta, 1, \{1\})$, onde δ é dada por: $\delta(1, 0) = 2$, $\delta(2, 0) = 3$, $\delta(3, 0) = 1$.

17. a) Um PD seria aquele que verifica se alguma das $|\Sigma|^n$ palavras de tamanho n é reconhecida por M ou não.

b) Seja k o número de estados de M. Um PD seria aquele que verifica se alguma das Σ^{k-1} palavras w tais que $n < |w| < n+k$ é reconhecida por M ou não. Para ver que nenhuma palavra w tal que $|w| \geq n+k$ precisa ser considerada, será mostrado que se $w \in L(M)$ e $|w| \geq n + k$, então há uma palavra z de $L(M)$ tal que $n < |z| < n + k$. Suponha que $w \in L(M)$ e $|w| \geq n+k$. Então há uma palavra o *menor* possível, w' tal que $w' \in L(M)$ e $|w'| \geq n + k$ (que pode ser w ou não). Pelo lema do bombeamento, existem u, v e x tais que $|v| > 0$, $|uv| \leq k$ e $uv^ix \in L(M)$ para todo $i \geq 0$. Mas, para u, v e x tais que $|v| > 0$ e $|uv| \leq w$, tem-se que $uv^0x \in L(M)$; assim, como uvx é a *menor* palavra de $L(M)$ maior ou igual a $n + k$, $|ux| < n + k$; e como $|v| \leq k$, $|ux| > n$.

c) Se M tem k estados, a primeira ocorrência de a em uma palavra, se houver alguma, deve ser, no mais tardar, o $k - 1$-ésimo símbolo da mesma. A segunda ocorrência deve ser, no mais tardar, o $2k - 1$-ésimo símbolo. E assim por diante. Conclui-se que um PD precisa verificar o reconhecimento por M de palavras de, no máximo, $nk - 1$ símbolos.

18. a) Sim. O AFD $(\{i\}, \{0\}, \delta, i, \{i\})$ tal que $\delta(i, 0) = i$ reconhece $\{0\}^*$.

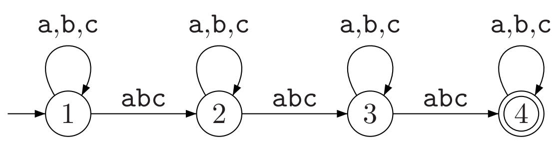
b) Sim. O número de estados de um AFD que reconhece uma linguagem finita deve ter no mínimo o tamanho da maior palavra reconhecida mais 1. Por exemplo, se uma palavra tiver 999.999.999.999 símbolos, o AFD terá no mínimo 1 trilhão de estados.

c) Não. $\{0\}^*\{1\}^*$ pode ser reconhecida, e $\{0^n1^n \mid n \geq 0\}$ não pode.

d) Não. Valem os exemplos do item anteior. Observe que as afirmativas são logicamente equivalentes.

2.3 Autômatos Finitos Não Determinísticos

1. a) Segue o diagrama de estados para um AFNE (para economizar estados):



b) O diagrama de estados para um AFN está mostrado na Figura 2.1.

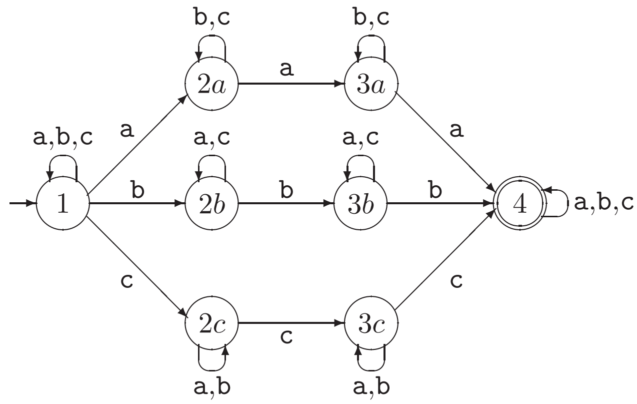


Figura 2.1: AFN para item (b).

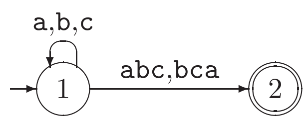


Figura 2.2: AFNE para item (c).

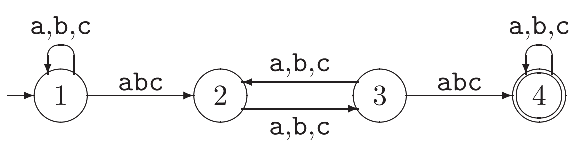


Figura 2.3: AFNE para ıtem (d).

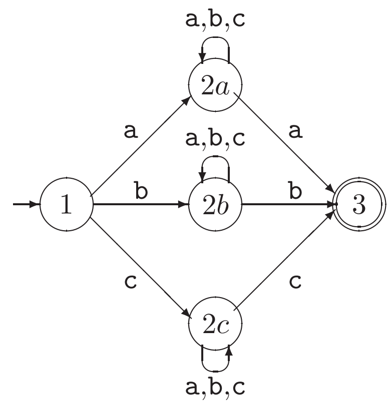


Figura 2.4: AFN para item (e).

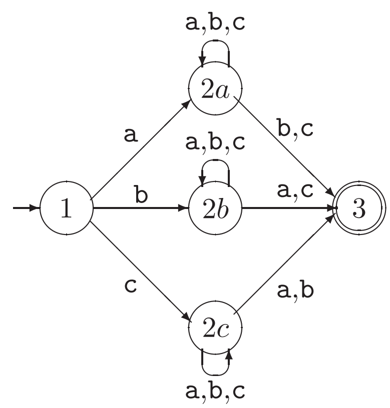


Figura 2.5: AFN para item (f).

c) O diagrama de estados para um AFNE está mostrado na Figura 2.2.

d) O diagrama de estados para um AFNE está mostrado na Figura 2.3.

e) O diagrama de estados para um AFN está mostrado na Figura 2.4.

f) O diagrama de estados para um AFN está mostrado na Figura 2.5.

g) O diagrama de estados para um AFN está mostrado na Figura 2.6.

h) O diagrama de estados para um AFN está mostrado na Figura 2.7.

i) O diagrama de estados para um AFN está mostrado na Figura 2.8.

2. Menor número de estados para AFNs e AFDs que reconheçam $L_n = \{xyx \mid x, y \in \{\mathtt{a}, \mathtt{b}\} \text{ e } |x| = n\}$:

   a) $n = 1$. Menor AFN: 4. Menor AFD: 5.

   b) $n = 2$. Menor AFN: 10. Menor AFD: 15.

   c) n arbitrário. Menor AFN: $3.2^n - 2$. Menor AFD: $(n+2).2^n - 1$.

3. Será mostrado, por indução sobre $|w|$, que $\hat{\delta}(\bigcup_{i=1}^{n} X_i, w) = \bigcup_{i=1}^{n} \hat{\delta}(X_i, w)$ para qualquer $w \in \Sigma^*$. Inicialmente, observe que $\hat{\delta}(\bigcup_{i=1}^{n} X_i, \lambda) = \bigcup_{i=1}^{n} X_i$, pela definição de $\hat{\delta}$.

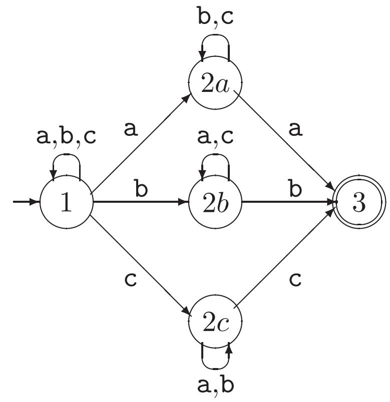


Figura 2.6: AFN para item (g).

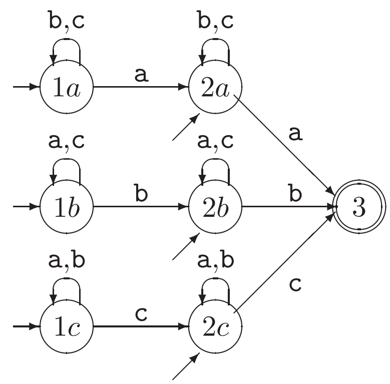


Figura 2.7: AFN para item (h).

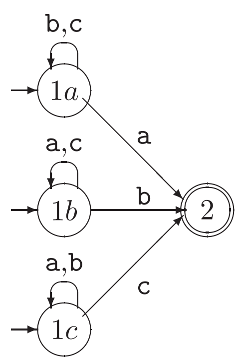


Figura 2.8: AFN para item (i).

Como, por esta mesma definição, $\bigcup_{i=1}^{n} X_i = \bigcup_{i=1}^{n} \hat{\delta}(X_i, \lambda)$, segue-se que $\hat{\delta}(\bigcup_{i=1}^{n} X_i, \lambda) = \bigcup_{i=1}^{n} \hat{\delta}(X_i, \lambda)$. Seja n um número natural arbitrário e suponha, como hipótese de indução, que $\hat{\delta}(\bigcup_{i=1}^{n} X_i, w) = \bigcup_{i=1}^{n} \hat{\delta}(X_i, w)$ para qualquer $w \in \Sigma^*$ de n símbolos. Basta, então, provar que o resultado vale para palavras de $n+1$ símbolos. Para isto, seja uma palavra arbitrária, ay, em que $a \in \Sigma$ e $|y| = n$. Por definição de $\hat{\delta}$, tem-se que $\hat{\delta}(\bigcup_{i=1}^{n} X_i, ay) = \hat{\delta}(\bigcup_{e \in \bigcup_{i=1}^{n} X_i} \delta(e,a), y)$. Pela hipótese de indução, esta é igual a $\bigcup_{e \in \bigcup_{i=1}^{n} X_i} \hat{\delta}(\delta(e,a), y)$. Como esta última é o mesmo que $\bigcup_{i=1}^{n} [\bigcup_{e \in X_i} \hat{\delta}(\delta(e,a), y)]$ que, por sua vez é igual a $\bigcup_{i=1}^{n} \hat{\delta}(\bigcup_{e \in X_i} \delta(e,a), y)$ pela hipótese de indução, tem-se a conclusão esperada.

4. Seja um AFN $M = (E, \Sigma, \delta, I, F)$. Um AFN com único estado inicial que reconhece $L(M)$ seria $M' = (E \cup \{i\}, \Sigma, \delta', i, F')$, onde i é um estado que não pertence a E e:

   - $F' = F$ se $I \cap F = \emptyset$; caso contrário, $F' = F \cup \{i\}$;
   - $\delta'(e,a) = \delta(e,a)$ para todo $(e,a) \in E \times \Sigma$;
   - $\delta'(i,a) = \bigcup_{e \in I} \delta(e,a)$ para todo $a \in \Sigma$.

5. Um AFN com *um único estado final* equivalente a M seria $M' = (\{1,2,3,f\}, \{a,b\}, \delta', \{1\}, \{1,f\})$, onde δ é dada por:

δ	a	b
1	$\{2,f\}$	$\{\}$
2	$\{3,f\}$	$\{\}$
3	$\{\}$	$\{3,f\}$

6. Seja um AFN $M = (E, \Sigma, \delta, I, F)$. Um AFN com único estado final que reconhece $L(M)$ seria $M' = (E \cup \{f\}, \Sigma, \delta', I', \{f\})$, onde f é um estado que não pertence a E e:

   - $I' = I$ se $I \cap F = \emptyset$; caso contrário, $I' = I \cup \{f\}$;
   - $\delta'(e,a) = \delta(e,a)$ se $\delta(e,a) \cap F = \emptyset$; caso contrário, $\delta'(e,a) = \delta(e,a) \cup \{f\}$;

7. a) $f\lambda(0) = \{0,1,2\}$; $f\lambda(1) = \{1,2\}$; $f\lambda(2) = \{2\}$.

   b) AFN $M' = (\{0,1,2\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \delta', \{0,1,2\}, \{2\})$, sendo δ' dada por:

δ'	a	b	c
0	$\{0,1,2\}$	$\emptyset$	$\emptyset$
1	$\emptyset$	$\{1,2\}$	$\emptyset$
2	$\emptyset$	$\emptyset$	$\{2\}$

   c) AFD $M'' = (\{\{0,1,2\}, \{1,2\}, \{2\}, \emptyset\}, \{\mathtt{a},\mathtt{b},\mathtt{c}\}, \delta'', \{0,1,2\}, \{\{0,1,2\}, \{1,2\}, \{2\}\})$, sendo δ'' dada por:

δ''	a	b	c
$\{0,1,2\}$	$\{0,1,2\}$	$\{1,2\}$	$\{2\}$
$\{1,2\}$	$\emptyset$	$\{1,2\}$	$\{2\}$
$\{2\}$	$\emptyset$	$\emptyset$	$\{2\}$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

8. AFD $(\{\{1,2\}, \{2\}, \{2'\}, \{3\}, \{3'\}\}, \{\mathtt{0},\mathtt{1}\}, \delta, \{1,2\}, \{\{1,2\}, \{2\}, \{3\}\})$, em que δ é dada por:

δ	`0`	`1`
$\{1,2\}$	$\{2'\}$	$\{3\}$
$\{2'\}$	$\{2\}$	$\emptyset$
$\{2\}$	$\{2'\}$	$\emptyset$
$\{3\}$	$\emptyset$	$\{3'\}$
$\{3'\}$	$\emptyset$	$\{3\}$
$\emptyset$	$\emptyset$	$\emptyset$

9. AFD $(\{\{p_0,p_1\},\{i_0,p_1\},\{i_0,i_1\},\{p_0,i_1\},\{p_0,p_1,i_1\},\{i_0,p_1,i_1\}\},\{\mathtt{0},\mathtt{1}\},\delta,\{p_0,p_1\},F)$, em que $F=\{\{i_0,i_1\},\{p_0,i_1\},\{p_0,p_1,i_1\},\{i_0,p_1,i_1\}\})$ e δ é dada por:

δ	`0`	`1`
$\{p_0,p_1\}$	$\{i_0,p_1\}$	$\{p_0,p_1,i_1\}$
$\{i_0,p_1\}$	$\{p_0,p_1\}$	$\{i_0,i_1\}$
$\{i_0,i_1\}$	$\{p_0,i_1\}$	$\{i_0,p_1\}$
$\{p_0,i_1\}$	$\{i_0,i_1\}$	$\{p_0,p_1\}$
$\{p_0,p_1,i_1\}$	$\{i_0,p_1,i_1\}$	$\{p_0,p_1,i_1\}$
$\{i_0,p_1,i_1\}$	$\{p_0,p_1,i_1\}$	$\{i_0,p_1,i_1\}$

10. Na questão 1 já foram apresentados os AFNEs.

11. Seja um AFNE $M=(E,\Sigma,\delta,I,F)$. A função de transição estendida para M, $\hat{\delta}:\mathcal{P}(E)\times\Sigma^*\to\mathcal{P}(E)$, pode ser definida recursivamente assim:

    a) $\hat{\delta}(\emptyset,w)=\emptyset$, para todo $w\in\Sigma^*$;

    b) $\hat{\delta}(A,\lambda)=A$, para todo $A\subseteq E$;

    c) $\hat{\delta}(A,w)=\bigcup_{e\in A}\bigcup_{xy=w}\hat{\delta}(\delta(e,x),y)$, para $A\subseteq E$.

    $\bigcup_{xy=w}$ quer dizer *união para todas as palavras x e y tais que $xy=w$.*

12. Dados dois AFs $M_1=(E_1,\Sigma_1,\delta_1,I_1,F_1)$ e $M_2=(E_2,\Sigma_2,\delta_2,I_2,F_2)$ em que $E_1\cap E_2=\emptyset$, um AFNλ que reconhece $L(M_1)\cup L(M_2)$ seria $M_3=(E_1\cup E_2\cup\{i\},\Sigma_1\cup\Sigma_2,\delta_3,\{i\},F_1\cup F_2)$, em que δ_3 é constituída das transições em δ_1 e δ_2 mais $\delta_3(i,\lambda)=I_1\cup I_2$ e $\delta_3(i,a)=\emptyset$ para todo $a\in\Sigma\cup\Sigma_2$, desde que $i\notin E_1\cup E_2$.

13. Como X é finito, pois é um subconjunto de E, e cada $\delta(e,a)$, para $e\in X$ e $a\in\Sigma$, é um conjunto de estados, basta mostrar que $f_\lambda(A)\cup f_\lambda(B)=f\lambda(A\cup B)$ para quaisquer $A,B\subseteq E$. Sejam, então, A e B subconjuntos arbitrários de E.

    $f_\lambda(A)\cup f_\lambda(B)\subseteq f\lambda(A\cup B)$.
    Seja $e\in f_\lambda(A)\cup f_\lambda(B)$. Tem-se dois casos a considerar: $e\in f_\lambda(A)$ e $e\in f_\lambda(B)$. Suponha, inicialmente, que $e\in f_\lambda(A)$. Será mostrado, por indução sobre o número mínimo de transições λ necessários para inclusão de um estado em $f\lambda(A)$, que $e\in f_\lambda(A\cup B)$. Se este número é zero, $e\in A$ pela definição de $f\lambda$; segue-se que $e\in A\cup B$. Novamente, pela definição de $f\lambda$, $e\in f_\lambda(A\cup B)$. Suponha, como hipótese de indução, que se o número mínimo de transições λ necessários para inclusão de um estado em $f\lambda(A)$ for n ($n\geq 0$), $e\in f_\lambda(A\cup B)$. Seja $n+1$ o número mínimo de transições λ necessários para inclusão de um certo estado e em $f\lambda(A)$; neste caso, pela definição de $f\lambda$, existe um estado e' tal que $e\in\delta(e',\lambda)$ e $e'\in f\lambda(A)$. Pela hipótese de indução, tem-se que $e'\in f\lambda(A\cup B)$. Pela definição de $f\lambda$, como $e\in\delta(e',\lambda)$ e $e'\in f\lambda(A\cup B)$, conclui-se que $e\in f\lambda(A\cup B)$. Portanto, $f_\lambda(A)\subseteq f\lambda(A\cup B)$. De forma análoga, mostra-se que $f_\lambda(B)\subseteq f\lambda(A\cup B)$. Conclusão: $f_\lambda(A)\cup f_\lambda(B)\subseteq f\lambda(A\cup B)$.

$f\lambda(A \cup B) \subseteq f_\lambda(A) \cup f_\lambda(B)$.
Seja $e \in f\lambda(A \cup B)$. Será mostrado, por indução sobre o número mínimo de transições λ necessários para inclusão de um estado em $f\lambda(A \cup B)$, que $e \in f_\lambda(A) \cup f_\lambda(B)$. Se este número é zero, $e \in A \cup B$ pela definição de $f\lambda$; segue-se que $e \in f_\lambda(A)$, se $e \in A$, e $e \in f_\lambda(B)$, se $e \in B$; logo, $e \in f_\lambda(A) \cup f_\lambda(B)$. Seja $n \geq 0$, e suponha, como hipótese de indução, que se o número mínimo de transições λ necessários para inclusão de um estado em $f\lambda(A \cup B)$ for n, $e \in f_\lambda(A) \cup f_\lambda(B)$. Seja $n + 1$ o número mínimo de transições λ necessários para inclusão de um certo estado e em $f\lambda(A \cup B)$; neste caso, pela definição de $f\lambda$, existe um estado e' tal que $e \in \delta(e', \lambda)$ e $e' \in f\lambda(A \cup B)$. Pela hipótese de indução, tem-se que $e' \in f\lambda(A) \cup f_\lambda(B)$. Pela definição de $f\lambda$, como $e \in \delta(e', \lambda)$, se $e' \in f\lambda(A)$, $e \in f\lambda(A)$, e se $e' \in f\lambda(B)$, $e \in f\lambda(B)$. Conclui-se que $e \in f\lambda(A) \cup f\lambda(B)$. Portanto, $f_\lambda(A \cup B)E \subseteq f\lambda(A) \cup f\lambda(B)$.

14. Seja um AFNλ $M = (E, \Sigma, \delta, I, F)$. Um AFN$\lambda$ equivalente, como pedido, seria $M' = (E \cup \{i, f\}, \Sigma, \delta', \{i\}, \{f\})$, em que $i, f \notin E$, $i \neq f$ e:

- $\delta'(e, a) = \delta(e, a)$ para todo $(e, a) \in E \times \Sigma \cup \{\lambda\}$;
- $\delta'(i, \lambda) = I$;
- $\delta'(i, a) = \emptyset$ para $a \in \Sigma$;
- $\delta'(e, \lambda) = \delta(e, \lambda) \cup \{f\}$ para $e \in F$;
- $\delta'(f, a) = \emptyset$ para $a \in \Sigma \cup \{\lambda\}$.

2.4 Linguagens Regulares: Propriedades

1. A diferença básica é que, no primeiro exemplo, são considerados três casos para v: v contendo apenas $\mathtt{a}$s, v contendo $\mathtt{a}$s e $\mathtt{b}$s, e v contendo apenas $\mathtt{b}$s; e, no segundo, é considerado apenas um caso: visto que $|uv| \leq k$, v contém apenas $\mathtt{a}$s.

2. Passo a passo:

$$\begin{aligned}
&\neg\exists k \in \mathbf{N}\forall z \in L[|z| \geq k \rightarrow \exists u, v, w(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1 \wedge \forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\neg\forall z \in L[|z| \geq k \rightarrow \exists u, v, w(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1 \wedge \forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L\neg[|z| \geq k \rightarrow \exists u, v, w(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1 \wedge \forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \neg\exists u, v, w(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1 \wedge \forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \forall u, v, w\neg(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1 \wedge \forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \forall u, v, w(\neg(z = uvw \wedge |uv| \leq k \wedge |v| \geq 1) \vee \neg\forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \forall u, v, w((z = uvw \wedge |uv| \leq k \wedge |v| \geq 1) \rightarrow \neg\forall i \in \mathbf{N} uv^iw \in L)] \\
&\quad \equiv \forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \forall u, v, w((z = uvw \wedge |uv| \leq k \wedge |v| \geq 1) \rightarrow \exists i \in \mathbf{N} uv^iw \notin L)].
\end{aligned}$$

Assim, a contrapositiva é: $\forall k \in \mathbf{N}\exists z \in L[|z| \geq k \wedge \forall u, v, w((z = uvw \wedge |uv| \leq k \wedge |v| \geq 1) \rightarrow \exists i \in \mathbf{N} uv^iw \notin L)] \rightarrow L$ não é regular.

3. a) Seja $L = \{\mathtt{0}^m\mathtt{1}^n \mid m < n\}$. Suponha que L seja regular. Sejam k a constante do LB e $z = \mathtt{0}^k\mathtt{1}^{k+1}$. Sejam, de acordo com o LB, u, v e w tais que $z = uvw$, $v \neq \lambda$, $|uv| \leq k$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, sendo $z = uvw$ e $|uv| \leq k$, v só contém $\mathtt{0}$s e, portanto, $uv^2w = \mathtt{0}^{k+|v|}\mathtt{1}^{k+1}$. E sendo $v \neq \lambda$, $uv^2w \notin L$. Isto contradiz o LB. Logo, L não é regular.

   b) Seja $L = \{\mathtt{0}^n\mathtt{1}^{2n} \mid n \geq 0\}$. Suponha que L seja regular. Sejam k a constante do LB e $z = \mathtt{0}^k\mathtt{1}^{2k}$. Sejam, de acordo com o LB, u, v e w tais que $z = uvw$, $v \neq \lambda$, $|uv| \leq k$

e $uv^iw \in L$ para todo $i \geq 0$. Mas, sendo $z = uvw$ e $|uv| \leq k$, v só contém $\mathtt{0}$s e, portanto, $uv^2w = \mathtt{0}^{k+|v|}\mathtt{1}^{2k}$. E sendo $v \neq \lambda$, $uv^2w \notin L$. Isto contradiz o LB. Logo, L não é regular.

c) Seja $L = \{\mathtt{0}^m\mathtt{1}^n\mathtt{0}^m \,|\, m, n \geq 0\}$. Suponha que L seja regular. Sejam k a constante do LB e $z = \mathtt{0}^k\mathtt{1}\mathtt{0}^k$. Sejam, de acordo com o LB, u, v e w tais que $z = uvw$, $v \neq \lambda$, $|uv| \leq k$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, sendo $z = uvw$ e $|uv| \leq k$, v só contém $\mathtt{0}$s e, portanto, $uv^2w = \mathtt{0}^{k+|v|}\mathtt{1}\mathtt{0}^k$. E como $\mathtt{1}\mathtt{0}^k$ é sufixo de uv^2w, $\mathtt{0}^k\mathtt{1}$ deve ser prefixo para que uv^2w esteja em L. Mas, sendo $v \neq \lambda$, $\mathtt{0}^k\mathtt{1}$ não é prefixo de $\mathtt{0}^{k+|v|}\mathtt{1}\mathtt{0}^k$ e,portanto, $uv^2w \notin L$. Isto contradiz o LB. Logo, L não é regular.

d) Seja $L = \{x\mathtt{c}x \,|\, x \in \{\mathtt{a}, \mathtt{b}\}^*\}$. Suponha que L é regular, e seja k a constante do LB. Seja $z = \mathtt{a}^k\mathtt{c}\mathtt{a}^k$, e sejam u, v e w tais que $z = uvw$, $v \neq \lambda$ e $|uv| \leq k$. Segue-se, pelo LB, que $uv^iw \in L$ para todo $i \geq 0$. Mas, como $|uv| \leq k$, $uv^2w = \mathtt{a}^{k+|v|}\mathtt{c}\mathtt{a}^k$; como $v \neq \lambda$, $uv^2w \notin L$. Contradição. Portanto, L não é regular.

e) Seja $L = \{\mathtt{1}\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$. Suponha que L é regular, e seja k a constante do LB. Seja $z = \mathtt{1}\mathtt{0}^k\mathtt{1}^k$, e sejam u, v e w tais que $z = uvw$, $v \neq \lambda$ e $|uv| \leq k$. Segue-se, pelo LB, que $uv^iw \in L$ para todo $i \geq 0$. Há dois casos a considerar:

- v contém o $\mathtt{1}$ inicial. Como $v \neq \lambda$ e $|uv| \leq k$, tem-se que $uw = \mathtt{0}^{k-|v|+1}\mathtt{1}^k$; assim, $uv^0w \notin L$.
- v não contém o $\mathtt{1}$ inicial. Como $v \neq \lambda$ e $|uv| \leq k$, tem-se que $uw = \mathtt{1}\mathtt{0}^{k-|v|}\mathtt{1}^k$; assim, $uv^0w \notin L$.

Contradição. Portanto, L não é regular.

f) Seja $L = \{\mathtt{0}^{n^2} \,|\, n \geq 0\}$. Suponha que L seja regular. Sejam k a constante do LB e $z = \mathtt{0}^{k^2}$. Como dito no LB, existem u, v e w tais que $z = uvw$, $v \neq \lambda$, $|uv| \leq k$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, sendo $z = uvw$, $uv^2w = \mathtt{0}^{k^2+|v|}$. Sendo $v \neq \lambda$, $k^2 + |v| > k^2$; e sendo $|uv| \leq k$, $k^2 + |v| \leq k^2 + k < k^2 + 2k + 1 = (k+1)^2$, o que mostra que $k^2 + |v| < (k+1)^2$. Como $k^2 < k^2 + |v| < (k+1)^2$, $uv^2w \notin L$, o que contradiz o LB. Portanto, L não é regular.

4. a) Seja $L = \{\mathtt{0}, \mathtt{1}\}^* - \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$. Supondo que L seja regular, $\overline{L}$ também é, visto que as linguagens regulares são fechadas sob complementação. Mas $\overline{L} = \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$, e esta não é regular. Contradição! Logo, L não é regular.

b) Seja $L = \{\mathtt{0}^m\mathtt{1}^n \,|\, m < n\} \cup \{\mathtt{0}^m\mathtt{1}^n \,|\, m > n\}$. Suponha que L é regular. Então $\overline{L} \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^*$ também é regular, pois as linguagens regulares são fechadas sob complementação e sob interseção. Mas, $\overline{L} \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^* = \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$, que não é regular. Contradição! Logo, L não é regular.

c) Seja $L = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \,|$ o número de $\mathtt{0}$s em w é igual ao número de $\mathtt{1}$s$\}$. Então, como as linguagens regulares são fechadas sob interseção, $L \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^*$ deve ser regular. Mas, $L \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^* = \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$, linguagem não regular. Contradição! Logo, L não é regular.

d) Seja $L = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \,|$ o número de $\mathtt{0}$s em w é igual ao número de $\mathtt{1}$s$\} - \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$. Se L fosse regular, então $L \cap \{\mathtt{1}\}^*\{\mathtt{0}\}^*$ também seria, pois as linguagens regulares são fechadas sob interseção. Mas, $L \cap \{\mathtt{1}\}^*\{\mathtt{0}\}^* = \{\mathtt{1}^n\mathtt{0}^n \,|\, n \geq 0\}$, que não é regular. Portanto, L não é regular.

5. a) A linguagem pode ser expressa assim: $L' = L \cap \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*\{\mathtt{a}\}\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$. Como L' é a interseção de duas linguagens regulares, L' é regular.

b) A linguagem pode ser expressa assim: $L' = L \cup \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*\{\mathtt{a}\}\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$. Como L' é a união de duas linguagens regulares, L' é regular.

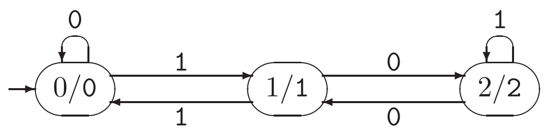


Figura 2.9: Máquina de Moore para o resto da divisão por 3.

c) A linguagem pode ser expressa assim:

$$L' = [L \cap \overline{\{\texttt{a},\texttt{b},\texttt{c}\}^*\{\texttt{a}\}\{\texttt{a},\texttt{b},\texttt{c}\}^*}] \cup [\overline{L} \cap \{\texttt{a},\texttt{b},\texttt{c}\}^*\{\texttt{a}\}\{\texttt{a},\texttt{b},\texttt{c}\}^*].$$

Como as linguagens regulares são fechadas sob complementação, interseção e união, segue-se que L' é regular.

d) A linguagem pode ser expressa assim: $L' = \overline{L} \cap \overline{\{\texttt{a},\texttt{b},\texttt{c}\}^*\{\texttt{a}\}\{\texttt{a},\texttt{b},\texttt{c}\}^*}$. Como as linguagens regulares são fechadas sob complementação e interseção, segue-se que L' é regular.

6. O AFD é $(E, \{\texttt{0},\texttt{1}\}, \delta, i, F)$, em que:

   - $E = \{\{0\},\{1,4\},\{3\},\{2,5\}\} \times \{000, 001, 010, 011, 100, 101, 110, 111\}$;
   - $\delta([X,x],d) = [\delta_1(X,d), \delta_2(x,d)]$ para:
     - $X \in \{\{0\},\{1,4\},\{3\},\{2,5\}\}$,
     - $x \in \{000, 001, 010, 011, 100, 101, 110, 111\}$,
     - $d \in \{\texttt{0},\texttt{1}\}$,

     sendo que δ_1 e δ_2 são as funções de transição dos AFDs dados;
   - $i = [\{0\}, 000]$;
   - $F = \{\{0\}\} \times \{000, 001, 010, 011\}$.

7. a) $\{\texttt{0}^m\texttt{1}^n \,|\, m < n\} \cup \{\texttt{0}^m\texttt{1}^n \,|\, m > n\} \cup \{\texttt{0}^n\texttt{1}^n \,|\, n \geq 0\} = \{\texttt{0}\}^*\{\texttt{1}\}^*$.

   b) O complemento de $\{\texttt{0},\texttt{1}\}^* - \{\texttt{01},\texttt{10}\}^*$ é $\{\texttt{01},\texttt{10}\}^*$.

   c) $\{\texttt{0}^n\texttt{1}^n \,|\, 0 \leq n \leq 10^{1000}\}$ é finita.

8. O conjunto das palavras sobre Σ_2 que têm como sufixo alguma palavra de L é $\Sigma_2^*(L \cap \Sigma_2^*)$. Esta é regular, visto que as linguagens regulares são fechadas sob interseção e concatenação.

2.5 Máquinas de Mealy e de Moore

1. A Figura 2.9 apresenta o diagrama de estados para uma máquina de Moore que determina o resto da divisão por 3 de um número na representação binária.

2. A Figura 2.10 apresenta o diagrama de estados para uma máquina de Moore que determina a quantidade de 1s presentes nos últimos 3 dígitos de palavras sobre $\{\texttt{0},\texttt{1}\}$.

3. A Figura 2.11 apresenta o diagrama de estados para uma máquina de Mealy que determina o quociente da divisão por 3 de um número na representação binária.

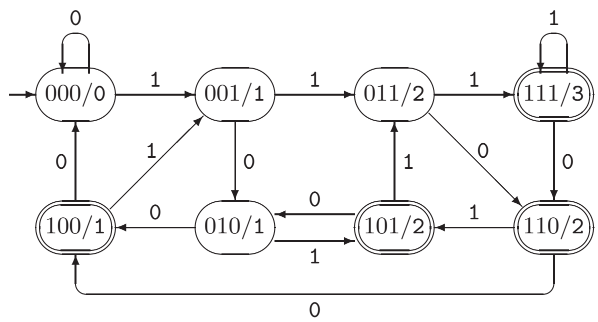


Figura 2.10: Máquina de Moore para número de 1s nos últimos 3.

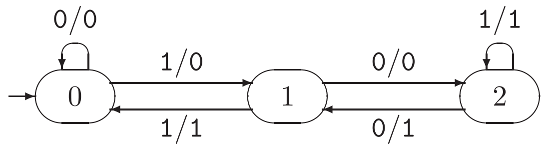


Figura 2.11: Máquina de Moore para o quociente da divisão por 3.

4. Máquina de Mealy para soma em binário:

$$(\{v_0, v_1\}, \{[0,0], [0,1], [1,0], [1,1]\}, \{0,1\}, \delta, \sigma, v_0)$$

onde δ e σ estão dados na tabela:

δ/σ	$[0,0]$	$[0,1]$	$[1,0]$	$[1,1]$
v_0	$v_0/0$	$v_0/1$	$v_0/1$	$v_1/0$
v_1	$v_0/1$	$v_1/0$	$v_1/0$	$v_1/1$

Evidentemente, o estado em que a máquina pára é importante. Se for v_1, existe um dígito 1 a acrescentar à saída.

5. A Figura 2.12 apresenta o diagrama de estados para uma máquina de Mealy equivalente à máquina de Moore referida.

6. A Figura 2.13 apresenta o diagrama de estados para uma máquina de Moore equivalente à máquina de Mealy refereida.

2.6 Expressões Regulares

1. a) $0 + (0+1)0 + 11$.

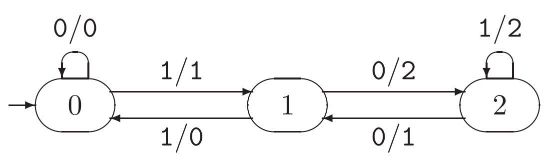


Figura 2.12: Máquina de Mealy equivalente a de Moore.

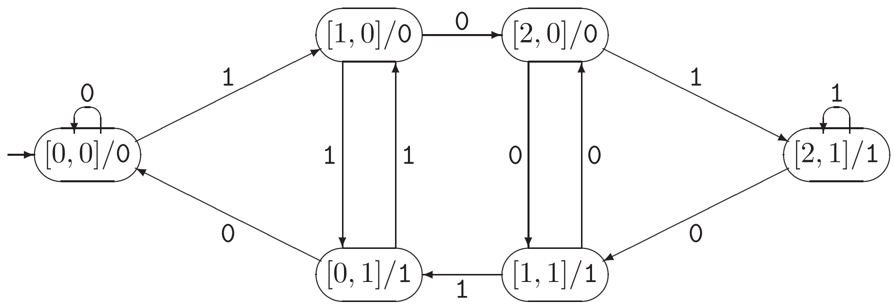


Figura 2.13: Máquina de Mealy equivalente a máquina de Moore.

b) $(00 + 111^*)^*0101(0 + 1)$.

2. a) Conjunto das palavras iniciadas com 0 e terminadas com 1.
   b) Conjunto das palavras com pelo menos um símbolo.
   c) Conjunto das palavras com pelo menos três símbolos, em que o antepenúltimo é 1.
   d) Conjunto das palavras sem ocorrências de 11.

3. a) $(a + b)^3(a + b)^*$.
   b) $a(a + b)((a + b)(a + b))^*$.
   c) $b^*(ab^*ab^*)^*$.
   d) $(a + b)^*bb(a + b)^*$.
   e) $(a + ba)^*bb(ab + a)^*$.
   f) $a^*(\lambda + b)a^*(\lambda + b)a^*$.
   g) $(c^*(a + b)c^*(a + b))^*c^*$.
   h) $(a + b + c)^*(a + b)(\lambda + c)$.

4. a)
$$\begin{aligned} \emptyset^* + \lambda^* &= \lambda + \lambda* && (13)\\ &= \lambda + \lambda && (14)\\ &= \lambda. && (3)\end{aligned}$$

   b) $0^* + 1^* + 0^*1^* + (0 + 1)^* = (0 + 1)^*$, visto que $r + s = s$ se $L(r) \subseteq L(s)$. No presente caso, $r = 0^* + 1^* + 0^*1^*$ e $s = (0 + 1)^*$.

   c) $(0+00)^*(1+01) = 0^*(000^*)^*(1+01)$, por (9). Esta, por sua vez, é igual a $0^*(1+01)$, pois $0^*(000^*)^* = 0^*$, já que $rs = r$ se $\lambda \in L(s)$ e $L(s) \subseteq L(r)$; no caso, $r = 0^*$ e $s = (000^*)^*$. Continuando: $0^*(1 + 01) = 0^*(\lambda 1 + 01)$, por (4), e $0^*(\lambda 1 + 01) = 0^*(\lambda + 0)1$, por (6). Finalmente, $0^*(\lambda + 0)1 = 0^*1$, pelo motivo já visto: $rs = r$ se $\lambda \in L(s)$ e $L(s) \subseteq L(r)$; no caso, $r = 0^*$ e $s = \lambda + 0$. Logo, $(0 + 00)^*(1 + 01) = 0^*1$.

   d)
$$\begin{aligned} (0+01)^*(0^*1^*+1)^* &= (0+01)^*((0^*1^*)^*1^*)^* && (18)\\ &= (0+01)^*((0+1)^*1^*)^* && (18)\\ &= (0+01)^*((0+1)^*)^* && (20)\\ &= (0+01)^*(0+1)^* && (11)\\ &= (0+1)^*. \end{aligned}$$

   Esta última se deve ao fato de que $rs = s$ se $\lambda \in L(r)$ e $L(r) \subseteq L(s)$; no caso, $r = (0 + 01)^*$ e $s = (0 + 1)^*$.

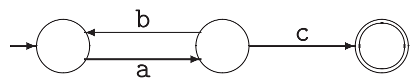


Figura 2.14: AFD para $(ab)^*ac$.

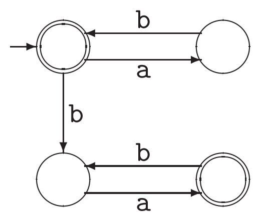


Figura 2.15: AFD para $(ab)^*(ba)^*$.

e) $((00 + 01 + 10 + 11)^*(0 + 1))^* = (0 + 1)^*$, baseado no fato de que $(rs)^* = s^*$, se $L(r) \subseteq L(s^*)$. No presente caso, $r = (00 + 01 + 10 + 11)^*$ e $s = (0 + 1)^*$.

f) $(0+1)^*0(0+1)^*1(0+1)^* = 1^*0(0+1)^*1(0+1)^*$, pois $(r+s)^*r(r+s)^* = s^*r(r+s)^*$ para quaisquer ERs r e s; para o presente caso, $r = 0$ e $s = 1$. Prosseguindo: $1^*0(0 + 1)^*1(0 + 1)^* = 1^*00^*1(0 + 1)^*$ pois, de forma similar, $(r + s)^*s(r + s)^* = r^*s(r + s)^*$. Portanto, $(0 + 1)^*0(0 + 1)^*1(0 + 1)^* = 1^*00^*1(0 + 1)^*$.

5. a) O diagrama de estados está mostrado na Figura 2.14.

   b) O diagrama de estados está mostrado na Figura 2.15.

   c) O diagrama de estados está mostrado na Figura 2.16.

   d) O diagrama de estados está mostrado na Figura 2.17.

   e) O diagrama de estados está mostrado na Figura 2.18.

6. As 16 possibilidades:

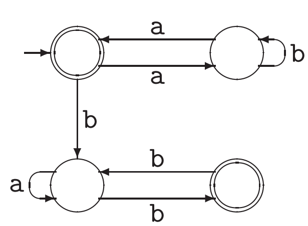


Figura 2.16: AFD para $(ab^*a)^*(ba^*b)^*$.

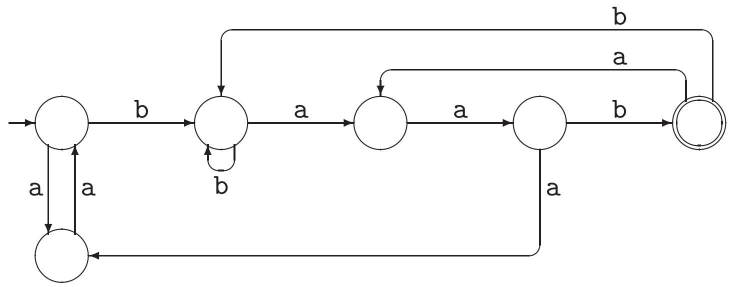


Figura 2.17: AFD para $(aa + b)^*baab$.

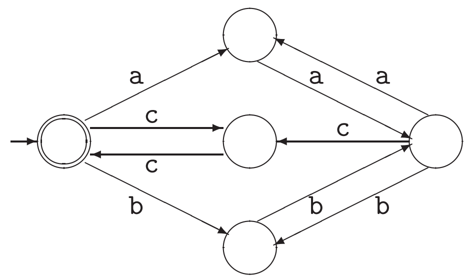


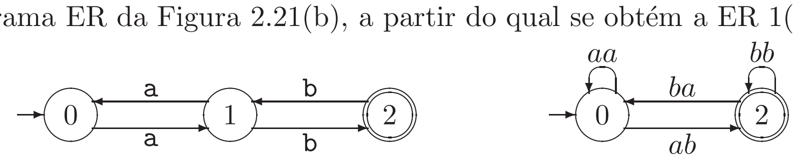
Figura 2.18: AFD para $((aa + bb)^*cc)^*$.

nenhum $\emptyset$:	$q^*r(s + tq^*r)^*$	$r = s = \emptyset$:	$\emptyset$
$q = \emptyset$:	$r(s + tr)^*$	$r = t = \emptyset$:	$\emptyset$
$r = \emptyset$:	$\emptyset$	$s = t = \emptyset$:	q^*r
$s = \emptyset$:	$q^*r(tq^*r)^*$	$q = r = s = \emptyset$:	$\emptyset$
$t = \emptyset$:	q^*rs^*	$q = r = t = \emptyset$:	$\emptyset$
$q = r = \emptyset$:	$\emptyset$	$q = s = t = \emptyset$:	r
$q = s = \emptyset$:	$r(tr)^*$	$r = s = t = \emptyset$:	$\emptyset$
$q = t = \emptyset$:	rs^*	$q = r = s = t = \emptyset$:	$\emptyset$

7. Eliminando-se o estado 3, obtém-se o diagrama ER da Figura 2.19(a). Eliminando-se o estado 1 deste último, obtém-se o diagrama ER da Figura 2.19(b), de onde se extrai a solução: $(aa)^*ab(bb + ba(aa)^*ab)^*$.

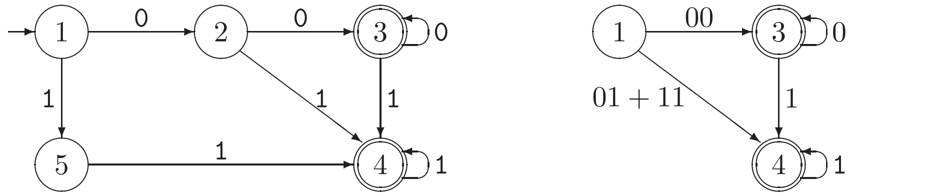
8. a) A Figura 2.20(a) mostra um AFD para a linguagem. Eliminando-se 2 e 5, obtém-se o diagrama ER da Figura 2.20(b). Em seguida, eliminando-se 4, obtém-se um diagrama ER a partir do qual se obtém a ER 000^*. Eliminando-se 3 em vez de 4, obtém-se um diagrama ER a partir do qual obtém a ER $(01 + 11 + 000^*1)1^*$. Assim, a ER é: $000^* + (01 + 11 + 000^*1)1^*$.

   b) A Figura 2.21(a) mostra um AFD para a linguagem. Eliminando-se 3, obtém-se o diagrama ER da Figura 2.21(b), a partir do qual se obtém a ER $1(1 + 00^*1)^*$.



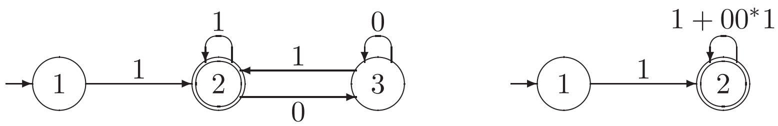
(a) Eliminando-se o estado 3. (b) Eliminando-se o estado 1.

Figura 2.19: Diagramas ER para a questão 7.



(a) O diagrama inicial. (b) Eliminando-se os estados 2 e 5.

Figura 2.20: Diagramas ER para a questão 8(a).



(a) O diagrama inicial. (b) Eliminando-se o estado 3.

Figura 2.21: Diagramas ER para a questão 8(b).

c) A Figura 2.22(a) mostra um AFD para a linguagem. Eliminando-se 2 e 3, obtém-se o diagrama ER da Figura 2.22(b), a partir do qual se obtém a ER $11^*00*1(1+00^*1)^*$.

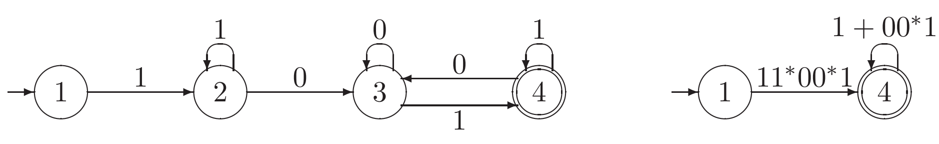
d) Considere a Figura 2.8 do livro. Eliminando-se *ip* e depois *pi* obtém-se o diagrama ER a partir do qual se obtém a ER $(00{+}11)^*(01{+}10)[00{+}11{+}(01{+}10)(00{+}11)^*(01{+}10)]^*$.

9. a) Seja um AFN $M = (E, \Sigma, \delta, I, F)$.

(a) Se $|I| > 1$, obtenha o AFNλ equivalente com um único estado inicial $M' = (E', \Sigma, \delta', \{i\}, F)$, em que:
- $E' = E \cup \{i\}$, em que $i \notin E$;
- $\delta'(i, \lambda) = I$;
- $\delta'(e, \lambda) = \emptyset$ para $e \in E$;
- $\delta'(i, a) = \emptyset$ para $(e, a) \in E \times \Sigma$;
- $\delta'(e, a) = \delta(e, a)$ para $(e, a) \in E \times \Sigma$;

senão M' é o próprio M (com $\delta'(e, \lambda) = \emptyset$ para todo estado).

(b) Seja, então, $M' = (E', \Sigma, \delta', I, F)$. Se $|F| > 1$, obtenha o AFNλ equivalente com um único estado final $M'' = (E' \cup \{f\}, \Sigma, \delta'', \{i\}, \{f\})$, em que:
- $f \notin E'$;
- $\delta''(e, \lambda) = \{f\}$ para todo $e \in F$;
- $\delta'(e, \lambda) = \emptyset$ para $e \in E' - F$;
- $\delta''(f, a) = \emptyset$ para todo $a \in \Sigma \cup \{\lambda\}$;
- $\delta''(e, a) = \delta'(e, a)$ para $(e, a) \in E' \times \Sigma$;



(a) O diagrama inicial. (b) Eliminando-se os estados 2 e 3.

Figura 2.22: Diagramas ER para a questão 8(c).

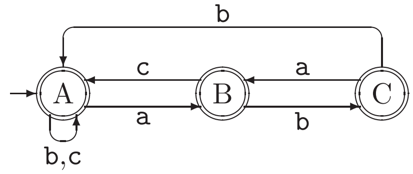


Figura 2.23: AFN obtido de GR, questão 2.

senão M'' é o próprio M' (com $\delta''(e, \lambda) = \emptyset$ para todo estado).

(c) Em seguida, basta executar o miolo do algoritmo da Figura 2.33, eliminando-se todos os estados, menos os estados inicial e final.

b) O processo indicado no item (a) pode ser adaptado facilmente para AFNλs.

10. Uma ER sem ocorrências de fecho de Kleene só pode denotar linguagens finitas. Isto será provado por indução no número de operadores da ER. ERs sem operadores são das formas $\emptyset$, λ e a para $a \in \Sigma$, as quais denotam linguagens finitas. Suponha, como hipótese de indução, que ERs com n ou menos operadores denotam linguagens finitas. Seja uma ER com $n+1$ operadores. Então a ER é da forma (rs) ou $(r+s)$, em que r e s são ERs com n ou menos operadores. Como, pela hipótese de indução, r e s denotam linguagens finitas, a ER, que denota no primeiro caso $L(r) \cup L(s)$, e no segundo $L(r)L(s)$, é também finita. Isto conclui a demonstração.

    Além do que foi provado acima, tem-se também que qualquer linguagem finita pode ser denotada por uma ER sem ocorrências de fecho de Kleene. A demonstração é simples: dada uma linguagem finita de n palavras $\{w_1, w_2, \ldots, w_n\}$, pode-se construir a seguinte ER, que denota a linguagem: $w_1 + w_2 + \cdots + w_n$.

2.7 Gramáticas Regulares

1. a) $(\{P\}, \{\mathtt{0}, \mathtt{1}\}, \{\}, P)$.

   b) $(\{P\}, \{\mathtt{0}, \mathtt{1}\}, \{P \rightarrow \lambda\}, P)$.

   c)
   $$\begin{aligned} P &\rightarrow \mathtt{0}A \mid \mathtt{1}A \mid \lambda \\ A &\rightarrow \mathtt{0}B \mid \mathtt{1}B \\ B &\rightarrow \mathtt{0}P \mid \mathtt{1}P \end{aligned}$$

   d)
   $$\begin{aligned} pp &\rightarrow \mathtt{0}\,ip \mid \mathtt{1}\,pi \mid \lambda \\ ip &\rightarrow \mathtt{0}\,pp \mid \mathtt{1}\,ii \\ pi &\rightarrow \mathtt{0}\,ii \mid \mathtt{1}\,pp \\ ii &\rightarrow \mathtt{0}\,pi \mid \mathtt{1}\,ip \end{aligned}$$

   e)
   $$\begin{aligned} P &\rightarrow \mathtt{0}A \mid \mathtt{1}P \mid \lambda \\ A &\rightarrow \mathtt{1}B \\ B &\rightarrow \mathtt{1}P \end{aligned}$$

   f)
   $$\begin{aligned} P &\rightarrow \mathtt{0}P \mid \mathtt{1}P \mid \mathtt{1}A \\ A &\rightarrow \mathtt{0}B \mid \mathtt{1}B \\ B &\rightarrow \mathtt{0} \mid \mathtt{1} \end{aligned}$$

2. A Figura 2.23 mostra um AFN para a linguagem.

3. A Figura 2.24 mostra um AFN para a linguagem.

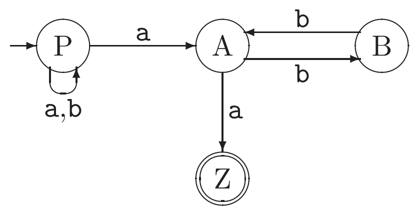


Figura 2.24: AFN obtido de GR, questão 3.

4. A gramática é:

$$
\begin{aligned}
pp &\rightarrow \mathtt{0}\,ip \,|\, \mathtt{1}\,pi \\
ip &\rightarrow \mathtt{0}\,pp \,|\, \mathtt{1}\,ii \\
pi &\rightarrow \mathtt{0}\,ii \,|\, \mathtt{1}\,pp \,|\, \lambda \\
ii &\rightarrow \mathtt{0}\,pi \,|\, \mathtt{1}\,ip
\end{aligned}
$$

5. Completando a prova do teorema em questão, será provado por indução sobre $|w|$ o lema

$$i \overset{*}{\Rightarrow} we \text{ se, e somente se, } e \in \hat{\delta}(\{i\}, w) \text{ para todo } e \in E \text{ e } w \in \Sigma^*.$$

Para $w = \lambda$ tem-se:

$$
\begin{aligned}
i \overset{*}{\Rightarrow} e &\leftrightarrow i = e \text{ pelas definições de } \overset{*}{\Rightarrow} \text{ e } R \\
&\leftrightarrow e \in \hat{\delta}(\{i\}, \lambda) \text{ pela definição de } \hat{\delta}
\end{aligned}
$$

Suponha que o resultado vale para um $x \in \Sigma^*$ arbitrário. Tem-se, então, para $w = xa$, onde $a \in \Sigma$:

$$
\begin{aligned}
i \overset{*}{\Rightarrow} xae &\leftrightarrow i \overset{*}{\Rightarrow} xe' \text{ e } e' \rightarrow ae \in R \text{ pelas definições de } \overset{*}{\Rightarrow} \text{ e } R \\
&\leftrightarrow e' \in \hat{\delta}(\{i\}, x) \text{ e } e' \rightarrow ae \in R \text{ pela hipótese de indução} \\
&\leftrightarrow e' \in \hat{\delta}(\{i\}, x) \text{ e } e \in \delta(e', a) \text{ pela definição de } R \\
&\leftrightarrow e \in \hat{\delta}(\{i\}, xa) \text{ pela definição de } \hat{\delta}
\end{aligned}
$$

6. Suponha que $x = a_1 a_2 \ldots a_n$, sendo $n \geq 2$. Então a regra $X \rightarrow xY$ pode ser substituída pelas regras:

$$
\begin{aligned}
X &\rightarrow a_1 R_1 \\
R_1 &\rightarrow a_2 R_2 \\
&\vdots \\
R_{n-1} &\rightarrow a_n Y
\end{aligned}
$$

em que R_1, R_2,..., R_n são variáveis novas, sem ocorrências em qualquer outra regra da gramática. De forma similar, a regra $X \rightarrow x$ pode ser substituída pelas mesmas regras, exceto a última, que fica sendo $R_{n-1} \rightarrow a_n$. Evidentemente, a linguagem gerada não se altera.

7. As construções mencionadas a seguir são todas possíveis, como mostrado no livro.

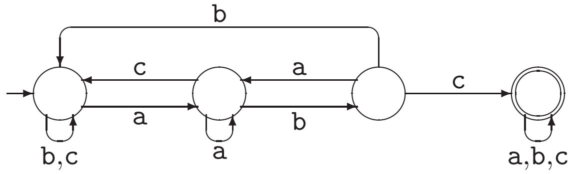


Figura 2.25: AFD da questão 1(a).

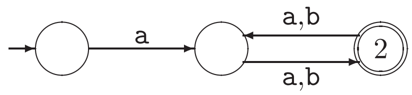


Figura 2.26: AFD da questão 1(b).

a) São necessárias, no mínimo, duas regras, uma com recursão e outra para por fim ao processo de recursão.

b) Sim. Pode-se, por exemplo, construir um AF que reconhece a linguagem denotada pela ER e, a partir dele, construir a GR.

c) Sim. Pode-se, por exemplo, a partir de um AFD (em formato de árvore ou não) que reconhece a linguagem, construir a GR.

d) Sim. Por exemplo: 1. da GR G obtém-se um AF M que reconhece $L(G)$; 2. a partir de M, obtém-se um AFD M' equivalente; 3. a partir de M' obtém-se uma GR G' que gera $L(M')$. G' é equivalente a G e tem a característica desejada.

8. Assim como no exercício anterior, os passos mencionados a seguir são todos possíveis, como mostrado no livro.

   a) Para verificar se $L(G_1) = \emptyset$ basta fazer:
      1. $M \leftarrow$ AF que aceita $L(G_1)$;
      2. se algum estado final de M é alcançável a partir do estado inicial, $L(G_1) \neq \emptyset$; caso contrário, $L(G_1) = \emptyset$.

   b) Para verificar se $L(G_1) \cap L(G_2) = \emptyset$ basta fazer:
      1. $M_1 \leftarrow$ AF que aceita $L(G_1)$; $M_1' \leftarrow$ AFD que aceita $L(M_1)$;
      2. $M_2 \leftarrow$ AF que aceita $L(G_2)$; $M_2' \leftarrow$ AFD que aceita $L(M_2)$;
      3. $M_3 \leftarrow$ AFD que aceita $L(M_1') \cap L(M_2')$ (produto);
      4. se algum estado final de M_3 é alcançável a partir do estado inicial, $L(G_1) \cap L(G_2) \neq \emptyset$; caso contrário, $L(G_1) \cap L(G_2) = \emptyset$.

2.8 Liguagens Regulares: Conclusão

Não há exercícios para esta seção.

2.9 Exercícios

1. a) O diagrama de estados está mostrado na Figura 2.25.
   b) O diagrama de estados está mostrado na Figura 2.26.
   c) O diagrama de estados está mostrado na Figura 2.27.
   d) O diagrama de estados está mostrado na Figura 2.28.

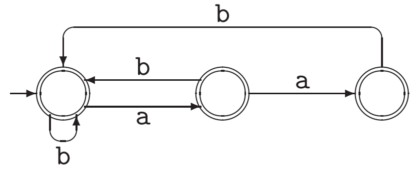


Figura 2.27: AFD da questão 1(c).

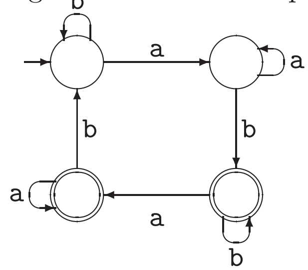


Figura 2.28: AFD da questão 1(d).

e) O diagrama de estados está mostrado na Figura 2.29.

f) O diagrama de estados está mostrado na Figura 2.30.

g) O diagrama de estados está mostrado na Figura 2.31.

h) Na Figura 2.32 está mostrado o diagrama de estados de um AFN com dois estados iniciais: 1 e 2. A partir do estado 1, é lido um prefixo x terminado em `b`, e a partir do estado 2 tal prefixo é λ (que também não termina em `a`). Um AFD equivalente está mostrado na Figura 2.33.

i) O diagrama de estados está mostrado na Figura 2.34.

j) O diagrama de estados está mostrado na Figura 2.35.

2. a) O diagrama de estados está mostrado na Figura 2.36.

b) O diagrama de estados está mostrado na Figura 2.37.

c) Para este exercício estou supondo que:

- toda palavra da linguagem deve ter no mínimo 4 símbolos;
- nos últimos 4 símbolos, nem sempre o número de `0`s é par, ou seja, ele pode ser par ou ímpar;
- entre o último `1` antes dos 4 últimos símbolos, se existir, e o primeiro `1` dos 4 últimos símbolos, se existir, deve haver um número par de símbolos.

A solução está mostrada na Figura 2.38.

d) A Figura 2.39 mostra um AFN para a linguagem.

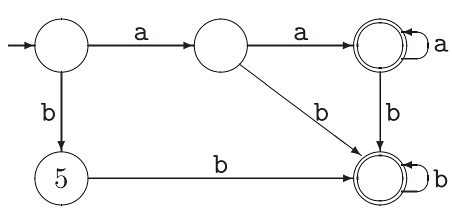


Figura 2.29: AFD da questão 1(e).

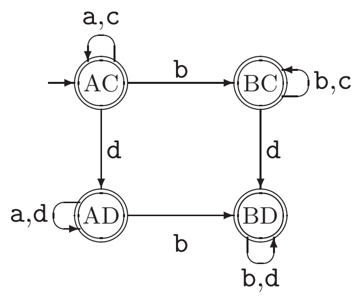


Figura 2.30: AFD da questão 1(f).

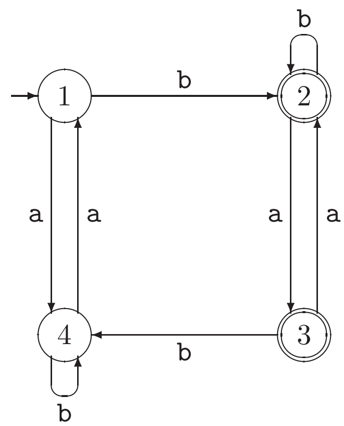


Figura 2.31: AFD da questão 1(g).

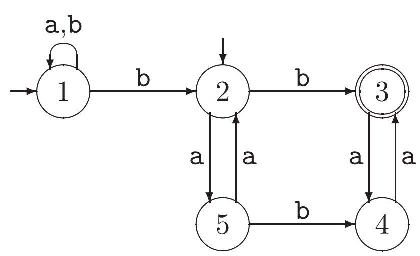


Figura 2.32: AFN da questão 1(h).

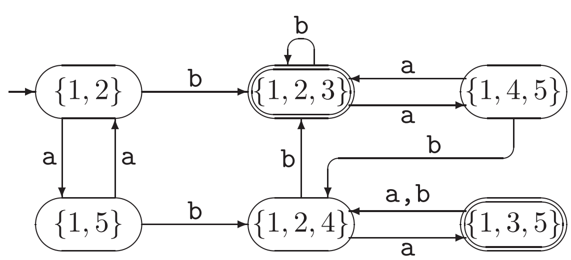


Figura 2.33: AFD da questão 1(h).

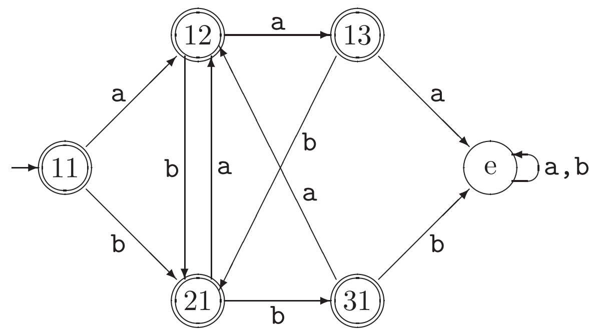


Figura 2.34: AFD da questão 1(i).

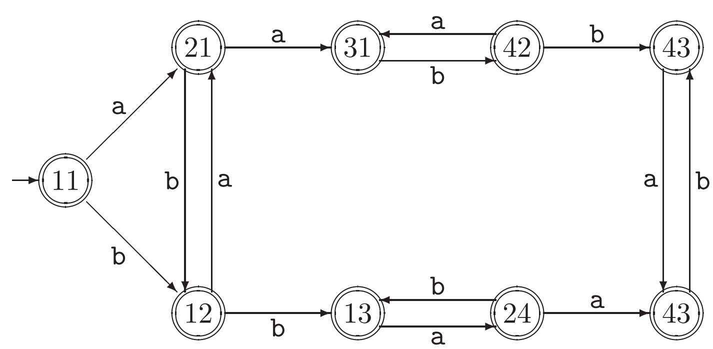


Figura 2.35: AFD da questão 1(j).

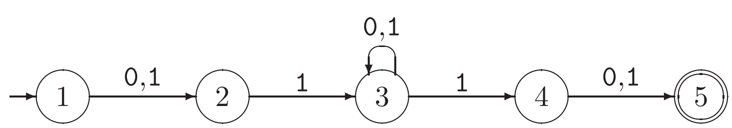


Figura 2.36: AFN da questão 2(a).

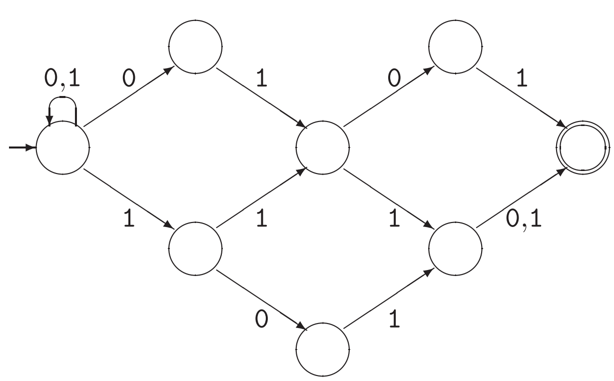


Figura 2.37: AFN da questão 2(b).

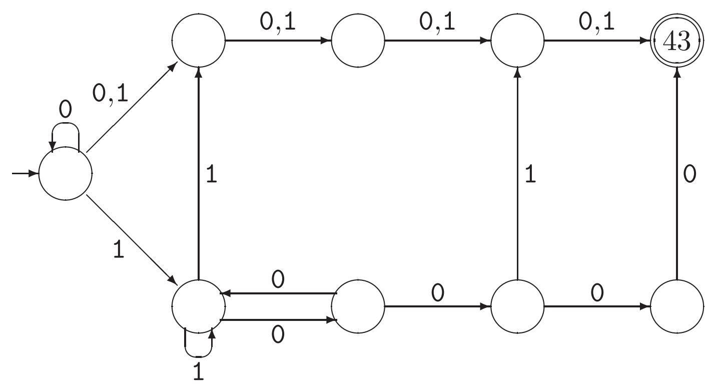


Figura 2.38: AFN da questão 2(c).

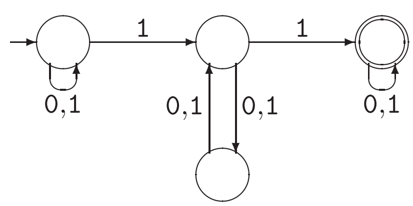


Figura 2.39: AFN da questão 2(d).

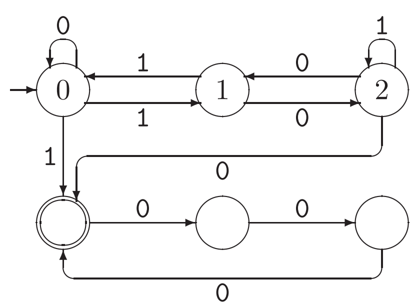


Figura 2.40: AFN da questão 2(e).

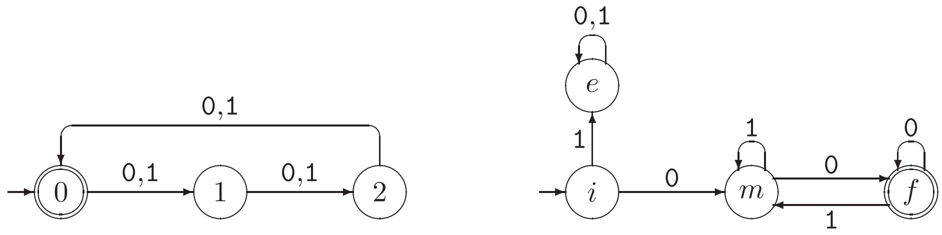


Figura 2.41: AFDs para L_1 e L_2, questão 3.

e) A Figura 2.40 mostra um AFN para a linguagem.

3. A Figura 2.41 mostra AFDs para L_1 e L_2. O produto dos dois resulta no AFD

$$(E, \{\mathtt{0}, \mathtt{1}\}, \delta, [0, i], \{[0, f]\})$$

, em que E e δ constam dos estados e transições que aparecem na tabela:

δ	0	1
$[0, i]$	$[1, m]$	$[1, e]$
$[0, m]$	$[1, f]$	$[1, m]$
$[1, m]$	$[2, f]$	$[2, m]$
$[2, m]$	$[0, f]$	$[0, m]$
$[0, f]$	$[1, f]$	$[1, m]$
$[1, f]$	$[2, f]$	$[2, m]$
$[2, f]$	$[0, f]$	$[0, m]$
$[0, e]$	$[1, e]$	$[1, e]$
$[1, e]$	$[2, e]$	$[2, e]$
$[2, e]$	$[0, e]$	$[0, e]$

Minimizando-se esse AFD, obtém-se um equivalente de 6 estados. São equivalentes:

- $[0, e]$, $[1, e]$ e $[2, e]$;
- $[1, m]$ e $[1, f]$,
- $[2, m]$ e $[2, f]$.

4. O conjunto de todas as mensagens já enviadas até este momento é um conjunto finito. Logo, é regular.

5. Sejam $M = (E, \Sigma, \delta, I, F)$ e $M' = (E', \Sigma, \delta', I', F')$. Observe que, pelo enunciado, $E' \subseteq E$, $I' \subseteq I$ e $F' \subseteq F$, e também que (vendo uma função como um conjunto) $\delta' \subseteq \delta$, sendo que δ' contém todas as transições que não envolvam os estados retirados.

   a) 1. $L(M') \subseteq L(M)$
   Seja $w \in \Sigma$. Como $I' \subseteq I$ e $\delta' \subseteq \delta$, para todo $i \in I'$ $\cup_{i \in I'} \hat{\delta}'(i, w) = \cup_{i \in I'} \hat{\delta}(i, w)$. Portanto, toda palavra reconhecida por M' é reconhecida por M.
   2. $L(M) \subseteq L(M')$
   Seja $w \in \Sigma$. Suponha que $w \in L(M)$. Então para qualquer estado $e \in E$ visitado no reconhecimento de w, existem x e y tais que $w = xy$, $e \in \hat{\delta}(i, x)$ para algum $i \in I$ e $f \in \hat{\delta}(e, y)$ para algum $f \in F$. Ou seja, e é alcançável a partir do estado

inicial i e a partir dele é alcançado o estado final f! Logo, todo estado e visitado no reconhecimento de w por M encontra-se em E', juntamente com todas as transições entre eles; com isso, w é reconhecida por M'.

b) $(\leftarrow)$

Suponha que o diagrama de estados de M' não tenha ciclos. Suponha que $L(M')$ seja infinita. Seja k o número de estados de M'. Sendo $L(M')$ infinita, tem palavras de tamanho maior ou igual a k. Mas, para reconhecer tais palavras, seria necessário passar por mais de k estados! Logo, $L(M) = L(M')$ não pode ser infinita. Portanto, se o diagrama de estados de M' não tem ciclos, $L(M)$ é finita.

$(\leftarrow)$

Suponha que o diagrama de estados de M' tenha ciclos. Seja e um estado de M' que faça parte de algum ciclo e suponha, então, que para um certo $v \in \Sigma^*$, $e \in \hat{\delta}'(e, v)$ (observe que $v \neq \lambda$, visto que AFN não tem transições λ). Sejam $i \in I'$ e $x \in \Sigma^*$ tais que $e \in \hat{\delta}'(i, x)$ (se não existissem tais i e x, e teria sido eliminado) e sejam $f \in F'$ e $y \in \Sigma^*$ tais que $f \in \hat{\delta}'(f, y)$ (se não existissem tais f e y, e teria sido eliminado). Então, $xv^iy \in L(M')$ para todo $i \geq 0$! Portanto, $L(M) = L(M')$ é infinita. Assim, se $L(M)$ é finita, o diagrama de estados de M' não tem ciclos.

6. a) $L(M_1) = \Sigma^*$.

   b) $L(M_2) = \{\lambda\}$.

7. a) Suponha que T_L seja uma progressão geométrica, e sejam n e r tais que $T_L = \{n, n + r, n + 2r, \ldots\}$. L nada mais é do que a linguagem denotada por $\mathtt{a}^n(\mathtt{a}^r)^*$. Como L pode ser denotada por uma expressão regular, L é regular.

   b) Seja $(E, \{\mathtt{a}\}, \delta, i, \{f_1, f_2, \ldots, f_k\})$ um AFD mínimo que reconheça L, e suponha que ele tenha k estados, $k > 0$. (Se $k = 0$, $L = \emptyset$.) Seja o conjunto $L_j = \{w \in \{\mathtt{a}\}^* \mid \hat{\delta}(i, w) = f_j\}$. Segue-se, por definição, que $L = \cup_{j=1}^{k} L_j$. Assim, basta mostrar que os conjuntos T_{L_j} são progressões aritméticas. Sejam n_j, para $1 \leq j \leq k$, os menores números naturais tais que $a^{n_j} \in L_j$. Como o AFD é mínimo, n_j existe para todo j. Como o autômato é determinístico e o alfabeto tem apenas um símbolo, para cada estado final f_j existe no máximo um ciclo que inicia e termina no mesmo; se não existir ciclo nenhum, tome $r_j = 0$, e se existir um ciclo, tome r_j igual ao número tal que $\hat{\delta}(f_j, \mathtt{a}^{r_j}) = f_j$. Segue-se que $L_j = \{a^{n_j + qr_j} \mid q \geq 0\}$ e, portanto, T_{L_j} é progressão aritmética.

8. a) Sim. Basta tomar $R_1 = \emptyset$ e $R_2 = \Sigma^*$.

   b) Não. Toda linguagem sobre Σ é subconjunto de Σ^*.

   c) Sim. Toda linguagem sobre Σ é subconjunto de Σ^*.

   d) Sim. Unindo-se duas linguagens não regulares de alfabetos disjuntos, obtém-se uma linguagem não regular. Unindo-se uma linguagem não regular L com a linguagem não regular $\overline{L}$, obtém-se Σ^*.

   e) Sim. A interseção de uma linguagem não regular L com a linguagem não regular $\overline{L}$ é $\emptyset$. Se uma linguagem não regular L está contida em outra, a interseção é L.

   f) Não. O complemento de uma linguagem não regular L não pode ser regular, pois se fosse, o seu complemento, que é L, seria regular!

9. a) $L = \{\mathtt{0}^n\mathtt{1}^{n+10} \mid n \geq 0\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{0}^k\mathtt{1}^{k+10}$. Como $z \in L$ e $|z| > k$, sejam, de acordo com o LB, u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, se $|uv| \leq k$ segue-se que v só pode conter 0s, e se $|v| > 0$, v contém pelo menos um 0; segue-se, então que $uv^2w = \mathtt{0}^{k+|v|}\mathtt{1}^{k+10} \notin L$, contradizendo o LB. Conclui-se que L não é regular.

b) $L = \{\mathtt{0}^n y \,|\, y \in \{\mathtt{0}, \mathtt{1}\}^* \text{ e } |y| \leq n\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{0}^k\mathtt{1}^k$, uma palavra de L. Como $|z| > k$, sejam, de acordo com o LB, u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Mas, se $|uv| \leq k$ e $|v| > 0$, segue-se que v é constituída de pelo menos um 0, e $uv^0w = \mathtt{0}^{k-|v|}\mathtt{1}^k$, com $|v| > 0$. Assim, $uv^0w \notin L$, o que contradiz o LB. Logo, L não é regular.

c) $L = \{\mathtt{0}^m\mathtt{1}^n \,|\, m \neq n\}$.

Suponha que L seja regular. Então, pelas propriedades de fechamento, $\overline{L}$ é regular e, assim, $\overline{L} \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^*$ também é regular. Mas $\overline{L} \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^* = \{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$, que não é regular. Contradição. Logo, L não é regular.

d) $L = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^{m+n} \,|\, m, n > 0\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{a}^k\mathtt{b}\mathtt{c}^{k+1} \in L$. Já que $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v só tem as e, assim, $uv^0w = \mathtt{a}^{k-|v|}\mathtt{b}\mathtt{c}^{k+1}$. E se $|v| > 0$, $(k - |v|) + 1 < k + 1$ e, portanto, $\mathtt{a}^{k-|v|}\mathtt{b}\mathtt{c}^{k+1} \notin L$, o que contradiz o LB. Logo, L não é regular.

e) $L = \{\mathtt{a}^n\mathtt{b}^{n^2} \,|\, n \geq 0\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{a}^k\mathtt{b}^{k^2} \in L$. Já que $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v só tem as e, assim, $uv^2w = \mathtt{a}^{k+|v|}\mathtt{b}^{k^2}$. E se $|v| > 0$, $(k + |v|)^2 > k^2$. Portanto, $\mathtt{a}^{k+|v|}\mathtt{b}^{k^2} \notin L$, o que contradiz o LB. Logo, L não é regular.

f) $L = \{\mathtt{a}^{n^3} \,|\, n \geq 0\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{a}^{k^3} \in L$. Como $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Dado que $z = uvw$, $uv^2w = \mathtt{a}^{k^3+|v|}$. Sendo $|v| > 0$, $k^3 + |v| > k^3$. De $|uv| \leq k$, segue-se que $|v| \leq k$ e, logo, $k^3 + |v| \leq k^3 + k$; como $k^3 + k < k^3 + 3k^2 + 3k + 1 = (k + 1)^3$, tem-se que $k^3 + |v| < (k + 1)^3$. Assim, $k^3 < k^3 + |v| < (k + 1)^3$ e, portanto, $k^3 + |v|$ não é um cubo perfeito e $\mathtt{a}^{k^3+|v|} \notin L$, contradizendo o LB. Conclusão: L não é regular.

g) $L = \{\mathtt{a}^m\mathtt{b}^n \,|\, n \leq m \leq 2n\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{a}^k\mathtt{b}^k$. Como $z \in L$ e $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v só tem as e, assim, $uv^0w = \mathtt{a}^{k-|v|}\mathtt{b}^k$. E se $|v| > 0$, $k - |v| < k$. Portanto, $\mathtt{a}^{k-|v|}\mathtt{b}^k \notin L$, o que contradiz o LB. Logo, L não é regular.

h) $L = \{xx \,|\, x \in \{\mathtt{a}, \mathtt{b}\}^*\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{a}^k\mathtt{b}\mathtt{a}^k\mathtt{b}$. Como $z \in L$ e $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v é uma subpalavra do prefixo de k as e, assim, $uv^2w = \mathtt{a}^{k+|v|}\mathtt{b}\mathtt{a}^k\mathtt{b}$. Sendo $|v| > 0$, se uv^2w tiver um número par de símbolos, a primeira metade conterá apenas 0s e os dois 1s ficarão na segunda metade. Portanto, $uv^2w \notin L$, o que contradiz o LB. Logo, L não é regular.

i) $L = \{x\overline{x} \mid x \in \{\mathtt{0},\mathtt{1}\}^*\}$.

$L \cap \{\mathtt{0}\}^*\{\mathtt{1}\}^* = \{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}$. Como as linguagens regulares são fechadas sob interseção, se L fosse regular, $\{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}$ também seria.

j) $L = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid w \neq w^R\}$.

Se L fosse regular, pelas propriedades de fechamento, $\overline{L}$ também seria. Mas $\overline{L}$ é o conjunto dos palíndromos sobre $\{\mathtt{0},\mathtt{1}\}$, que não é regular, como provado a seguir usando-se o LB.

Suponha que $\overline{L}$ seja regular. Seja k a constante do LB, e seja $z = \mathtt{0}^k\mathtt{1}\mathtt{0}^k$. Como $z \in L$ e $|z| > k$, o LB diz que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v é uma subpalavra do prefixo de k 0s e, assim, $uv^2w = \mathtt{0}^{k+|v|}\mathtt{110}^k\mathtt{b}$. Sendo $|v| > 0$, a primeira metade de uv^2w (incluindo o símbolo do meio se $|uv^2w|$ for ímpar) conterá apenas 0s e 1 ficará na segunda metade. Portanto, $uv^2w \notin \overline{L}$, o que contradiz o LB. Logo, $\overline{L}$ não é regular.

k) $L = \{w \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}^* \mid$ o número de as, bs e cs, em w, é o mesmo$\}$.

Suponha que L é regular. Como $\{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^*$ também é regular e as linguagens regulares são fechadas sob interseção, $L \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^*$ deveria ser regular. Mas $L \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^* = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$, e esta última não é regular. Logo, L não é regular.

l) Seja $L = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid$ o número de 0s em w é um cubo perfeito$\}$.

Suponha que L é regular. Como $\{\mathtt{0}\}^*$ também é regular e as linguagens regulares são fechadas sob interseção, $L \cap \{\mathtt{0}\}^*$ deveria ser regular. Mas $L \cap \{\mathtt{0}\}^* = \{\mathtt{a}^{n^3} \mid n \geq 0\}$, e esta última não é regular (como já mostrado em questão anterior). Logo, L não é regular.

m) Seja $L = \{\mathtt{0}^m\mathtt{1}^n \mid mdc(m,n) = 1\}$.

Suponha que L é regular. Seja k a constante do lema do bombeamento e considere $z = \mathtt{0}^{(k+1)!+1}\mathtt{1}^{(k+1)!}$. Pelo referido lema, existem u, v e w tais que $z = uvw$, $|v| > 0$, $|uv| \leq k$ e $uv^iw \in L$ para todo $i \geq 0$. Mas se $z = uvw$, $|v| > 0$ e $|uv| \leq k$, $uv^{(k+1)!+2}w \notin L$, pois, sendo $|uv| \leq k$,

- o número de as de $uv^{(k+1)!+2}w$ é $(k+1)! + 1 + ((k+1)! + 2 - 1)|v| = [(k+1)! + 1](1 + |v|)$, que é divisível por $1 + |v|$;
- o número de bs continua sendo $(k+1)!$, que também é divisível por $1 + |v|$ (pois $2 \leq 1 + |v| \leq k + 1$, já que $|v| > 0$);

e com isso, o máximo divisor comum de $[(k+1)! + 1](1 + |v|)$ e $(k+1)!$ é no mínimo $1 + |v|$.

n) $L = \{\mathtt{a}^k\mathtt{b}^m\mathtt{c}^n \mid k \neq m$ ou $m \neq n\}$.

Suponha que L é regular. Como $\{\mathtt{a}\}^*\{\mathtt{b}\}^*$ também é regular e as linguagens regulares são fechadas sob interseção, $L \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*$ deveria ser regular. Mas $L \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^* = \{\mathtt{a}^k\mathtt{b}^m \mid k \neq m\}$, e esta última não é regular (como provado em questão anterior. Logo, L não é regular.

o) $\{\mathtt{0}^m\mathtt{1}^n\mathtt{0}^n \mid m,n > 0\}$.

Suponha que L seja regular. Seja k a constante do LB, e seja $z = \mathtt{01}^k\mathtt{0}^k$. Como $z \in L$ e $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$

e $uv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v é uma subpalavra do prefixo $\texttt{01}^k$ de z. Considera-se dois casos:

Caso 1. v começa com $\texttt{0}$. Então uv^0w começa com $\texttt{1}$. Logo, $uv^0w \notin L$.

Caso 2. v não começa com $\texttt{0}$. Então $uv^0w = \texttt{01}^{k-|v|}\texttt{0}^k$ e, como $|v| > 0$, $uv^0w \notin L$.

Portanto, em qualquer caso $uv^0w \notin L$, o que contradiz o LB. Logo, L não é regular.

10. **a)** É regular. ER que a denota: $([0,1] + [1,0])^*$.

    b) Não é regular. Seja L a linguagem em questão. Então $L \cap \{[0,1]\}^*\{[1,0]\}^* = \{[0,1]^n[1,0]^n \mid n \geq 0\}$. Como esta última não é regular e as linguagens regulares são fechadas sob interseção, L não é regular.

    c) É regular. A seguinte ER denota as palavras em que $b_1 = 0$: $([0,0]+[1,0][1,1]^*[0,1])^*$. Analogamente, tem-se uma ER para as com $b_1 = 1$: $([1,1] + [0,1][0,0]^*[1,0])^*$.

    d) Não é regular. Seja L a linguagem em questão. Então, assim como no item (b), $L \cap \{[0,1]\}^*\{[1,0]\}^* = \{[0,1]^n[1,0]^n \mid n \geq 0\}$. Como esta última não é regular e as linguagens regulares são fechadas sob interseção, L não é regular.

11. Seja L uma linguagem regular infinita. Pelo LB, dada uma palavra x de tamanho maior que a constante referida no lema, e tal palavra *existe*, pois L é infinita, ela pode ser dividida em três partes u, v e w, de forma que:

    - $uvw = x$,
    - $v \neq \lambda$ e
    - $uv^kw \in L$ para todo $k \geq 0$.

    Assim, seja uma dessas palavras $uvw = x$. Tome $m = |uw|$ e $n = |v|$ (note que $n > 0$). Pelo LB, $uv^kw \in L$ para todo $k \geq 0$, de onde se tira que $|uv^kw| = |uw| + k|v| = m + kn$ para todo $k \geq 0$. Isto mostra, como requerido, que existem $m, n \in \mathrm{N}$, $n > 0$, tais que para todo $k \geq 0$, L contém alguma palavra z (a palavra uv^kw) tal que $|z| = m + kn$

12. *Uma generalização do lema do bombeamento:* Seja L uma linguagem regular. Então existe uma constante $k > 0$ tal que para qualquer palavra $xz \in L$ com $|z| \geq k$ existem u, v e w que satisfazem as seguintes condições:

    - $z = uvw$;
    - $|uv| \leq k$;
    - $v \neq \lambda$; e
    - $xuv^iw \in L$ para todo $i \geq 0$.

    Uso do lema para provar que $L = \{\texttt{0}^m\texttt{1}^n\texttt{0}^n\} \mid m, n > 0$ não é regular:

    Suponha que L seja regular. Seja k a constante do lema, e seja a palavra $xz = \texttt{01}^k\texttt{0}^k$, onde $x = \texttt{0}$ e $z = \texttt{1}^k\texttt{0}^k$. Como $xz \in L$ e $|z| > k$, o LB afirma que existem u, v e w tais que $z = uvw$, $|uv| \leq k$, $|v| > 0$ e $xuv^iw \in L$ para todo $i \geq 0$. Se $z = uvw$ e $|uv| \leq k$, então v só tem $\texttt{1}$s e, assim, $xuv^2w = \texttt{01}^{k+|v|}\texttt{0}^k$. E se $|v| > 0$, $\texttt{01}^{k+|v|}\texttt{0}^k \notin L$, o que contradiz o lema. Logo, L não é regular.

13. *Variante do lema do bombeamento:* Seja L uma linguagem regular. Então existe uma constante $k > 0$ tal que para qualquer palavra $z \notin L$ com $|z| \geq k$ existem u, v e w que satisfazem as seguintes condições:

    - $z = uvw$;

- $|uv| \leq k$;
- $v \neq \lambda$; e
- $uv^iw \notin L$ para todo $i \geq 0$.

A demonstração é similar ao do lema original. Este lema, como o original, serve para provar, que uma linguagem não é regular. E, assim como quando se usa o lema original, a prova é feita por contradição. Segue um exemplo.

Seja $L = \{\mathtt{a}^m\mathtt{b}^n \mid m \neq n\}$. Suponha que L seja uma linguagem regular. Seja k a constante referida no lema, e seja $z = \mathtt{a}^k\mathtt{b}^k$. Como $z \notin L$ e $|z| > k$, o lema diz que que existem u, v e w de forma que as seguintes condições se verificam: $z = uvw$, $|uv| \leq k$, $v \neq \lambda$ e $uv^iw \notin L$ para todo $i \geq 0$. Neste caso, v só tem $\mathtt{a}$s, pois $z = uvw = \mathtt{a}^k\mathtt{b}^k$ e $|uv| \leq k$, e v tem pelo menos um $\mathtt{a}$, pois $v \neq \lambda$. Isto implica que $uv^2w = \mathtt{a}^{k+|v|}\mathtt{b}^k \in L$, o que contraria o lema. Conclui-se, assim, que L não é linguagem regular.

14. **a)** Dado um AFN $M = (E, \Sigma, \delta, I, F)$ para L, obtém-se um AFN M' para $\mathit{pref}(L)$: $M' = (E, \Sigma, \delta, I, F')$, em que

$$F' = \{e \in E \mid \hat{\delta}(\{e\}, w) \cap F \neq \emptyset \text{ para alguma } w \in \Sigma^*\}.$$

Em outras palavras: basta tornar estados finais todos os estados a partir dos quais se alcance algum estado final.

b) Dado um AFN $M = (E, \Sigma, \delta, I, F)$ para L, obtém-se um AFN M' para $\mathit{suf}(L)$: $M' = (E, \Sigma, \delta, I', F)$, em que

$$I' = \{e \in E \mid e \in \hat{\delta}(I, w) \text{ para alguma } w \in \Sigma^*\}.$$

Em outras palavras: basta tornar estados iniciais todos os estados alcancáveis a partir de algum estado inicial.

c) Dado um AFN $(E, \Sigma, \delta, I, F)$ que reconheça L, constrói-se um AFN que reconhece $\mathit{rev}(L)$, $(E, \Sigma, \delta', F, I)$ (observe a troca dos conjuntos de estados inciais e finais), sendo δ' tal que todas as transições são *invertidas*, ou seja, $\delta'(e, a) = \{e' \mid e \in \delta(e', a)\}$.

d) Pelo exercício anterior, $\mathit{rev}(L)$ é regular. Como as linguagens regulares são fechadas sob concatenação e $\mathit{crev}(L) = L\mathit{rev}(L)$, segue-se que $\mathit{crev}(L)$ é regular.

e) Seja um AFN $M = (E, \Sigma, \delta, I, F)$ tal que $L(M) = L$. A partir de M constrói-se o seguinte AFN que reconhece $\mathit{mpal}(L)$: $(E', \Sigma, \delta', I', F')$, onde:

- $E' = E \times E$;
- $I' = I \times F$;
- (veja a ilustração na Figura 2.42) para cada $(e_1, e_2) \in E'$ e $a \in \Sigma$:

$$\delta'([e_1, e_2], a) = \{[e'_1, e'_2] \mid e'_1 \in \delta(e_1, a) \text{ e } e_2 \in \delta(e'_2, a)\}; \text{ e}$$

- $F' = \{[e, e] \mid e \in E\}$.

f) Suponha que $L \neq \emptyset$ e seja $M = (E, \Sigma, \delta, I, F)$ um AFN para L *sem estados inúteis*, ou seja, em que para todo estado e existem x e y tais que $e \in \hat{\delta}(I, x)$ e $\hat{\delta}(\{e\}, y) \cap F \neq \emptyset$. (Qualquer estado que não satisfaça essas condições pode ser eliminado sem alterar a linguagem reconhecida.) Será mostrado como obter, a partir de M, um AFN-λ para $dc(L)$ com $2|E|^2$ estados $M' = (E', \Sigma, \delta', I', F')$. Nas explicações a seguir, considere M' simulando M, de forma que uma palavra M' está reconhecendo yx, como conseqüência de M reconhecer xy.

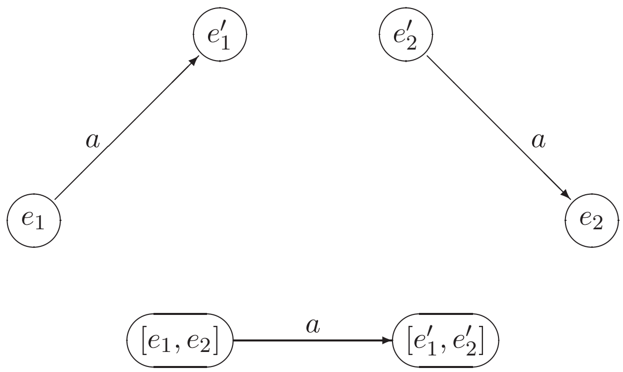


Figura 2.42: Ilustração para o exercício 14(e).

- $E' = E \times E \times \{p, s\}$.
  Seja uma palavra xy reconhecida por M. Então M' deve reconhecer yx. Um estado $[e_1, e_2, s]$ de M' modela o caso em que M, começando pelo estado e_1, já está no estado e_2 e leu parte do sufixo y (por isso, s). E um estado $[e_1, e_2, p]$ de M' modela o caso em que M, começando pelo estado e_1, já está no estado e_2 e leu parte do prefixo x (por isso, p).
- $I' = \{[e, e, s] \mid e \in E\}$.
  Quando M' começa no estado $[e, e, s]$, simula o caso em que M, começando pelo estado e, está prestes a começar a ler um sufixo y.
- $F' = \{[e, e, p] \mid e \in E\}$.
  Quando M' terminar no estado $[e, e, p]$, terá acabado de simular M lendo o prefixo x e terminando no estado e.
- Para cada estado $e \in E$ haverá duas cópias do AFN M:

  $\delta([e, e', p], a) = [e, \delta(e', a), p]$ para todo $e' \in E$ e todo $a \in \Sigma$;
  $\delta([e, e', s], a) = [e, \delta(e', a), s]$ para todo $e' \in E$ e todo $a \in \Sigma$;

  Uma cópia do AFN M é para reconhecer sufixos y (s) e a outra para prefixos x (p).
  A ligação do reconhecimento de um sufixo (s) com o reconhecimento de um prefixo é feita pelas λ-transições:

  $\delta([e, f, s], \lambda) = \{[e, i, p] \mid i \in I\}$ para todo $e \in E$ e $f \in F$.

g) Similar ao item (e). Seja um AFN $M = (E, \Sigma, \delta, I, F)$ tal que $L(M) = L$. A partir de M constrói-se o seguinte AFN que reconhece $pm(L)$: $(E', \Sigma, \delta', I', F')$, onde:

- $E' = E \times E$;
- $I' = I \times F$;
- para cada $(e_1, e_2) \in E'$ e $a \in \Sigma$:

  $$\delta'([e_1, e_2], a) = \{[e'_1, e'_2] \mid e'_1 \in \delta(e_1, a) \text{ e } e_2 \in \delta(e'_2, b) \text{ para algum } b \in \Sigma\}; \text{ e}$$

- $F' = \{[e, e] \mid e \in E\}$.

h) Sejam dois AFNs $M_1 = (E_1, \Sigma, \delta_1, I_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, I_2, F_2)$ tais que $L(M_1) = L_1$ e $L(M_2) = L_2$. (É sempre possível, e trivial, tornar os alfabetos idênticos sem alterar as linguagens.) Um AFN $M_3 = (E_3, \Sigma, \delta_3, I_3, F_3)$ que reconhece $\hat{\xi}(L_1, L_2)$ é tal que:

- $E_3 = E_1 \times E_2$;
- $I_3 = I_1 \times I_2$;
- para cada $(e_1, e_2) \in E_3$ e $a \in \Sigma$:

$$\delta_3([e_1, e_2], a) = \{[e_1', e_2] \mid e_1' \in \delta_1(e_1, a)\} \cup \{[e_1, e_2'] \mid e_2' \in \delta_2(e_2, a)\}; \text{ e}$$

- $F_3 = F_1 \times F_2$.

15. Seja um AFN $M = (E, \Sigma, \delta, i, F)$. Um AFD M' equivalente sem transições para o estado inicial: $M' = (E \cup \{i'\}, \Sigma, \delta', i', F')$, tal que $i' \notin E$, $\delta'(e, a) = \delta(e, a)$ para todo $e \in E$ e todo $a \in \Sigma$, $\delta'(i', a) = \delta(i, a)$ para todo $a \in \Sigma$, e $F' = F$ se $i \notin F$, e $F' = F \cup \{i'\}$ se $i \in F$.

16. **a)** Por indução sobre n. Primeiro, observe que se L_1 e L_2 são regulares, $L_1 \cup L_2$ é regular, pois as linguagens regulares são fechadas sob união. Seja $n \geq 2$ e n linguagens regulares L_1, L_2, $\ldots$, L_n. Suponha, como hipótese de indução, que $\bigcup_{2 \leq i \leq n} L_i$ seja regular. Dada essa hipótese e o fato de que as linguagens regulares são fechadas sob união, segue-se que

$$\bigcup_{2 \leq i \leq n+1} L_i = (\bigcup_{2 \leq i \leq n} L_i) \cup L_{n+1}$$

é regular.

b) Cada linguagem $\{\mathtt{a}^i\mathtt{b}^i\}$, para $i \geq 2$, é uma linguagem regular, mas

$$\bigcup_{i \geq 2} \{\mathtt{a}^i\mathtt{b}^i\} = \{\mathtt{a}^i\mathtt{b}^i \mid i \geq 2\}$$

não é.

17. $(\rightarrow)$ Suponha que $L \cup \{\lambda\}$ é regular. Como $(L \cup \{\lambda\}) - \{\lambda\} = L - \{\lambda\}$ e as linguagens regulares são fechadas sob diferença, $L - \{\lambda\}$ é regular.

$(\rightarrow)$ Suponha que $L - \{\lambda\}$ é regular. Como $(L - \{\lambda\}) \cup \{\lambda\} = L \cup \{\lambda\}$ e as linguagens regulares são fechadas sob união, $L \cup \{\lambda\}$ é regular.

18. Seja r uma expressão regular que denota L. Seja r' a expressão regular obtida de r substituindo-se cada símbolo do alfabeto, a, de r por $h(a)$. Mostra-se facilmente, por indução a partir da definição de expressão regular, que r' denota $h(L)$.

Agora, para provar o fechamento sob homorfismo inverso, seja um AFD para L, $M = (E, \Sigma, \delta, i, F)$. Seja o AFD $M' = (E, \Sigma, \delta', i, F)$ em que $\delta'(e, a) = \hat{\delta}(e, h(a))$ para todo $e \in E$ e $a \in \Sigma$. Pode-se provar, por indução sobre $|w|$, que $\delta'(e, w) = \hat{\delta}(e, h(w))$. Como os estados iniciais e finais dos dois AFDs são os mesmos, segue-se que $w \in L(M')$ se, e somente se, $h(w) \in L(M)$ e, portanto, $L(M') = h^{-1}(L)$.

19. Seja $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ e $h : \{\mathtt{a}^n\mathtt{b}\} \rightarrow \{\mathtt{1}\}$ tal que $h(\mathtt{a}) = \lambda$ e $h(\mathtt{b}) = \mathtt{1}$. Segue-que que $h(L) = \{\mathtt{1}\}^*$ e, assim, $h(L)$ é regular, embora L não seja.

20. Para cada $a \in \Sigma$, seja $er(a)$ uma expressão regular que denota $s(a)$ (que existe, pois $s(a)$ é regular). Seja r uma expressão regular que denota L. Seja r' a expressão regular obtida de r substituindo-se cada símbolo do alfabeto, a, de r por $er(a)$. Mostra-se facilmente, por indução a partir da definição de expressão regular, que r' denota $s(L)$.

21. Seja um AFD $M = (E, \Sigma, \delta, i, F)$ que reconheça L_1. Um AFD que reconhece L_1/L_2 é o AFD $M' = (E, \Sigma, \delta, i, F')$, cuja única diferença com relação a M_1 é o conjunto de estados finais, assim definido: $F' = \{e \in E \mid \exists y \in L_2 \hat{\delta}(e, y) \in F\}$.

22. **a)** Suponha que $L \cup F$ seja regular. $F - L$ também é regular, pois é finita. Logo, seu complemento, $\overline{F - L}$, também é regular. Como a interseção de duas linguagens regulares é regular, segue-se que $(L \cup F) \cap \overline{F - L}$ é regular. Mas, $(L \cup F) \cap \overline{F - L} = L$, que não é regular. Contradição. Portanto, $L \cup F$ não é regular.

 b) Suponha que $L - F$ seja regular. $F \cap L$ também é regular, pois é finita. Como a união de duas linguagens regulares é regular, segue-se que $(L - F) \cup (F \cap L)$ é regular. Mas, $(L - F) \cup (F \cap L) = L$, que não é regular. Contradição. Portanto, $L - F$ não é regular.

23. **a)** Não. Por exemplo, $L \cup \Sigma^* = \Sigma^*$ para uma linguagem não regular L sobre Σ.

 b) Não. Por exemplo, $L\Sigma^* = \Sigma^*$ para uma linguagem não regular L sobre Σ *que contenha* λ.

 c) Sim. Seja $subp(L_1) = \{w \mid w$ é uma subpalavra de $L_1\}$. Toda subpalavra de uma palavra x é o prefixo de um sufixo de x (e vice-versa). Assim, $subp(L_1) = pref(suf(L_1))$. Segue, pelos exercícios 14(a) e 14(b), que $subp(L_1)$ é regular.

24. **a)** ($\rightarrow$) Suponha que $L(M) = \emptyset$. Como M não reconhece nada, segue-se o resultado.

 ($\leftarrow$) Suponha que o AFD M não reconheça palavra de tamanho menor que n. Com o objetivo de se chegar a uma contradição, suponha que $L(M) \neq \emptyset$. Então a *menor* palavra aceita por M tem tamanho maior ou igual a n. Com isso, pelo lema do bombeamento, para uma para dessas, z, existem u, v e w tais que $z = uvw$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Em particular, $uw \in L$ e, como $|uw| < |z|$ (já que $|v| > 0$) e como z é uma *menor* palavra que M aceita, chega-se a uma contradição. Conclui-se que $L(M) = \emptyset$.

 b) ($\rightarrow$) Suponha que $L(M)$ é finita. Se M reconhecesse uma palavra z de tamanho maior ou igual a n, então, pelo lema do bombeamento, existiriam u, v e w tais que $z = uvw$, $|v| > 0$ e $uv^iw \in L$ para todo $i \geq 0$. Com isso, $L(M)$ seria infinita! Logo, M não reconhece palavra de tamanho maior ou igual a n.

 ($\leftarrow$) Suponha que $L(M)$ é infinita. Então M reconhece palavra a partir de qualquer tamanho e, assim, reconhece palavras de tamanho maior ou igual a $2n$. Seja, então uma palavra z, de tamanho *menor* possível, maior ou igual a $2n$. Pelo lema do bombeamento, existem u, v e w tais que $z = uvw$, $|v| > 0$, $|uv| \leq k$ e $uv^iw \in L$ para todo $i \geq 0$. Em particular, $uw \in L$. Como $|v| > 0$, essa última palavra é menor que z. Como ela não pode ser maior ou igual a $2n$ (pois z é a menor possível de tamanho maior ou igual a $2n$) e $|uv| \leq k$, conclui-se que $|uw|$ é maior ou igual a n e menor que $2n$.

25. Todos os passos dos PDs a seguir são corroborados por algoritmos desenvolvidos no capítulo.

 a) Um PD para o problema seria: Obtenha (AFNs e depois) AFDs M_1 para $L(r_1)$ e M_2 para $L(r_2)$. Em seguida, a partir de M_1 e M_2, obtenha AFDs M_3 para $L(M_1) - L(M_2) = L(M_1) \cap \overline{L(M_2)}$ e M_4 para $L(M_2) - L(M_1) = L(M_2) \cap \overline{L(M_1)}$. Finalmente, verifique se ambos os AFDs, M_3 e M_4, reconhecem alguma palavra de tamanho menor que o respectivo número de estados; se sim, $L(r_1) \neq L(r_2)$; caso contrário, $L(r_1) = L(r_2)$.

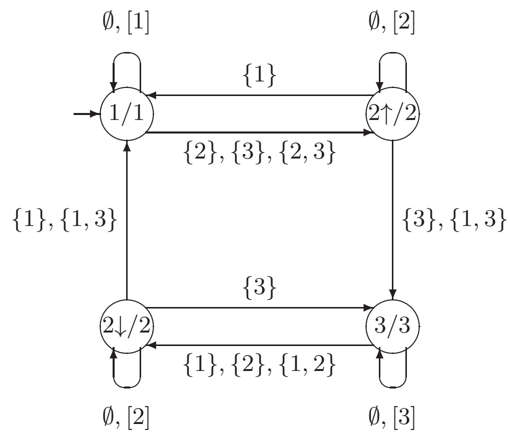
 b) Similar ao anterior: Obtenha AFDs M_1 para $L(r_1)$ e M_2 para $L(r_2)$. Em seguida, obtenha um AFD M_3 para $L(M_1) - L(M_2) = L(M_1) \cap \overline{L(M_2)}$. Finalmente, verifique se M_3 reconhece alguma palavra de tamanho menor que o número de estados; se sim, $L(r_1) \not\subseteq L(r_2)$; caso contrário, $L(r_1) \subseteq L(r_2)$.

c) Um PD para o problema seria: Obtenha um (AFN e depois) AFD M para $L(r_1)$. Em seguida, minimize M. Se o resultado for um AFD do tipo $(\{e\}, \Sigma, \delta, e, \{e\})$, em que $\delta(e, a) = e$ para todo $a \in \Sigma$, $L(r_1) = \Sigma^*$; caso contrário, $L(r_1) \neq \Sigma^*$.

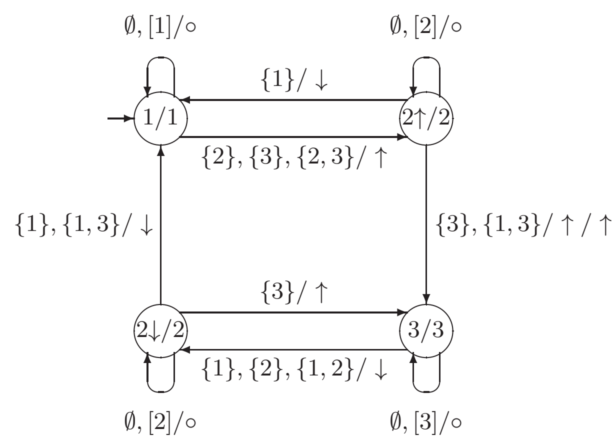
26. Tem-se: $\hat{\delta}(A, \texttt{10}) = \hat{\delta}(B, \texttt{10}) = \hat{\delta}(c, \texttt{10}) = \{C\}$ e $\hat{\delta}(D, \texttt{10}) = \{A, C\}$. Portanto, uma resposta é $w = \texttt{0}$ e $e = C$.

    Um algoritmo basta testar as computações possíveis do AFN para palavras w de tamanho no máximo igual ao número de estados do AFN menos 1.

27. Segue uma máquina de Moore:



28. Segue uma máquina de Moore/Mealy:



29. Acrescentar um símbolo ao alfabeto de saída e, para cada estado final f, fazer $\sigma'(f) = \sigma(f)a$, onde σ é a função de saída anterior, σ' a nova e a é o símbolo acrescentado. Além disso, $\sigma'(e) = \sigma(e)$ se e não é estado final. Dessa forma, a palavra é considerada aceita se, e somente se, o último símbolo da palavra de saída é a. E a saída propriamente, deve ser considerada como sendo a saída atual, removidas todas as ocorrências de a.

30. Não aumenta. Seja $P = \{x_1, x_2, \ldots, x_n\}$ o conjunto das palavras que aparecem nas contra-domínio da função de saída: $\sigma : E \to P$. Uma máquina de Mealy equivalente a uma máquina de Mealy M com tal função de saída seria uma máquina M' idêntica a M, só que sua função de saída seria do tipo $\sigma' : E \to \Delta$, onde $\Delta = \{a_1, a_2, \ldots, a_n\}$ e $\sigma'(e, a) = a_i$ sse $\sigma(e, a) = x_i$. Desta forma, a saída de M seria obtida da saída $s(i', w)$ de M' simplesmente substituindo-se nesta cada a_i por x_i.

31. **a)** ER: $(1(0+1))^*(\lambda+1)$

GR:

$$I \to \mathtt{1}P \mid \lambda$$
$$P \to \mathtt{0}I \mid \mathtt{1}I \mid \lambda$$

b) ER: $\lambda + 0[(1+01)(0+10)]^*(\lambda+0) + 1[(0+10)(1+01)]^*(\lambda+1)$

GR:

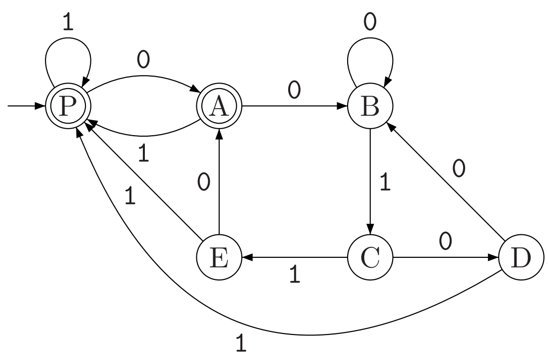
$$P \to \mathtt{0}Z_1 \mid \mathtt{1}U_1 \mid \lambda$$
$$Z_1 \to \mathtt{0}Z_2 \mid \mathtt{1}U_1 \mid \lambda$$
$$Z_2 \to \mathtt{1}U_1 \mid \lambda$$
$$U_1 \to \mathtt{0}Z_1 \mid \mathtt{1}U_2 \mid \lambda$$
$$U_2 \to \mathtt{0}Z_1 \mid \lambda$$

c) ER: $(1+0111)^*(\lambda+0+01+011)$

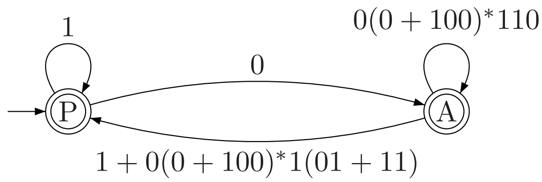
GR:

$$P \to \mathtt{1}P \mid \mathtt{0}A \mid \lambda$$
$$A \to \mathtt{1}B \mid \lambda$$
$$B \to \mathtt{1}C \mid \lambda$$
$$B \to \mathtt{1}P \mid \lambda$$

d) Segue um diagrama ER inicial:



Eliminando-se D, E, C e B, nessa ordem, obtém-se o diagrama ER:



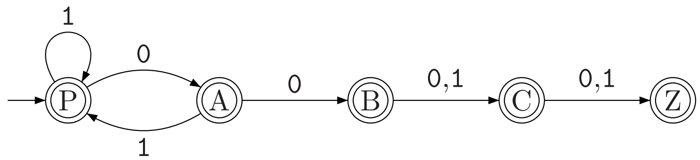
ER: $[1+0[0(0+100)^*110]^*[1+0(0+100)^*1(01+11)]]^*(\lambda+0)$.

GR:

$$P \to \mathtt{0}A \mid \mathtt{1}P \mid \lambda$$
$$A \to \mathtt{0}B \mid \lambda$$
$$B \to \mathtt{0}B \mid \mathtt{1}C$$
$$C \to \mathtt{0}D \mid \mathtt{1}E$$
$$D \to \mathtt{0}B \mid \mathtt{1}P$$

$E \to \mathtt{0}A \mid \mathtt{1}P$

e) Segue um AFD para a linguagem:



ER: $[(1+01)^*[\lambda+0+00(\lambda+0+1)(\lambda+0+1)]$.

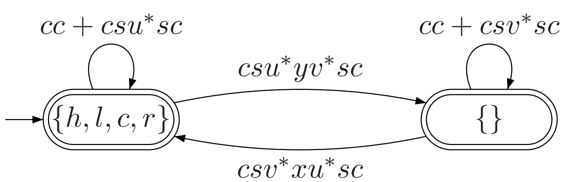
GR:

$P \to \mathtt{0}A \mid \mathtt{1}P \mid \lambda$
$A \to \mathtt{0}B \mid \mathtt{1}P \mid \lambda$
$B \to \mathtt{0}C \mid \mathtt{1}C \mid \lambda$
$C \to \mathtt{0} \mid \mathtt{1} \mid \lambda$

32. Ordem de eliminação de estados: $\{l,r\}$, $\{h,c\}$, $\{r\}$, $\{l\}$, $\{h,c,r\}$, $\{h,l,c\}$, $\{h,l,r\}$, $\{c\}$. Sejam:

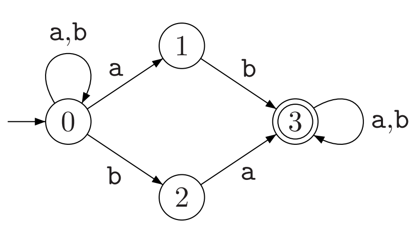
- $x = r(cc)^*cl + l(rr)^*cr$;
- $y = lc(cc)^*r + rc(rr)^*l$;
- $u = ss + rr + lc(cc)^*cl + rc(rr)^*cr$;
- $v = ss + r(cc)^*r + l(rr)^*l$;

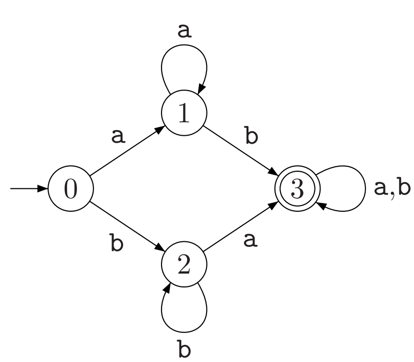
Obtém-se o diagrama ER:



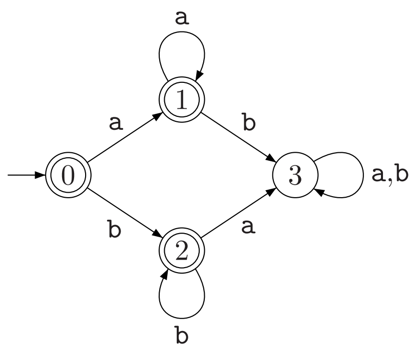
Uma ER é: $(cc+csu^*sc)^*csu^*yv^*sc[cc+csv^*sc+csv^*xu^*sc(cc+csu^*sc)^*csu^*yv^*sc]^*$.

33. **a)** Pode-se obter trivialmente um AFN e um AFD para $L(r_1)$ com os diagramas de estados:



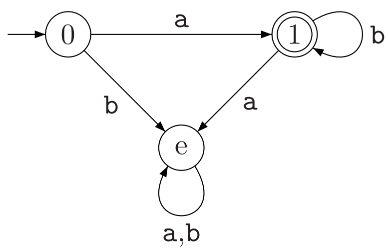


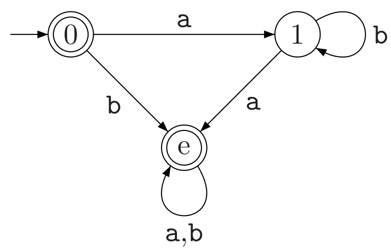
Do último, obtém-se o AFD que reconhece $\overline{L(r_1)}$:



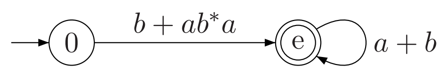
Deste obtém-se, finalmente a ER $\lambda + a^+ + b^+$.

b) AFDs para $L(r_2)$ e $\overline{L(r_2)}$:



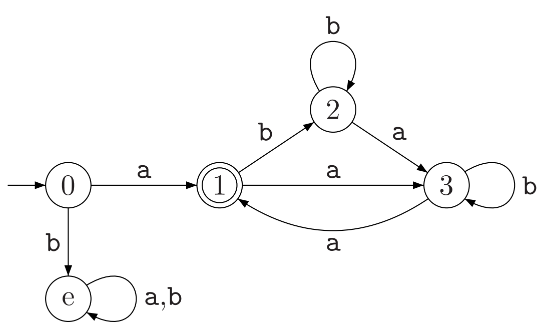


Do último obtém-se o diagrama ER:

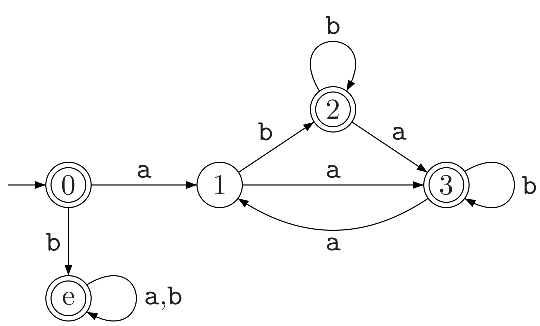


A ER é, então, $(b + ab^*a)(a + b)^*$.

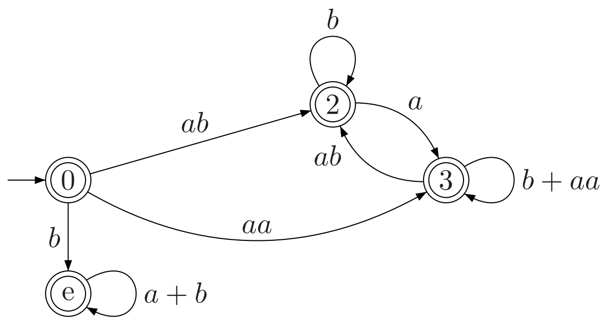
c) Primeiro, obtém-se um AFD para $L(r_3)$:



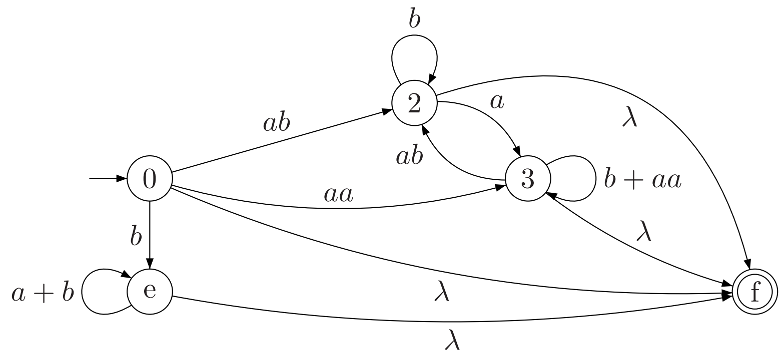
Em seguida, um AFD para $\overline{L(r_3)}$:



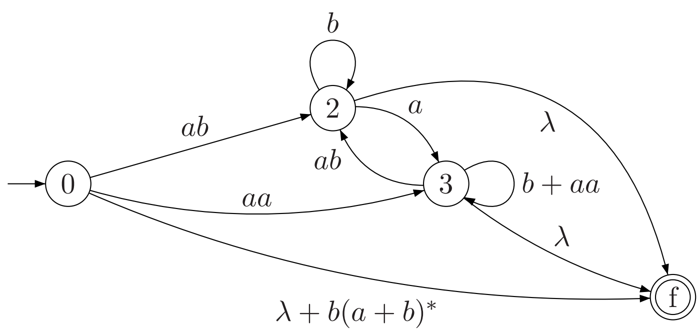
O diagrama ER com o estado 1 eliminado:



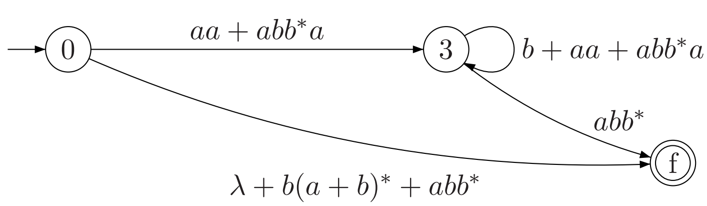
Para simplificar a obtenção da ER, é conveniente acrescentar um estado, que será o novo estado final, e transições sob λ dos estados finais anteriores para ele (o que, evidentemente, não altera a linguagem aceita):



Eliminando-se o estado e (lembrando que $r\lambda = r$ para qualquer ER r):

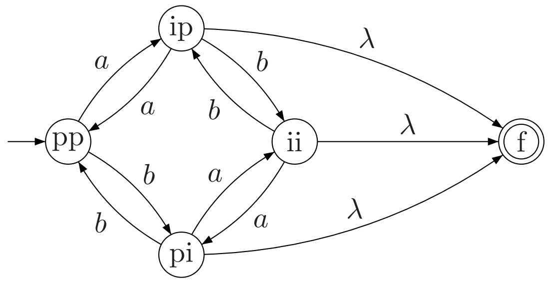


Eliminando-se o estado 2, obtém-se:

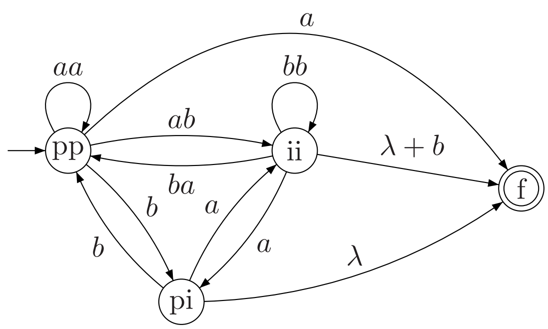


Finalmente, obtém-se a ER: $\lambda + b(a+b)^* + abb^* + (aa+abb^*a)(b+aa+abb^*a)^*abb^*$.

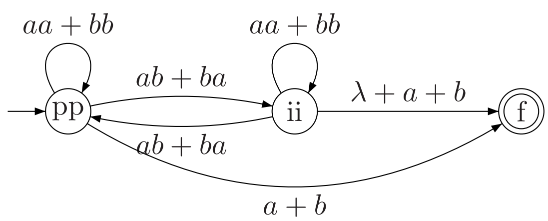
d) Um diagrama ER inicial, já com estado final e transições λ, como no exercício anterior:



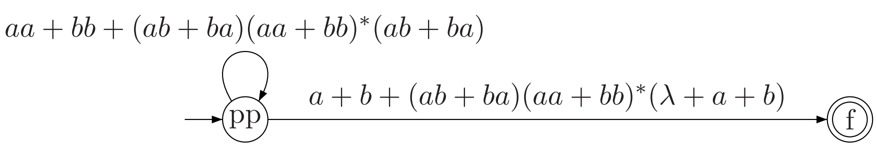
Eliminando-se o estado ip:



Eliminando-se o estado pi:

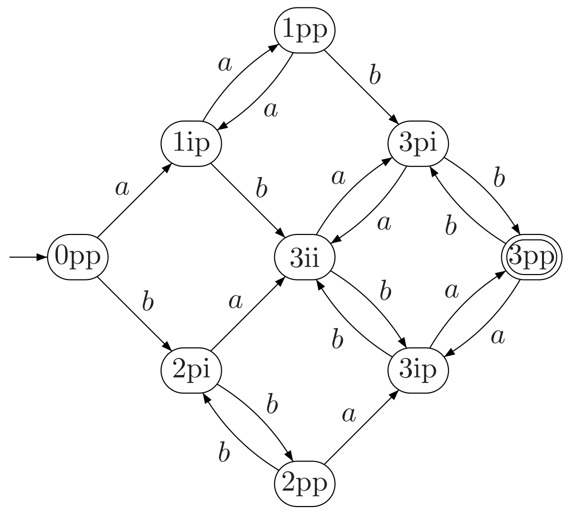


Finalmente, eliminando-se o estado ii:

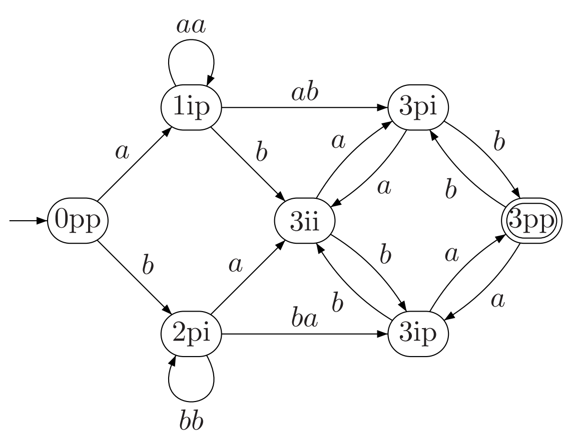


A ER: $[aa + bb + (ab + ba)(aa + bb)^*(ab + ba)]^*[a + b + (ab + ba)(aa + bb)^*(\lambda + a + b)]$.

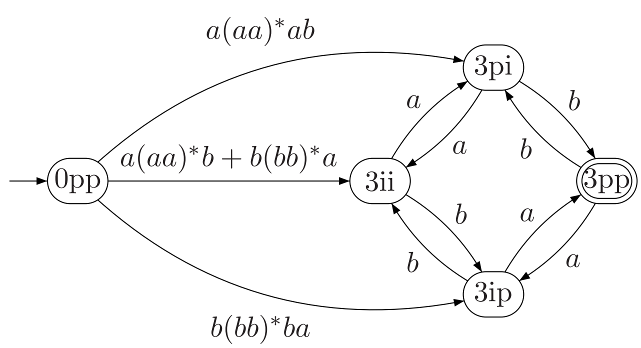
e) Um diagrama ER inicial, obtido a partir do produto dos AFDs para $L(r_1)$ (ver item a) e $L(r_4)$ (ver item d):



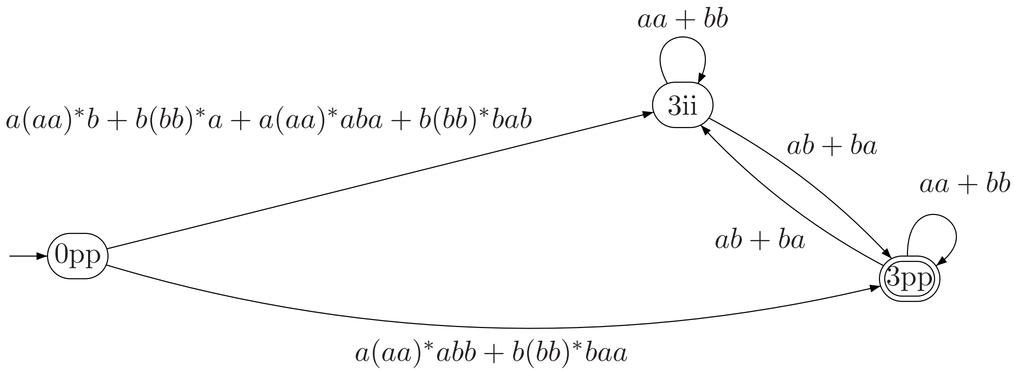
Eliminando-se os estados 1pp e 2pp, obtém-se:



Eliminando-se os estados 1ip e 2pi:

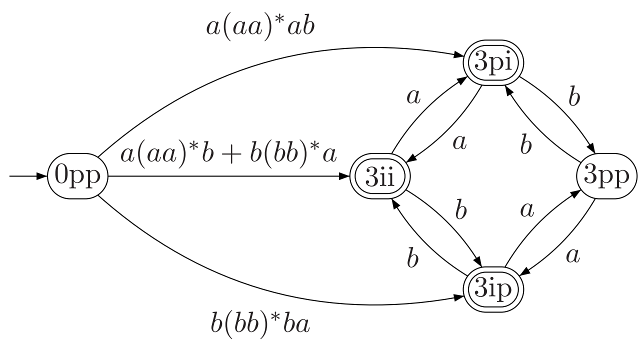


Eliminando-se os estados 3ip e 3pi:

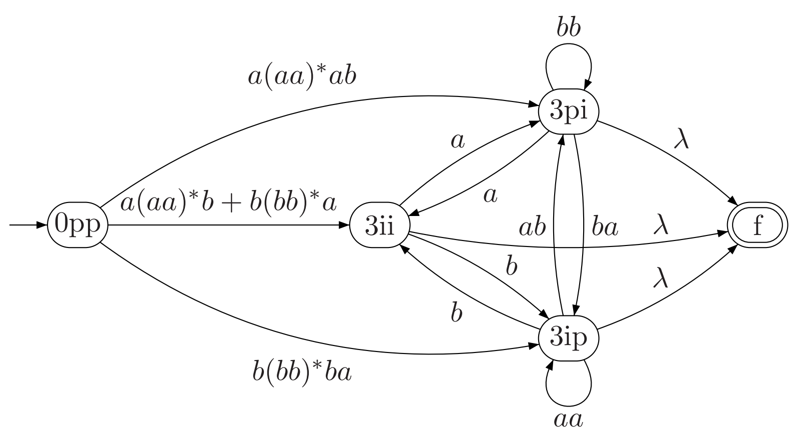


Após eliminar o estado 3ii, obtém-se a ER: $r+st^*$, onde onde $r=a(aa)^*abb+b(bb)^*baa$, $s=(a(aa)^*b+b(bb)^*a+a(aa)^*aba+b(bb)^*bab)(aa+bb)^*(ab+ba)$ e $t=aa+bb+(ab+ba)(aa+bb)^*(ab+ba)$.

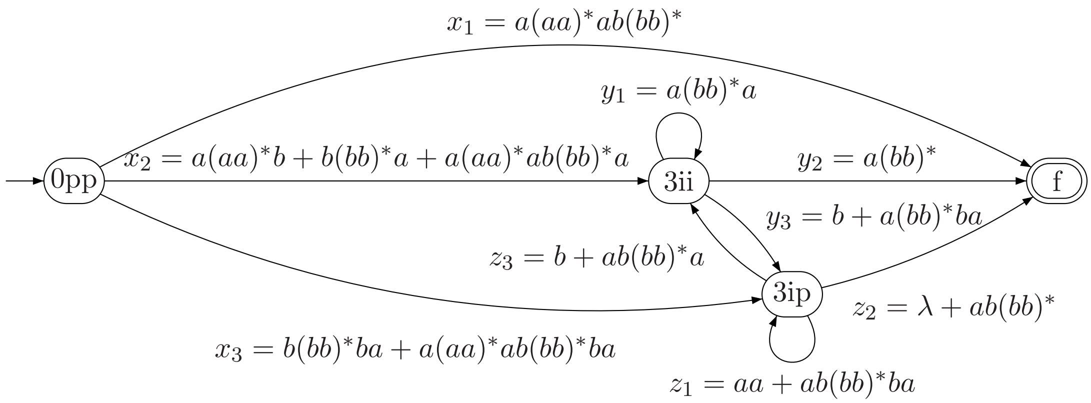
f) O diagrama ER inicial é o mesmo que o do exemplo anterior, só que com os estados finais sendo 3ii, 3ip e 3pi. Fazendo-se como no exemplo anterior, após a eliminação dos estados 1pp, 2pp, 1ip e 2pi, obtém-se:



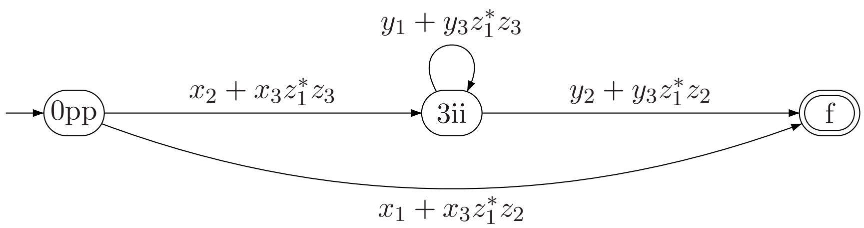
Após eliminar 3pp e acrescentar um novo estado final com transições sob λ:



Eliminando-se 3pi e criando-se nomes para as ERs de cada transição, para simplificar o trabalho posterior, obtém-se:



Para simplificar, serão usados os rótulos criados na figura anterior. Com isto, eliminando-se 3ip, obtém-se:

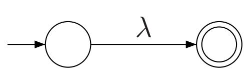


A ER: $x_1 + x_3 z_1^* z_2 + (x_2 + x_3 z_1^* z_3)(y_1 + y_3 z_1^* z_3)^*(y_2 + y_3 z_1^* z_2)$.

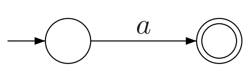
34. Para maior legibilidade serão usados diagramas de estado.

a) AFN para $\emptyset$:

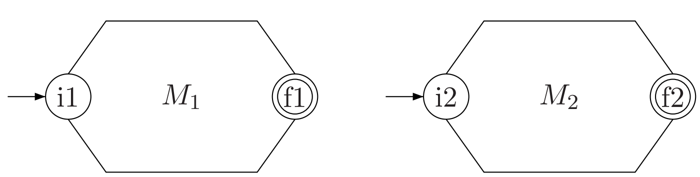
AFN para $\{\lambda\}$:



AFN para $\{a\}$:

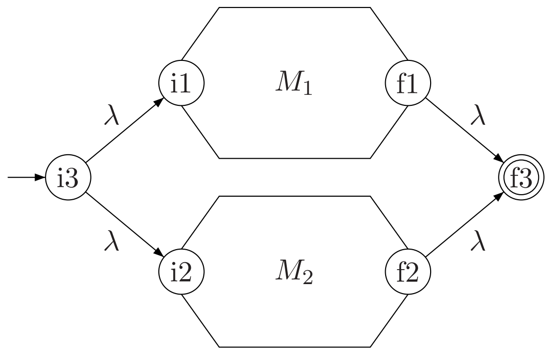


b) Suponha a existência de AFNλs M_1 e M_2 satisfazendo às três condições. Diagramaticamente:

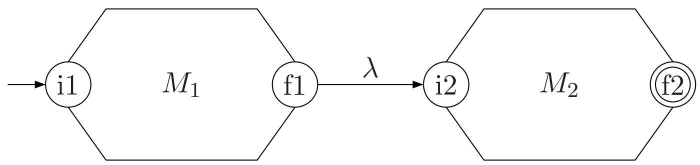


A seguir, mostra-se como construir AFNλs para $L(M_1) \cup L(M_2)$, $L(M_1)L(M_2)$ e $L(M_1)^*$.

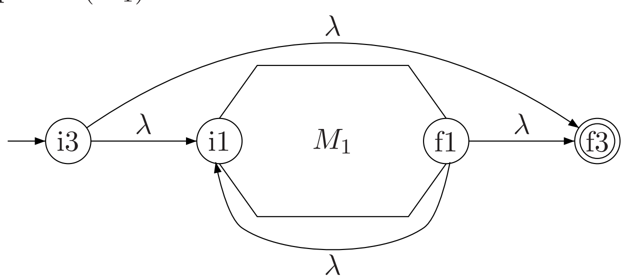
AFNλ para $L(M_1) \cup L(M_2)$:



AFNλ para $L(M_1)L(M_2)$:



AFNλ para $L(M_1)^*$:



As três condições, sendo asseguradas na parte base (a) e também nos processos de construção da parte indutiva (b), além de não atrapalhar em nada, simplificam esses mesmos processos, visto que propiciam a necessidade de considerar *apenas* os casos em que M_1 e M_2 satisfaçam essas condições. Por exemplo, não é preciso considerar casos em que M_1 ou M_2 tenham mais de um estado final.

35. Dada uma ER r sob Σ, seja $\gamma(r)$ uma ER *modificada* em que os símbolos repetidos são substituídos por novos, de forma que $\gamma(r)$ fique sem símbolos repetidos. Seja $\Sigma(r)$ o conjunto dos símbolos que ocorrem em r. (Exemplo: se $r = (a+bb)^*((aa)^*b)$, com $\Sigma(r) = \{a, b\}$, poder-se-ia ter $\gamma(r) = (a + bb_1)^*((a_1a_2) * b_2)$, com $\Sigma(\gamma(r)) = \{a, a_1, a_2, b, b_1, b_2\}^*$.) Seja ainda $\sigma(x)$ o símbolo de $\Sigma(r)$ correspondente a $x \in \Sigma(\gamma(r))$. (Para o exemplo, $\sigma(a) = \sigma(a_1) = \sigma(a_2) = a$ e $\sigma(b) = \sigma(b_1) = \sigma(b_2) = b$.) Para a construção do AFN serão usadas três funções, definidas recursivamente a seguir.

    Dada uma ER modificada r, $\text{prm}(r)$ dá o conjunto dos símbolos de $\Sigma(r)$ que podem iniciar uma palavra de $L(r)$. Recursivamente:

    (a) $\text{prm}(\emptyset) = \emptyset$,
    $\text{prm}(\lambda) = \emptyset$,

prm(x) = $\{x\}$ para todo $x \in \Sigma(r)$;

(b) $\text{prm}(r+s) = \text{prm}(r) \cup \text{prm}(s)$,
$\text{prm}(rs) = \text{prm}(r) \cup \text{prm}(s)$, se $\lambda \in L(r)$, e $\text{prm}(rs) = \text{prm}(r)$, caso contrário,
$\text{prm}(r^*) = \text{prm}(r)$.

Dada uma ER modificada r, $\text{ult}(r)$ dá o conjunto dos símbolos de $\Sigma(r)$ que podem terminar uma palavra de $L(r)$. Recursivamente:

(a) $\text{ult}(\emptyset) = \emptyset$,
$\text{ult}(\lambda) = \emptyset$,
$\text{ult}(x) = \{x\}$ para todo $x \in \Sigma(r)$;

(b) $\text{ult}(r+s) = \text{ult}(r) \cup \text{ult}(s)$,
$\text{ult}(rs) = \text{ult}(r) \cup \text{ult}(s)$, se $\lambda \in L(s)$, e $\text{ult}(rs) = \text{ult}(s)$, caso contrário,
$\text{ult}(r^*) = \text{ult}(r)$.

Dada uma ER modificada r e um símbolo $x \in \Sigma(r)$, $\text{prx}(r, x)$ dá o conjunto dos símbolos de $\Sigma(r)$ que podem vir após x em palavras de $L(r)$. Recursivamente:

(a) $\text{prx}(x, x) = \emptyset$ para todo $x \in \Sigma(r)$;

(b) $\text{prx}(r+s, x) = \text{prx}(r, x)$, se $x \in \Sigma(r)$, e $\text{prx}(r+s, x) = \text{prx}(s, x)$, se $x \in \Sigma(s)$,
$\text{prx}(rs, x) = \text{prx}(r, x)$, se $x \in \Sigma(r) - \text{ult}(r)$, $\text{prx}(rs, x) = \text{prx}(r, x) \cup \text{prm}(s)$, se $x \in \text{ult}(r)$, e $\text{prx}(rs, x) = \text{prx}(s, x)$, se $x \in \Sigma(s)$,
$\text{prx}(r^*, x) = \text{prx}(r, x)$, se $x \in \Sigma(r) - \text{ult}(r)$, e $\text{prx}(r^*, x) = \text{prx}(r, x) \cup prm(r)$, se $x \in \text{ult}(r)$.

O seguinte AFN, denominado autômato de Glushkov, reconhece $L(r)$:

$$M_r = (E_r \cup \{i\}, \Sigma, \delta_r, \{i\}, F_r), \text{ em que:}$$

- $E_r = \Sigma(\gamma(r))$;
- $\delta_r(i, a) = \{y \in \text{prm}(\gamma(r)) \mid \sigma(y) = a\}$, e $\delta_r(x, a) = \{y \in \text{prx}(\gamma(r), x) \mid \sigma(y) = a\}$ para todo $x \in \Sigma(\gamma(r))$ e $a \in \Sigma$;
- $F_r = \text{ult}(\gamma(r)) \cup \{i\}$, se $\lambda \in L(r)$, e $F_r = \text{ult}(\gamma(r))$, se $\lambda \notin L(r)$.

36. Definição recursiva de $f\lambda(X)$, sendo $X \subseteq E$:

(a) $X \subseteq f\lambda(X)$;

(b) se $e \in f\lambda(X)$ e $r \in R$ e $\lambda \in L(r)$, então $\delta(e, r) \subseteq f\lambda(X)$.

Denotando por $C_{e,x}$ o conjunto $\bigcup_{r \in R \wedge x \in L(r)} \delta(e, r)$, $\hat{\delta}$ pode ser definida recursivamente assim:

(a) $\hat{\delta}(\emptyset, w) = \emptyset$ para $w \in \Sigma^*$,
$\hat{\delta}(X, \lambda) = f\lambda(X)$ para $X \subseteq E$;

(b) $\hat{\delta}(X, w) = \bigcup_{xy = w \wedge x \neq \lambda} \hat{\delta}(\bigcup_{e \in f\lambda(X)} C_{e,x}, y)$ para $X \subset E$ e $w \in \Sigma^+$.

37. Seja um diagrama ER $D = (E, \Sigma, \delta, I, F)$. $L(D) = \{w \in \Sigma^* \mid \hat{\delta}(I, w) \cap F \neq \emptyset\}$.

38. Seja $\Sigma = \{a_1, a_2, \ldots, a_n\}$. Então pode-se substituir as expressões básicas para expressar Σ e Σ^* por $a_1 + a_2 + \cdots a_n$ e $(a_1 + a_2 + \cdots a_n)^*$, respectivamente, e substituir as ERs das formas $r \cap s$ (interseção) e r^c (complementação) por ERs assim obtidas, respectivamente:

- $r \cap s$:
    1. obter um ADF M_r para $L(r)$ e outro para $L(s)$ (teorema 12);
    2. obter um ADF $M_{r\cap s}$ para $L(r \cap s)$ fazendo o produto de M_r e M_s (teorema 4);
    3. obter uma ER a partir de $M_{r\cap s}$ (teorema 13).
- r^c:
    1. obter um ADF M_r para $L(r)$ (teorema 12);
    2. obter um ADF M_{r^c} para $\overline{L(r)}$ a partir de M_r (teorema 4);
    3. obter uma ER a partir de M_{r^c} (teorema 13).

39. Dada uma gramática linear à direita $G = (V, \Sigma, R, P)$, pode-se obter uma linear à esquerda $G^R = (V, \Sigma, R', P)$ em que $R' = \{X \to w^R \,|\, X \to w \in R\}$, e vice-versa. Pode-se provar por indução sobre o número de regras utilizadas para gerar w, que $w \in L(G)$ se, e somente se, $w^R \in L(G^R)$. Como a classe das linguagens regulares é fechada sob reverso, segue-se que as gramáticas lineares à esquerda geram exatamente a classe das linguagens regulares.

40. **a)** Seja um AFNE $M = (E, \Sigma, \delta, I, F)$. Uma gramática que gera $L(M)$ seria $(E \cup \{P\}, \Sigma, R, P)$, em que $P \notin E$ e R consta da regras:
    - $P \to we'$, para cada $w \neq \lambda$ e e' tais que $e' \in \delta(e, w)$ para algum $e \in f\lambda(I)$;
    - $P \to \lambda$, se $f\lambda(I) \cap F \neq \emptyset$;
    - $e \to we''$, para cada $w \neq \lambda$ e e'' tais que $e'' \in \delta(e', w)$ para algum $e' \in f\lambda(\{e\})$;
    - $e \to \lambda$, para cada e tal que $f\lambda(\{e\}) \cap F \neq \emptyset$.

    b) Seja uma gramática (V, Σ, R, P). Um AFNE que reconhece $L(G)$ seria $(V, \Sigma, \delta, \{P\}, F)$, em que:
    - $Y \in \delta(X, w)$ se, e somente se, $X \to wY \in R$;
    - $F = \{X \,|\, X \to \lambda \in R\}$.

41. Sim, é possível. Por exemplo, a gramática

$$
\begin{aligned}
P &\to \mathtt{0}A \,|\, \lambda \\
A &\to P\mathtt{1}
\end{aligned}
$$

gera $\{\mathtt{0}^n\mathtt{1}^n \,|\, n \geq 0\}$, uma linguagem não regular.

Capítulo 3

Autômatos de Pilha

3.1 Uma Introdução Informal

Nesta seção não há exercícios.

3.2 Autômatos de Pilha Determinísticos

1. ($\rightarrow$) Sejam $[e_1, z_1] \in \delta(e, a, b)$ e $[e_2, z_2] \in \delta(e, a', b')$ duas transi¸oes compatíveis, ou seja, tais que $(a = a' \vee a = \lambda \vee a' = \lambda) \wedge (b = b' \vee b = \lambda \vee b' = \lambda)$. Existem 9 casos a considerar:

   1. $a = a' \wedge b = b'$. Então $[e, ay, b\alpha] \vdash [e_1, y, z_1\alpha]$ e $[e, ay, b\alpha] \vdash [e_2, y, z_2\alpha]$.
   2. $a = a' \wedge b = \lambda$. Então $[e, ay, b'\alpha] \vdash [e_1, y, z_1b'\alpha]$ e $[e, ay, b'\alpha] \vdash [e_2, y, z_2\alpha]$.
   3. $a = a' \wedge b' = \lambda$. Então $[e, ay, b\alpha] \vdash [e_1, y, z_1\alpha]$ e $[e, ay, b\alpha] \vdash [e_2, y, z_2b\alpha]$.
   4. $a = \lambda \wedge b = b'$. Então $[e, a'y, b\alpha] \vdash [e_1, a'y, z_1\alpha]$ e $[e, a'y, b\alpha] \vdash [e_2, y, z_2\alpha]$.
   5. $a = \lambda \wedge b = \lambda$. Então $[e, a'y, b'\alpha] \vdash [e_1, a'y, z_1b'\alpha]$ e $[e, a'y, b'\alpha] \vdash [e_2, y, z_2\alpha]$.
   6. $a = \lambda \wedge b' = \lambda$. Então $[e, a'y, b\alpha] \vdash [e_1, a'y, z_1\alpha]$ e $[e, a'y, b\alpha] \vdash [e_2, y, z_2b\alpha]$.
   7. $a' = \lambda \wedge b = b'$. Então $[e, ay, b\alpha] \vdash [e_1, y, z_1\alpha]$ e $[e, ay, b\alpha] \vdash [e_2, ay, z_2\alpha]$.
   8. $a' = \lambda \wedge b = \lambda$. Então $[e, ay, b'\alpha] \vdash [e_1, y, z_1b'\alpha]$ e $[e, ay, b'\alpha] \vdash [e_2, y, z_2\alpha]$.
   9. $a' = \lambda \wedge b' = \lambda$. Então $[e, ay, b\alpha] \vdash [e_1, y, z_1\alpha]$ e $[e, ay, b\alpha] \vdash [e_2, ay, z_2b\alpha]$.

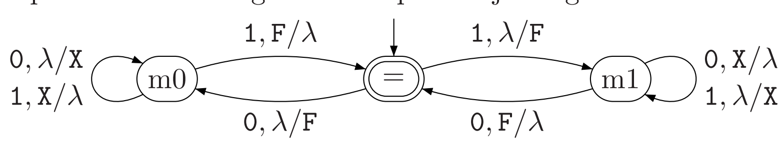
   Para cada caso, mostrou-se que as duas transições podem ocorrer simultaneamente para a mesma configuração.

   ($\leftarrow$) Suponha que duas transições $[e_1', z_1] \in \delta(e_1, a_1, b_1)$ e $[e_2', z_2] \in \delta(e_2, a_2, b_2)$ podem ocorrer simultaneamente a partir de uma configuração $[e, x, \alpha]$. Então:

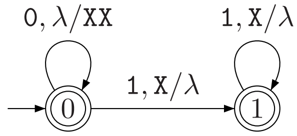
   i. $e_1 = e_2 = e$;
   ii. $a_1 = a_2 =$ primeiro símbolo de x, ou a_1 ou a_2, ou ambos, são λ;
   iii. $b_1 = b_2 =$ primeiro símbolo de α, ou b_1 ou b_2, ou ambos, são λ.

   Em outras palavras, as duas transições são compatíveis.

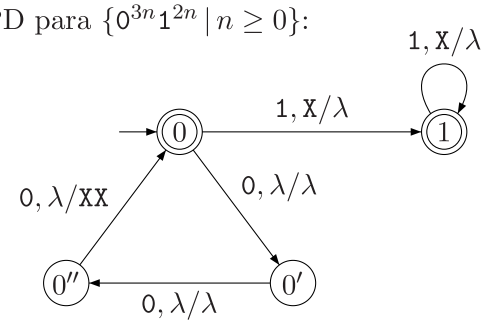
2. Um APD que satisfaz as exigências é aquele cujo diagrama de estados é:



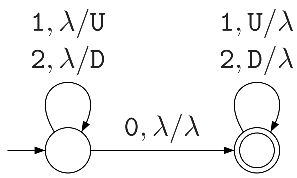
3. **a)** Um APD para $\{\texttt{0}^n\texttt{1}^{2n} \mid n \geq 0\}$:



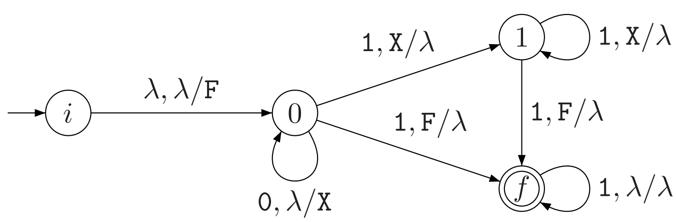
b) Um APD para $\{\texttt{0}^{3n}\texttt{1}^{2n} \mid n \geq 0\}$:



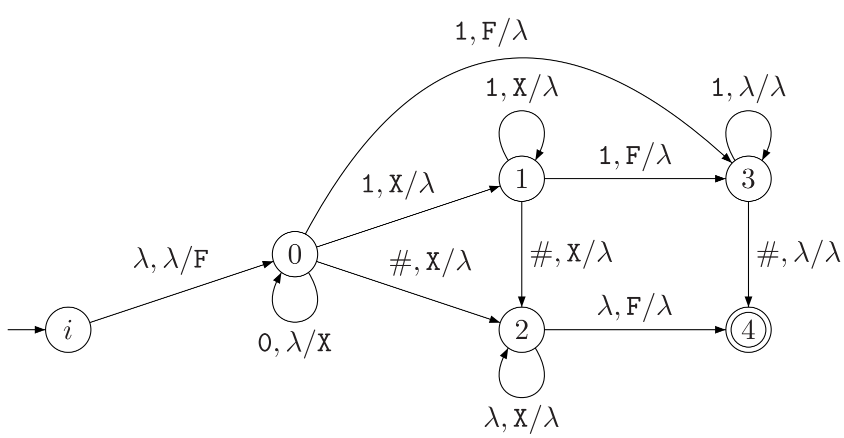
c) Um APD para $\{w\texttt{0}w^R \mid w \in \{\texttt{1},\texttt{2}\}^*\}$:



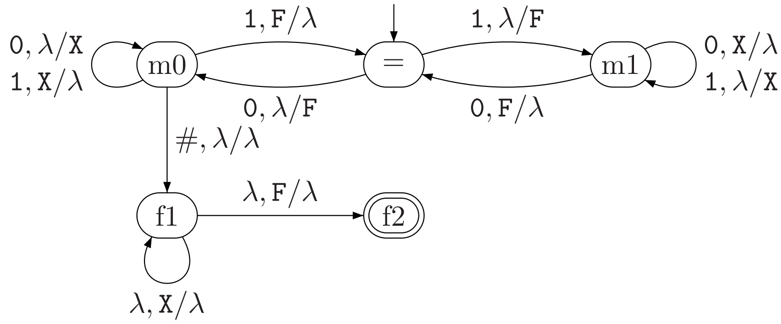
d) Um APD para $\{\texttt{0}^m\texttt{1}^n \mid m < n\}$:



e) Um APD para $\{\texttt{0}^m\texttt{1}^n\# \mid m \neq n\}$:



f) Um APD para $\{w\# \in \{\texttt{0},\texttt{1}\}^* \mid$ o número de $\texttt{0}$s em w é maior que o de $\texttt{1}$s$\}$:

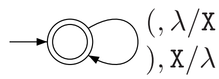


4. **a)** Para reconhecer uma palavra w, não há como um APD reconhecer o término da primeira metade de w para que possa finalizar o seu armazenamento e começar a ler a segunda metade e compará-la com a primeira.

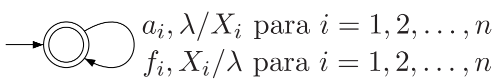
   b) Durante seu processamento, o APD deve necessariamente registrar a diferença entre os números de `0`s e de `1`s. Para isso, deve ser usada a pilha, pois não existe limite para a diferença. Com isto, a pilha poderá não estar vazia quando a palavra terminar. E o APD não tem como saber que a palavra termina.

   c) Como no item anterior, se o número de `0`s da palavra for maior que o de `1`s, o APD poderá estar com a pilha não vazia ao término da palavra (representado a diferença), sem possibilidade de reconhecer que a palavra terminou.

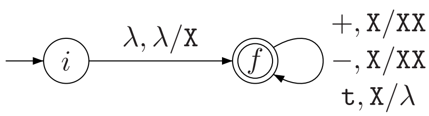
   d) Análoga ao item (b).

5. Um APD para parênteses balanceados: 

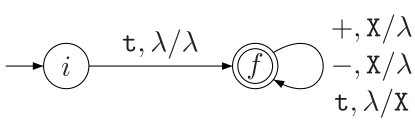
   Generalizando para n parênteses balanceados (a_i, f_i):



6. Um APD para expressões prefixadas é aquele cujo diagrama de estados é:

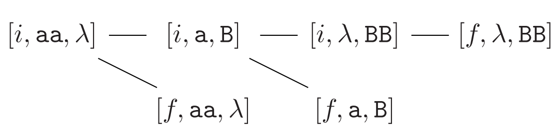


7. O diagrama de estados de um APD para expressões posfixadas:



3.3 Autômatos de Pilha Não Determinísticos

1. **a)** Árvore de computações para `aa`:



Árvore de computações para bb:

$[i, \texttt{bb}, \lambda]$ —— $[f, \texttt{bb}, \lambda]$

Árvore de computações para aabcc:

$[i, \texttt{aabcc}, \lambda]$ —— $[i, \texttt{abcc}, \texttt{B}]$ —— $[i, \texttt{bcc}, \texttt{BB}]$ —— $[f, \texttt{bcc}, \texttt{BB}]$ —— $[f, \texttt{cc}, \texttt{CB}]$ —— $[f, \texttt{c}, \texttt{B}]$

$[f, \texttt{aabcc}, \lambda]$ $\quad$ $[f, \texttt{abcc}, \texttt{B}]$

Árvore de computações para aabcc:

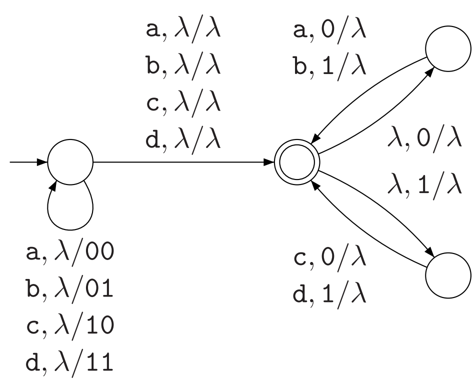
$[i, \texttt{aabcbc}, \lambda]$ —— $[i, \texttt{abcbc}, \texttt{B}]$ —— $[i, \texttt{bcbc}, \texttt{BB}]$ —— $[f, \texttt{bcbc}, \texttt{BB}]$ —— $[f, \texttt{cbc}, \texttt{CB}]$ —— $[f, \texttt{bc}, \texttt{B}]$

$[f, \texttt{aabcbc}, \lambda]$ $\quad$ $[f, \texttt{abcbc}, \texttt{B}]$ $\quad$ $[f, \lambda, \lambda]$ —— $[f, \texttt{c}, \texttt{C}]$

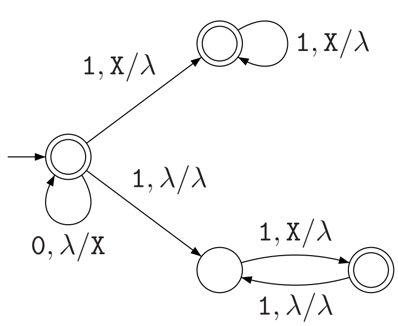
Apenas aabcc é reconhecida por M.

b) $L(M) = \{\texttt{a}^n(\texttt{bc})^n \mid n \geq 0\}$.

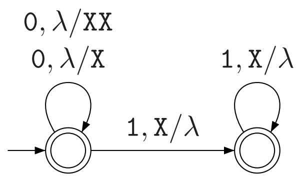
2. APN para $\{w \in \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}^* \mid w = w^R\}$:



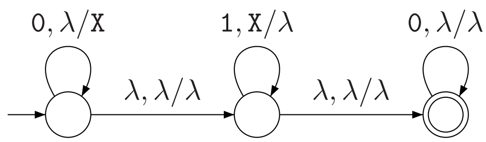
3. **a)** APN para $\{\texttt{0}^n\texttt{1}^n \mid n \geq 0\} \cup \{\texttt{0}^n\texttt{1}^{2n} \mid n \geq 0\}$:



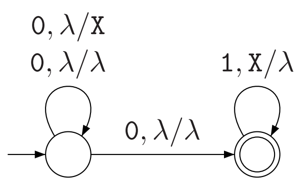
b) APN para $\{\mathtt{0}^n\mathtt{1}^k \mid n \le k \le 2n\}$:



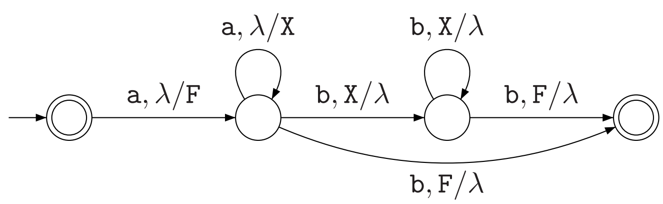
c) APN para $\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^k \mid n, k \ge 0\}$:



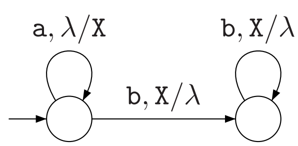
d) APN para $\{\mathtt{0}^m\mathtt{1}^n \mid m > n\}$:



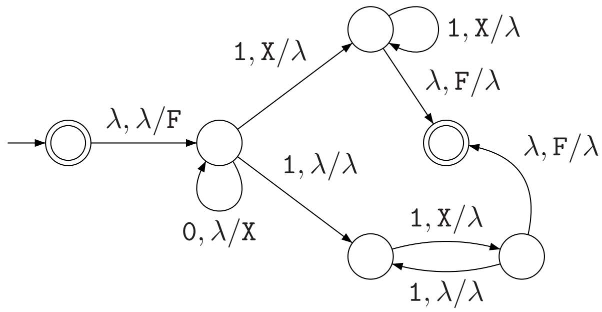
4. Reconhecimento por estado final:



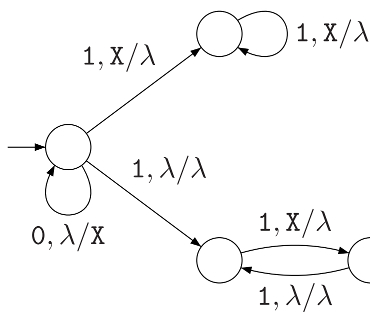
Reconhecimento por pilha vazia:



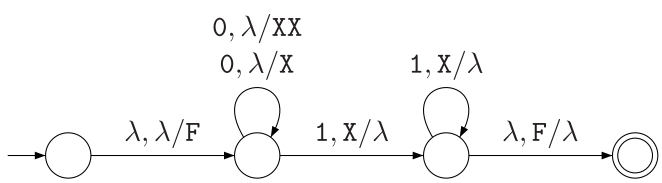
5. **a)** Por estado final:



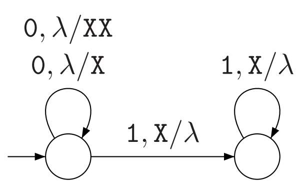
Por pilha vazia:



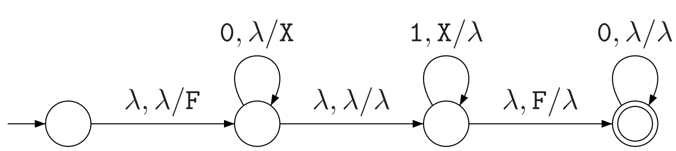
b) Por estado final:



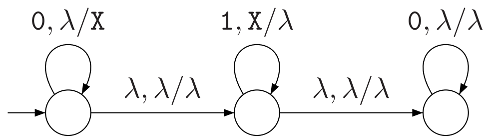
Por piha vazia:



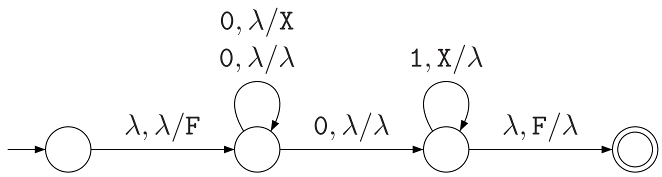
c) Por estado final:



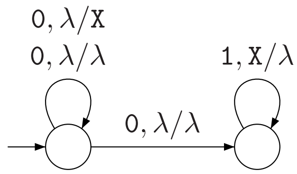
Por pilha vazia:



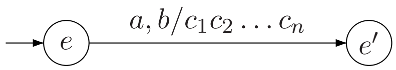
d) Por estado final:



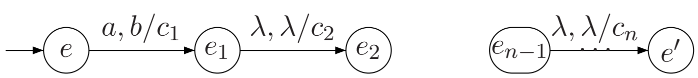
Por pilha vazia:



6. Uma transição da forma



para $n \geq 2$, tem o mesmo efeito que n transições:



em que os $n-1$ estados e_1, e_2, $\ldots$, e_{n-1} são estados *novos*.

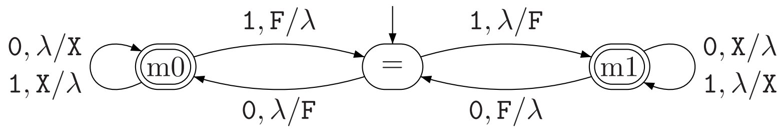
7. Seja um AFNE $M = (E, \Sigma, \delta, I, F)$. Um APN equivalente seria

$$M' = (\{i, m, f\} \cup N, \Sigma, E, \delta', \{i\}, \{f\})$$

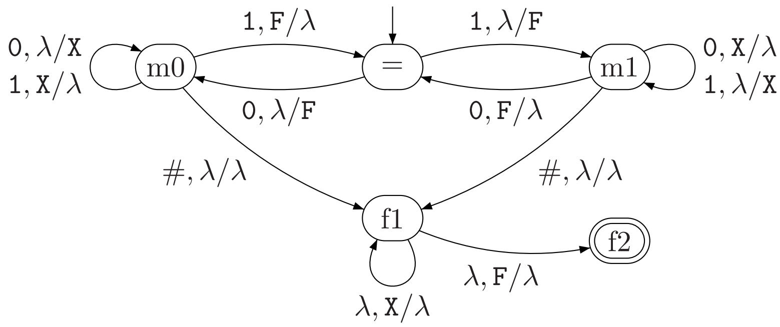
em que N e δ' são dados por:

- $\delta'(i, \lambda, \lambda) = \{[m, e] \,|\, e \in I\}$;
- começando com $N = \emptyset$, para cada transição do AFNE da forma $e' \in \delta(e, w)$, $w \in \Sigma^*$:
  - se $|w| \leq 1$, $[m, e'] \in \delta'(m, w, e)$;
  - se $w = a_1 a_2 \cdots a_n$ para $n \geq 2$, fazer $[e_1, e'] \in \delta'(m, a_1, e)$, $[e_2, \lambda] \in \delta'(e_1, a_2, \lambda)$, $\ldots$, $[m, \lambda] \in \delta'(e_{n-1}, a_n, \lambda)$, sendo e_1, e_2, $\ldots$, e_{n-1} estados *novos* acrescidos a N.
- para cada $e \in F$, $[f, \lambda] \in \delta'(m, \lambda, e)$;

8. APD para $L = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid$ o número de 0s em w difere do de 1s$\}$:



APD para $L\#$:



9. Seja um AFD $M = (E, \Sigma, \delta, i, F)$. Um APD equivalente que reconhece por estado final e também por estado final e pilha vazia é $(E, \Sigma, \{\mathtt{X}\}, \delta', i, F)$ em que para todo $e \in E$ e todo $a \in \Sigma$, $\delta'(e, a, \lambda) = [\delta(e, a), \lambda]$.

   Um APD que reconhece $L(M) \cup \{\lambda\}$ por pilha vazia é $(E \cup \{i'\}, \Sigma, \{\mathtt{X}\}, \delta', i')$ em que $i' \notin E$ e δ' é assim definida:

   - $\delta'(i', \lambda, \lambda) = [i, \lambda]$, se $i \in F$, e $\delta'(i', \lambda, \lambda) = [i, \mathtt{X}]$, se $i \notin F$;
   - para cada transição $\delta(e, a) = e'$:
     - se $e \in F$ e $e' \in F$, $\delta'(e, a, \lambda) = [e', \lambda]$;
     - se $e \in F$ e $e' \notin F$, $\delta'(e, a, \lambda) = [e', \mathtt{X}]$;
     - se $e \notin F$ e $e' \in F$, $\delta'(e, a, \mathtt{X}) = [e', \lambda]$;
     - se $e \notin F$ e $e' \notin F$, $\delta'(e, a, \mathtt{X}) = [e', \mathtt{X}]$.

10. Seja um AP $M = (E, \Sigma, \Gamma, \delta, I, F)$. Um AP que reconhece $L(M) - \{\lambda\}$ é $(E \cup \{i'\} \cup N, \Sigma, \Gamma \cup \{\mathtt{F}\}, \delta', \{i'\}, F)$, em que $i' \notin E$, $\mathtt{F} \notin \Gamma$ e δ' e novos estados em N são assim obtidos:

    - $\delta'(i', \lambda, \lambda) = \{[i, \mathtt{F}] \mid i \in I\}$;
    - para cada transição $[e', z] \in \delta(e, a, \mathtt{X})$:
      - $[e', z] \in \delta'(e, a, \mathtt{X})$, e
      - se $a \neq \lambda$, colocar um *novo* estado d em N e acrescentar as transições $[d, \lambda] \in \delta'(e, a, \mathtt{X})$ e $[e', z] \in \delta'(d, \lambda, \mathtt{F})$.

3.4 Gramáticas Livres do Contexto

1. a) Para $\{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\} \cup \{\mathtt{0}^n\mathtt{1}^{2n} \mid n \geq 0\}$:

   $P \rightarrow A \mid B$
   $A \rightarrow \mathtt{0}A\mathtt{1} \mid \lambda$
   $B \rightarrow \mathtt{0}B\mathtt{11} \mid \lambda$

   b) Para $\{\mathtt{0}^n\mathtt{1}^k \mid n \leq k \leq 2n\}$:

   $P \rightarrow \mathtt{0}P\mathtt{1} \mid \mathtt{0}P\mathtt{11} \mid \lambda$

   c) Para $\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^k \mid n, k \geq 0\}$:

$$P \rightarrow AB$$
$$A \rightarrow \mathtt{0}A\mathtt{1} \mid \lambda$$
$$B \rightarrow \mathtt{0}B \mid \lambda$$

d) Para $\{\mathtt{0}^m\mathtt{1}^n \mid m > n\}$:

$$P \rightarrow \mathtt{0}P\mathtt{1} \mid \mathtt{0}P \mid \mathtt{0}$$

2. a) Para $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^{3m+2n+1} \mid m, n \geq 0\}$:

$$P \rightarrow \mathtt{a}P\mathtt{ccc} \mid X$$
$$X \rightarrow \mathtt{b}X\mathtt{cc} \mid \mathtt{c}$$

b) Para $\{\mathtt{a}^n\mathtt{b}^{2n+k}\mathtt{c}^{3k} \mid n, k \geq 0\}$:

$$P \rightarrow AB$$
$$A \rightarrow \mathtt{a}A\mathtt{bb} \mid \lambda$$
$$B \rightarrow \mathtt{b}B\mathtt{ccc} \mid \lambda$$

c) Para $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid n > m + k\}$:

$$P \rightarrow ABC$$
$$A \rightarrow \mathtt{a}A\mathtt{b} \mid \lambda$$
$$B \rightarrow \mathtt{b}B \mid \mathtt{b}$$
$$C \rightarrow \mathtt{b}C\mathtt{c} \mid \lambda$$

3. a) Para L_1:

$$X \rightarrow \mathtt{0}X\mathtt{11} \mid \mathtt{0}X\mathtt{111} \mid \lambda$$

b) Para L_2:

$$Y \rightarrow AB$$
$$A \rightarrow \mathtt{a}A\mathtt{bb} \mid \lambda$$
$$B \rightarrow \mathtt{b}B\mathtt{c} \mid \lambda$$

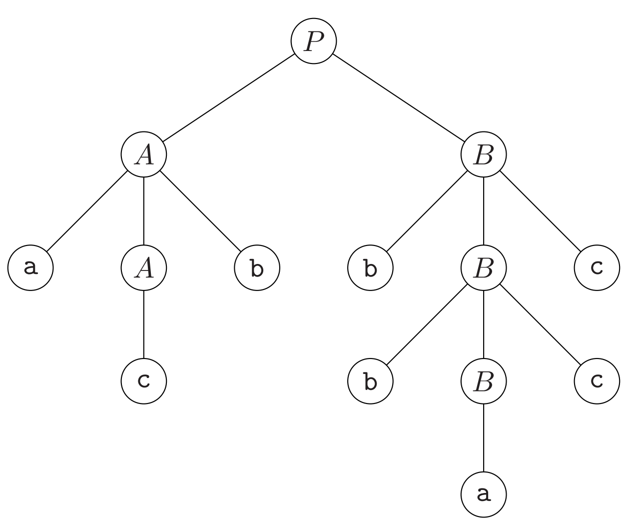
c) Para $(L_1 \cup L_2)^2$:

$$P \rightarrow ZZ$$
$$Z \rightarrow X \mid Y$$

mais as regras de (a) e (b).

4. a) $P \Rightarrow AB \Rightarrow \mathtt{a}A\mathtt{b}B \Rightarrow \mathtt{acb}B \Rightarrow \mathtt{acbb}B\mathtt{c} \Rightarrow \mathtt{acbbb}B\mathtt{cc} \Rightarrow \mathtt{acbbbacc}$

b) AD:



c) $L(G) = \{\mathtt{a}^n\mathtt{cb}^n \mid n \geq 0\}\{\mathtt{b}^n\mathtt{ac}^n \mid n \geq 0\}$.

5. a) G é ambígua, pois existem duas derivações *mais à esquerda* da palavra `aaabb`:

$$P \Rightarrow \mathtt{a}P\mathtt{b} \Rightarrow \mathtt{aaa}P\mathtt{bb} \Rightarrow \mathtt{aaabb}$$
$$P \Rightarrow \mathtt{aa}P\mathtt{b} \Rightarrow \mathtt{aaa}P\mathtt{bb} \Rightarrow \mathtt{aaabb}$$

b) Uma gramática não ambígua equivalente a G é

$$P \rightarrow \mathtt{a}P\mathtt{b} \mid X$$
$$X \rightarrow \mathtt{aa}X\mathtt{b} \mid \lambda$$

6. Sim. Uma gramática ambígua que gera $\{\lambda\}$:

$$P \rightarrow X \mid \lambda$$
$$X \rightarrow \lambda$$

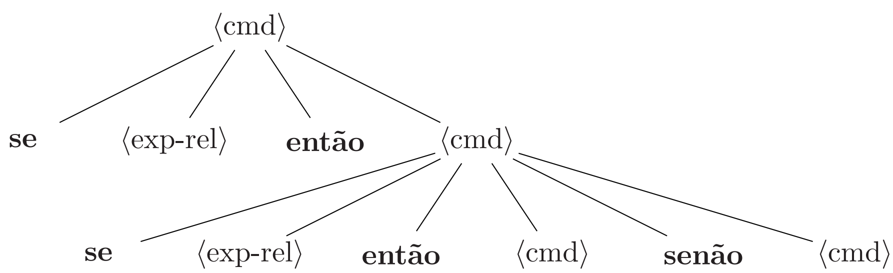
Uma gramática ambígua que gera $\{\mathtt{0}\}$:

$$P \rightarrow X \mid \mathtt{0}$$
$$X \rightarrow \mathtt{0}$$

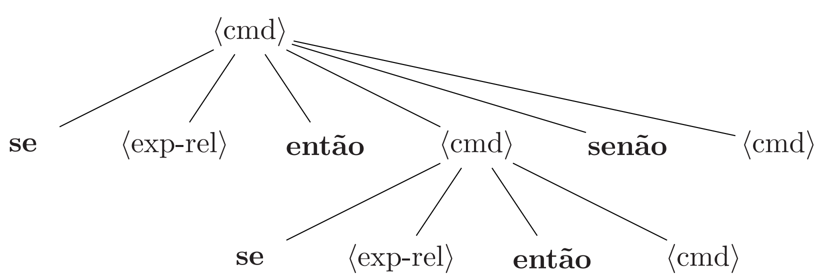
7. Como ⟨cmd⟩ e ⟨exp-rel⟩ são úteis, para mostrar a ambiguidade basta apresentar duas subADs com raiz rotulada ⟨cmd⟩ e gerando uma (a mesma) forma sentencial contendo apenas terminais e essas duas variáveis. Duas ADs para

se ⟨exp-rel⟩ **então se** ⟨exp-rel⟩ **então** ⟨cmd⟩ **senão** ⟨cmd⟩

são:



e:



Para eliminar a ambigüidade favorecendo a interpretação relativa à primeira AD acima, pode-se substituir as regras por:

⟨cmd⟩ → **se1** | **se2**

⟨se1⟩ → **se**⟨exp-rel⟩ **então** ⟨cmd⟩ **senão** ⟨cmd⟩ |
⟨outrocmd⟩

$\langle$se2$\rangle \to$ **se**$\langle$exp-rel$\rangle$ **então** $\langle$cmd$\rangle$ |
se$\langle$exp-rel$\rangle$ **então** $\langle$se1$\rangle$ **senão** $\langle$se2$\rangle$

$\langle$outrocmd$\rangle \to$ outros...

8. Variáveis anuláveis: $\mathcal{A} = \{A, B, P\}$. A GLC resultante é:

   $P \to BPA \mid PA \mid BA \mid BP \mid B \mid A \mid \lambda$

   $A \to \mathtt{a}A \mid \mathtt{a}$

   $B \to B\mathtt{ba} \mid \mathtt{ba}$

   ($P \to P$ não foi acrescida por motivos óbvios.)

9. a) Conjuntos $enc(X)$: $enc(P) = \{P, A, B, C\}$, $enc(A) = \{A, B, C\}$, $enc(B) = \{B\}$ e $enc(C) = \{C\}$. A GLC resultante é:

   $P \to BC \mid \mathtt{b}B \mid \mathtt{b} \mid \mathtt{c}C \mid \mathtt{c}$

   $A \to \mathtt{b}B \mid \mathtt{b} \mid \mathtt{c}C \mid \mathtt{c}$

   $B \to \mathtt{b}B \mid \mathtt{b}$

   $C \to \mathtt{c}C \mid \mathtt{c}$

   b) A variável A é inútil, visto que não existem u e v tais tais que $P \overset{*}{\Rightarrow} uAv$.

10. a) <u>Para FNC</u>: se $n = 0$, o tamanho é 1, correspondendo à derivação $P \Rightarrow \lambda$. Se $n \geq 1$, o tamanho é $2n - 1$. Segue demonstração:

    O tamanho é o número de vértices internos da árvore de derivação. Uma subAD S constituída apenas pelos vértices internos é estritamente binária e tem n folhas (que são os pais dos n terminais). Será provado por indução sobre n que S contém $2n-1$ vértices. Se S contém apenas uma folha, contém apenas um vértice; e $2 \times 1 - 1 = 1$. Seja $n \geq 1$, e suponha, como hipótese de indução que se S contém n folhas, então contém $2n - 1$ vértices. Então, para S com $n + 1$ folhas, se forem retiradas duas folhas filhas do mesmo pai (o que é possível, visto que S é estritamente binária e tem mais de um vértice), a AD resultante tem n folhas e, pela hipótese de indução, tem $2n - 1$ vértices. Recolocando-se as duas folhas, vê-se que S tem $2n - 1 + 2$ vértices, ou seja, $2(n + 1) - 1$ vértices.

    <u>Para FNG</u>: se $n = 0$, o tamanho é 1, correspondendo à derivação $P \Rightarrow \lambda$. Se $n \geq 1$, o tamanho é n, pois cada passo de derivação gera exatamente mais um terminal.

    b) <u>Para FNC</u>: se $n = 0$, a profundidade da AD é 1, correspondendo à derivação $P \Rightarrow \lambda$. Se $n \geq 1$, a profundidade máxima da AD é n, pois a AD de profundidade máxima é conseguida de forma que em cada nível da AD, exceto o 0 (da raiz) e o último (rotulado com um terminal), existem exatamente dois vértices, um rotulado com um terminal, e o outro com um não terminal, filhos do mesmo pai. (O caso em que todos os filhos da esquerda são sempre terminais é suficiente para provar o resultado.)

    <u>Para FNG</u>: o mesmo que para FNC. A AD máxima ocorre quando a gramática é regular (a geração de cada terminal provoca um aumento de nível).

    c) <u>Para FNC</u>: se $n = 0$, a profundidade da AD é 1, correspondendo à derivação $P \Rightarrow \lambda$. Se $n \geq 1$, a profundidade mínima da AD é $\lceil \log_2 n \rceil + 1$, pois a profundidade de uma árvore estritamente binária de n folhas em que o penúltimo nível está completo é $\lceil \log_2 n \rceil$; e a profundidade mínima da AD é um a mais.

    <u>Para FNG</u>: se $n = 0$ ou $n = 1$, a profundidade da AD é 1, correspondendo à derivação $P \Rightarrow \lambda$ ou $P \Rightarrow a$, $a \in \Sigma$. Se $n \geq 2$, a profundidade mínima da AD é 2,

correspondendo a uma derivação da forma

$$P \Rightarrow a_1X_2X_3 \ldots X_n \stackrel{n-1}{\Rightarrow} a_1a_2a_3 \ldots a_n$$

usando-se regras das formas $P \rightarrow a_1X_2X_3 \ldots X_n$ e regras $X_i \rightarrow a_i$.

11. Eliminando-se a recursão direta à esquerda, obtém-se:

$E \rightarrow \mathtt{a}Z \mid \mathtt{a}$

$Z \rightarrow +EZ \mid *EZ \mid +E \mid *E$

12. Algoritmo para obtenção de GLC na FNG:

Entrada: (1) uma GLC $G = (V, \Sigma, R, P)$.
Saída: uma GLC G' equivalente a G, na FNG.
$P' \leftarrow$ uma variável que não pertence a V;
se P ocorre do lado direito de alguma regra **então**
  $V' \leftarrow V \cup \{P'\}$; $R' \leftarrow R \cup \{P' \rightarrow P\}$;
senão
  $V' \leftarrow V$; $R' \leftarrow R$; $P' \leftarrow P$
fimse;
elimine regras λ (algoritmo da Figura 3.22);
elimine regras unitárias (algoritmo da Figura 3.24);
/* aqui pode-se eliminar variáveis inúteis */
Numerar as varíveis de 1 a $|V'|$, sendo 1 o número de P';
/* o número de uma varível X será designado $\#X$ */
para cada $A \in V'$ na ordem de numeração **faça**
  enquanto existe regra $A \rightarrow By \in R'$ tal que $\#B \leq \#A$ **faça**
    se $\#B < \#A$ **então**
      aplique Teorema 24, substituindo B, atualizando R'
    senão
      aplique Teorema 23, atualizando R' e V'
    fimse
  fimenquanto
fimpara;
aplique o teorema 24 na *ordem decrescente* de numeração das varíveis $V \cup \{P'\}$;
aplique o teorema 24 para as regras introduzidas na aplicação do Teorema 23;
providencie a troca de terminais por variáveis no lado direito das regras;
retorne $G' = (V', \Sigma, R', P')$.

13. Eliminando-se regras λ, obtém-se:

$P \rightarrow ABC \mid AB \mid BC \mid B$

$A \rightarrow \mathtt{a}A\mathtt{b} \mid \mathtt{ab}$

$B \rightarrow \mathtt{b}B \mid \mathtt{b}$

$C \rightarrow \mathtt{b}C\mathtt{c} \mid \mathtt{bc}$

Eliminando-se regras de cadeias, obtém-se:

$P \rightarrow ABC \mid AB \mid BC \mid \mathtt{b}B \mid \mathtt{b}$

$A \rightarrow \mathtt{a}A\mathtt{b} \mid \mathtt{ab}$

$B \rightarrow \mathtt{b}B \mid \mathtt{b}$

$C \rightarrow \mathtt{b}C\mathtt{c} \mid \mathtt{bc}$

Aplicando-se o Teorema 24 nas regras P:

$$\begin{aligned}
P &\rightarrow \mathtt{a}A\mathtt{b}BC \,|\, \mathtt{ab}BC \,|\, \mathtt{a}A\mathtt{b}B \,|\, \mathtt{ab}B \,|\, \mathtt{b}BC \,|\, \mathtt{b}C \,|\, \mathtt{b}B \,|\, \mathtt{b} \\
A &\rightarrow \mathtt{a}A\mathtt{b} \,|\, \mathtt{ab} \\
B &\rightarrow \mathtt{b}B \,|\, \mathtt{b} \\
C &\rightarrow \mathtt{b}C\mathtt{c} \,|\, \mathtt{bc}
\end{aligned}$$

Trocando-se terminais por variáveis:

$$\begin{aligned}
P &\rightarrow \mathtt{a}AYBC \,|\, \mathtt{a}YBC \,|\, \mathtt{a}AYB \,|\, \mathtt{a}YB \,|\, \mathtt{b}BC \,|\, \mathtt{b}C \,|\, \mathtt{b}B \,|\, \mathtt{b} \\
A &\rightarrow \mathtt{a}AY \,|\, \mathtt{a}Y \\
B &\rightarrow \mathtt{b}B \,|\, \mathtt{b} \\
C &\rightarrow \mathtt{b}CZ \,|\, \mathtt{b}Z \\
Y &\rightarrow \mathtt{b} \\
Z &\rightarrow \mathtt{c}
\end{aligned}$$

Um APN para a linguagem é $(\{i, f\}, \Sigma, \{P, A, B, C, Y, Z\}, \delta, \{i\}, \{f\})$, em que δ consta das transições:

- $\delta(i, \lambda, \lambda) = \{[f, P]\}$;
- $\delta(f, \mathtt{a}, P) = \{[f, AYBC], [f, YBC], [f, AYB], [f, YB]\}$;
- $\delta(f, \mathtt{b}, P) = \{[f, BC], [f, C], [f, B], [f, \lambda]\}$;
- $\delta(f, \mathtt{a}, A) = \{[f, AY], [f, Y]\}$;
- $\delta(f, \mathtt{b}, B) = \{[f, B], [f, \lambda]\}$;
- $\delta(f, \mathtt{b}, C) = \{[f, CZ], [f, Z]\}$;
- $\delta(f, \mathtt{b}, Y) = \{[f, \lambda]\}$;
- $\delta(f, \mathtt{b}, Z) = \{[f, \lambda]\}$;

14. Seja uma GLC qualquer $G = (V, \Sigma, R, P)$. Um AP que reconhece $L(G)$ é $(\{i, f\}, \Sigma, V \cup \Sigma, \delta, \{i\}, \{f\})$, onde δ é assim determinada:
    - $\delta(i, \lambda, \lambda) = [f, P]$;
    - para cada $X \rightarrow w \in R$, $\delta(f, \lambda, X) = [f, w]$;
    - para cada $a \in \Sigma$, $\delta(f, a, a) = [f, \lambda]$.

15. Uma gramática com duas variáveis:

$$\begin{aligned}
\langle i, \lambda, i\rangle &\rightarrow \lambda \,|\, (\langle i, \mathtt{0}, i\rangle \\
\langle i, \mathtt{0}, i\rangle &\rightarrow (\langle i, \mathtt{0}, i\rangle\langle i, \mathtt{0}, i\rangle \\
\langle i, \mathtt{0}, i\rangle &\rightarrow)\langle i, \lambda, i\rangle
\end{aligned}$$

16. A gramática, já sem variáveis inúteis:

$$\begin{aligned}
P &\rightarrow \langle e_0, \lambda, e_0\rangle \,|\, \langle e_0, \lambda, e_1\rangle \\
\langle e_0, \lambda, e_0\rangle &\rightarrow \lambda \\
\langle e_0, \lambda, e_1\rangle &\rightarrow \mathtt{0}\langle e_0, \mathtt{A}, e_1\rangle \\
\langle e_0, \mathtt{A}, e_1\rangle &\rightarrow \mathtt{0}\langle e_0, \mathtt{A}, e_1\rangle\langle e_1, \mathtt{A}, e_1\rangle \,|\, \mathtt{1}\langle e_1, \lambda, e_1\rangle
\end{aligned}$$

$\langle e_1, \mathtt{A}, e_1\rangle \rightarrow \mathtt{1}\langle e_1, \lambda, e_1\rangle$

$\langle e_1, \lambda, e_1\rangle \rightarrow \lambda$

17. Algoritmo para obter uma GLC a partir de um AP:

Entrada: um AP $M = (E, \Sigma, \Gamma, \delta, I, F)$, $I \neq \emptyset$, $F \neq \emptyset$.
Saída: uma GLC (V, Σ, R, P) que aceita $L(M)$.
$R \leftarrow \{P \rightarrow [i, \lambda, f] \mid i \in I \text{ e } f \in F\}$;
$N \leftarrow \{[i, \lambda, f] \mid i \in I \text{ e } f \in F\}$;
$V \leftarrow \{P\}$;
repita
  $[e, X, d] \leftarrow$ uma variável em N;
  $N \leftarrow N - \{[e, X, d]\}$;
  $V \leftarrow V \cup \{[e, X, d]\}$;
  se $e = d$ e $X = \lambda$ **então**
    $R \leftarrow R \cup \{[e, X, d] \rightarrow \lambda\}$
  fimse; **se** $[e', z] \in \delta(e, a, X)$, $a \in \Sigma \cup \{\lambda\}$ e $X \in \Gamma \cup \{\lambda\}$ **então**
    se $z = \lambda$ **então**
      $R \leftarrow R \cup \{[e, X, d] \rightarrow a[e', \lambda, d]\}$;
      $N \leftarrow N \cup (\{[e', \lambda, d]\} - V)$
    senão
      seja $z = Y_1 Y_2 \ldots Y_n$, $n \geq 1$;
      para cada $d_1, d_2, \ldots, d_{n-1}) \in E^{n-1}$ **faça**
        $R \leftarrow R \cup \{[e, X, d] \rightarrow a[e', Y_1, d_1][d_1, Y_2, d_2] \ldots [d_{n-1}, Y_n, d]\}$;
        $N \leftarrow N \cup (\{[e', Y_1, d_1], [d_1, Y_2, d_2], \ldots, [d_{n-1}, Y_n, d]\} - V)$
      fimpara
    fimse
  fimse;
  se $[e', z] \in \delta(e, a, \lambda)$, $a \in \Sigma \cup \{\lambda\}$ e $X \neq \lambda$ **então**
    se $z = \lambda$ **então**
      $R \leftarrow R \cup \{[e, X, d] \rightarrow a[e', X, d]\}$;
      $N \leftarrow N \cup (\{[e', X, d]\} - V)$
    senão
      seja $z = Y_1 Y_2 \ldots Y_n$, $n \geq 1$;
      para cada $d_1, d_2, \ldots, d_n) \in E^n$ **faça**
        $R \leftarrow R \cup \{[e, X, d] \rightarrow a[e', Y_1, d_1] \ldots [d_{n-1}, Y_n, d_n][d_n, X, d]\}$;
        $N \leftarrow N \cup (\{[e', Y_1, d_1], \ldots, [d_{n-1}, Y_n, d_n], [d_n, X, d]\} - V)$
      fimpara
    fimse
  fimse;
até $N = \emptyset$;
retorne (V, Σ, R, P).

18. Será usado o algoritmo da questão anterior. Imediatamente antes do comando **repita** tem-se:

- $V = \{P\}$
- $N = \{[0, \lambda, 1]\}$
- $R = \{P \rightarrow [0, \lambda, 1]\}$

Ao atingir o final do **repita** pela primeira vez:

- $V = \{P, [0, \lambda, 1]\}$
- $N = \{[0, \mathtt{A}, 1]\}$
- $R = \{ \quad P \rightarrow [0, \lambda, 1],$
  $[0, \lambda, 1] \rightarrow \mathtt{a}[0, \mathtt{A}, 1]\}$

Ao atingir o final do **repita** pela segunda vez:

- $V = \{P, [0, \lambda, 1], [0, \mathtt{A}, 1]\}$
- $N = \{[1, \mathtt{B}, 1], [0, \mathtt{A}, 0], [1, \mathtt{A}, 1]\}$
- $R = \{$
$$\begin{array}{rcl} P & \to & [0, \lambda, 1], \\ [0, \lambda, 1] & \to & \mathtt{a}[0, \mathtt{A}, 1], \\ [0, \mathtt{A}, 1] & \to & \mathtt{b}[1, \mathtt{B}, 1] \mid \\ & & \mathtt{a}[0, \mathtt{A}, 0][0, \mathtt{A}, 1] \mid \\ & & \mathtt{a}[0, \mathtt{A}, 1][1, \mathtt{A}, 1]\} \end{array}$$

A variável $[0, \mathtt{A}, 0]$ é inútil, pois é impossível, no AP, partir do estado 0 com `A` na pilha, ler uma palavra e voltar ao estado 0, com a pilha o vazia. Assim, ela será eliminada de N e de R, embora o algoritmo não faça isso, obtendo-se:

- $V = \{P, [0, \lambda, 1], [0, \mathtt{A}, 1]\}$
- $N = \{[1, \mathtt{B}, 1], [1, \mathtt{A}, 1]\}$
- $R = \{$
$$\begin{array}{rcl} P & \to & [0, \lambda, 1], \\ [0, \lambda, 1] & \to & \mathtt{a}[0, \mathtt{A}, 1], \\ [0, \mathtt{A}, 1] & \to & \mathtt{b}[1, \mathtt{B}, 1] \mid \\ & & \mathtt{a}[0, \mathtt{A}, 1][1, \mathtt{A}, 1]\} \end{array}$$

Com tal atualização, ao atingir o final do **repita** pela terceira vez:

- $V = \{P, [0, \lambda, 1], [0, \mathtt{A}, 1], [1, \mathtt{B}, 1]\}$
- $N = \{[1, \mathtt{A}, 1], [1, \lambda, 1]\}$
- $R = \{$
$$\begin{array}{rcl} P & \to & [0, \lambda, 1], \\ [0, \lambda, 1] & \to & \mathtt{a}[0, \mathtt{A}, 1], \\ [0, \mathtt{A}, 1] & \to & \mathtt{b}[1, \mathtt{B}, 1] \mid \\ & & \mathtt{a}[0, \mathtt{A}, 1][1, \mathtt{A}, 1], \\ [1, \mathtt{B}, 1] & \to & \mathtt{c}[1, \lambda, 1]\} \end{array}$$

Ao atingir o final do **repita** pela quarta vez:

- $V = \{P, [0, \lambda, 1], [0, \mathtt{A}, 1], [1, \mathtt{B}, 1], [1, \mathtt{A}, 1]\}$
- $N = \{[1, \lambda, 1]\}$
- $R = \{$
$$\begin{array}{rcl} P & \to & [0, \lambda, 1], \\ [0, \lambda, 1] & \to & \mathtt{a}[0, \mathtt{A}, 1], \\ [0, \mathtt{A}, 1] & \to & \mathtt{b}[1, \mathtt{B}, 1] \mid \\ & & \mathtt{a}[0, \mathtt{A}, 1][1, \mathtt{A}, 1], \\ [1, \mathtt{B}, 1] & \to & \mathtt{c}[1, \lambda, 1], \\ [1, \mathtt{A}, 1] & \to & \mathtt{b}[1, \mathtt{B}, 1]\} \end{array}$$

Ao final do **repita** pela quinta vez:

- $V = \{P, [0, \lambda, 1], [0, \mathtt{A}, 1], [1, \mathtt{B}, 1], [1, \mathtt{A}, 1], [1, \lambda, 1]\}$
- $N = \{\}$

- $$\begin{array}{rrcl} R = \{ & P & \to & [0,\lambda,1], \\ & [0,\lambda,1] & \to & \mathtt{a}[0,\mathtt{A},1], \\ & [0,\mathtt{A},1] & \to & \mathtt{b}[1,\mathtt{B},1] \mid \\ & & & \mathtt{a}[0,\mathtt{A},1][1,\mathtt{A},1], \\ & [1,\mathtt{B},1] & \to & \mathtt{c}[1,\lambda,1], \\ & [1,\mathtt{A},1] & \to & \mathtt{b}[1,\mathtt{B},1], \\ & [1,\lambda,1] & \to & \lambda\} \end{array}$$

19. Como sugerido no final do Teorema 27, será provado:

$$[e,X,d_n] \stackrel{*}{\Rightarrow} w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n] \text{ se, e somente se, } [e,w,X] \stackrel{*}{\vdash} [d_0,\lambda,Y_1\dots Y_n]$$

para todo $e, d_0, \dots d_n \in E$, $X \in \Gamma \cup \{\lambda\}$, $Y_1, \dots Y_n \in \Gamma$ e $w \in \Sigma^*$.

Assim, sejam $e, d_0, \dots d_n \in E$, $X \in \Gamma \cup \{\lambda\}$, $Y_1, \dots Y_n \in \Gamma$ e $w \in \Sigma^*$ arbitrários. Primeiro, será mostrado por indução sobre k que

$$\text{se } [e,X,d_n] \stackrel{k}{\Rightarrow} w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n] \text{ então } [e,w,X] \stackrel{*}{\vdash} [d_0,\lambda,Y_1\dots Y_n].$$

Inicialmente, suponha que $[e,X,d_n] \stackrel{0}{\Rightarrow} w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n]$. Segue-se que $w=\lambda$, $n=1$, $X=Y_1$, $d_0=e$ e, portanto, $[e,\lambda,X] \stackrel{0}{\vdash} [e,\lambda,X]$, como requerido. Seja $k \geq 0$. Suponha, como hipótese de indução, que

$$\text{se } [e,X,d_n] \stackrel{k}{\Rightarrow} w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n] \text{ então } [e,w,X] \stackrel{*}{\vdash} [d_0,\lambda,Y_1\dots Y_n].$$

Suponha que $[e,X,d_n] \stackrel{k+1}{\Rightarrow} w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n]$. Considerando apenas derivações mais à esquerda (não há perda de generalidade), tem-se que:

$$[e,X,d_n] \stackrel{k}{\Rightarrow} x[c,Z,d_i][d_i,Y_{i+1},d_{i+1}]\cdots[d_{n-1},Y_n,d_n] \Rightarrow w[d_0,Y_1,d_1]\cdots[d_{n-1},Y_n,d_n]$$

sendo que $w = xa$ e a última regra aplicada foi da forma:

$$[c,Z,d_i] \to a[d_0,Y_1,d_1]\cdots[d_{i-1},Y_i,d_i].$$

Pela hipótese de indução, $[e,x,X] \stackrel{*}{\vdash} [c,\lambda,ZY_{i+1}\dots Y_n]$. Considera-se cada um dos formatos possíveis para a última regra aplicada:

- $\underline{[c,\lambda,c] \to \lambda}$. Neste caso, tem-se: $a=\lambda$, $Z=\lambda$, $i=0$ e $c=d_0$. Logo, $[c,\lambda,ZY_{i+1}\dots Y_n] \stackrel{0}{\vdash} [d_0,\lambda,Y_1\dots Y_n]$ e, portanto,

  $$[e,w,X] \stackrel{*}{\vdash} [c,\lambda,ZY_{i+1}\dots Y_n] \stackrel{0}{\vdash} [d_0,\lambda,Y_1\dots Y_n].$$

  .

- $\underline{[c,Z,d] \to a[e',\lambda,d]}$. Neste caso, tem-se: $i=1$, $d_0=e'$, $d_1=d$ e $Y_1=\lambda$. E, como tal regra surge devido à transição $[e',\lambda] \in \delta(c,a,Z)]$, segue-se que $[c,a,Z] \vdash [e',\lambda,\lambda]$ e, portanto,

  $$[e,w,X] \stackrel{*}{\vdash} [c,a,ZY_{i+1}\dots Y_n] \vdash [d_0,\lambda,Y_1\dots Y_n].$$

  .

- Regra $[e, X, d_n] \rightarrow a[e', Y_1, d_1] \ldots [d_{n-1}, Y_n, d_n]$. Aqui tem-se: $w = a$ e $d_0 = e'$. E, como tal regra surge devido à transição $[e', Y_1 \ldots Y_n] \in \delta(e, a, X)]$, $[e, a, X] \overset{*}{\vdash} [e', \lambda, Y_1 \ldots Y_n]$.
- Regra $[e, Z, d] \rightarrow a[e', Z, d]$, $Z \in \Gamma$. Aqui tem-se: $w = a$, $n = 1$, $d_0 = e'$, $d_1 = d$ e $Y_1 = Z$. E, como tal regra surge devido à transição $[e', \lambda] \in \delta(e, a, \lambda)]$, $[e, a, Z] \vdash [e', \lambda, Z]$.
- Regra $[e, Z, c_{m+1}] \rightarrow a[e', Y_1, c_1] \ldots [c_{m-1}, Y_m, c_m][c_m, Z, c_{m+1}]$. Aqui tem-se: $w = a$, $d_0 = e'$, $c_i = d_i$ para $i = 1, \ldots, m$, $n = m + 1$ e $Y_n = Z$. E, como tal regra surge devido à transição $[e', Y_1 \ldots Y_m] \in \delta(e, a, \lambda)]$, $[e, a, Z] \overset{*}{\vdash} [e', \lambda, Y_1 \ldots Y_m Z]$.

3.5 Linguagens Livres do Contexto: Propriedades

1. a) Seja $L = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w$ tem número par de $\mathtt{0}$s$\}$. Seja uma palavra arbitrária $z \in L$ de pelo menos dois símbolos. Há dois casos a considerar:

   Caso 1: z não contém $\mathtt{0}$s. Então, tome $u = v = w = y = \lambda$ e $x = z$. Isso satisfaz as 4 condições do lema.

   Caso 2: z contém $\mathtt{0}$s. Então, tome $u = \mathtt{1}^m$, sendo m o número de $\mathtt{1}$s antes do primeiro $\mathtt{0}$, $v = \mathtt{0}$, $w = \mathtt{1}^n$, sendo n o número de $\mathtt{1}$s entre o primeiro e o segundo $\mathtt{0}$, $x = \mathtt{0}$ e y contendo o resto de z. Isso satisfaz as 4 condições do lema.

   Conclusão: L satisfaz o LB para LLCs.

   b) Seja $L = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w$ tem número igual de $\mathtt{0}$s e$\mathtt{1}$s$\}$. Seja uma palavra arbitrária $z \in L$ de pelo menos dois símbolos. Tal palavra contém pelo menos um $\mathtt{0}$ e um $\mathtt{1}$ e é possível fazer $z = u\mathtt{0}w\mathtt{1}y$ (no caso, $v = \mathtt{0}$ e $x = \mathtt{1}$) ou $z = u\mathtt{1}w\mathtt{0}y$ (no caso, $v = \mathtt{1}$ e $x = \mathtt{0}$), sendo u, v, w, x e y como no LB. Em qualquer caso, as 4 condições do lema são satisfeitas. Conclusão: L satisfaz o LB para LLCs.

2. a) Suponha que $L = \{\mathtt{a}^{n^2} \mid n \geq 0\}$ é LLC. Seja k a constante do LB e $z = \mathtt{a}^{k^2}$. Sejam u, v, w, x e y tais que $uvwxy = z$, $|vx| > 0$ e $|vwx| \leq k$. Neste caso,

   - $|uv^2wx^2y| > k^2$, pois $|vx| > 0$; e
   - $|uv^2wx^2y| = |z| + |vx| \leq k^2 + k$, já que $|vwx| \leq k$, e por sua vez $k^2 + k < k^2 + 2k + 1 = (k + 1)^2$; ou seja, $|uv^2wx^2y| < (k + 1)^2$.

   Como $0 < |uv^2wx^2y| < (k + 1)^2$, o número de símbolos de uv^2wx^2y não é quadrado perfeito e, assim, $uv^2wx^2y \notin L$, contradizendo o LB. Logo, L não é LL

   b) Suponha que $L = \{\mathtt{a}^n\mathtt{b}^{2n}\mathtt{a}^n \mid n \geq 0\}$ é LLC. Seja k a constante do LB e $z = \mathtt{a}^k\mathtt{b}^{2k}\mathtt{a}^k$. Sejam u, v, w, x e y tais que $uvwxy = z$, $|vx| > 0$ e $|vwx| \leq k$. Tem-se dois casos a considerar:

   (a) vx contém algum $\mathtt{a}$ do prefixo de n $\mathtt{a}$s. Então, como $|vwx| \leq k$, uv^0wx^0y é da forma $\mathtt{a}^{k-p}\mathtt{b}^{2k-q}\mathtt{a}^k$, em que $1 \leq p \leq k$ (pois $|vx| > 0$) e $0 \leq q < k$. Logo, $uv^0wx^0y \notin L$.

   (b) vx não contém $\mathtt{a}$ do prefixo de n $\mathtt{a}$s. Então, como $|vwx| \leq k$, uv^0wx^0y é da forma $\mathtt{a}^k\mathtt{b}^{2k-q}\mathtt{a}^{k-p}$, em que $0 \leq p \leq k$, $0 \leq q \leq k$ e $p + q > 0$ (pois $|vx| > 0$). Logo, $uv^0wx^0y \notin L$.

   Como $0 < |uv^0wx^0y| \notin L$ em qualquer caso, contradiz-se o LB. Logo, L não é LLC.

   c) Suponha que $L = \{\mathtt{a}^n\mathtt{b}^k\mathtt{c}^n\mathtt{d}^k \mid k, n > 0\}$ é LLC. Seja k a constante do LB e $z = \mathtt{a}^k\mathtt{b}^k\mathtt{c}^k\mathtt{d}^k$. Sejam u, v, w, x e y tais que $uvwxy = z$, $|vx| > 0$ e $|vwx| \leq k$. Pelo LB, $uv^iwx^iy \in L$ para todo $i \geq 0$. Como $|vx| > 0$, tem-se 4 casos a considerar:

1. vx contém as. Como $|vwx| \leq k$, não há cs em vx. Assim, $uv^2wx^2y \notin L$.
2. vx contém bs. Como $|vwx| \leq k$, não há ds em vx. Assim, $uv^2wx^2y \notin L$.
3. vx contém cs. Como $|vwx| \leq k$, não há as em vx. Assim, $uv^2wx^2y \notin L$.
4. vx contém ds. Como $|vwx| \leq k$, não há bs em vx. Assim, $uv^2wx^2y \notin L$.

Contradição. Portanto, L não é LLC.

3. a) $\overline{L_1}$ é LLC. Uma GLC para $\overline{L_1}$:
   $P \rightarrow \mathtt{a}P\mathtt{b} \,|\, A \,|\, B \,|\, \mathtt{b}X\mathtt{a}$
   $A \rightarrow \mathtt{a}A \,|\, \mathtt{a}$
   $B \rightarrow \mathtt{a}B \,|\, \mathtt{b}$
   $X \rightarrow \mathtt{a}X \,|\, \mathtt{b}X \,|\, \lambda$

   b) $L_1 \cap L_2$ é LLC, pois L_1 é LLC e L_2 é regular.

   c) Como L_2 é regular, $\overline{L_2}$ é regular. E como L_1 é LLC, $L_1 \cap \overline{L_2}$ é LLC.

4. Não. Seja X uma linguagem qualquer não LLC sobre o alfabeto Σ. $X \subset \Sigma^*$ e Σ^* é LLC.

5. a) É LLC. Uma GLC para a linguagem:
   $P \rightarrow \mathtt{a}P\mathtt{b}P \,|\, \mathtt{b}P\mathtt{a}P \,|\, \mathtt{c}P \,|\, \lambda$

   b) Seja L a linguagem em questão. Suponha que L seja LLC. Então:

$$L \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^* = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \,|\, n \geq 0\}$$

   Como as LLCs são fechadas sob interseçcão com linguagens regulares, tem-se uma contradição, visto que $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \,|\, n \geq 0\}$ não é LLC. Portanto, L não é LLC.

   c) Seja L a linguagem em questão. Suponha que L seja LLC. Então:

$$L \cap \{\mathtt{c}\}^* = \{\mathtt{c}^{n^2} \,|\, n \geq 0\}$$

   Como as LLCs são fechadas sob interseçcão com linguagens regulares, tem-se uma contradição, visto que $\{\mathtt{c}^{n^2} \,|\, n \geq 0\}$ não é LLC. Portanto, L não é LLC.

6. Seja L uma LLC qualquer e $G = (V, \Sigma, R, P)$ uma GLC que a reconhece. A GLC $G' = (V, \Sigma, R', P)$, onde $R' = \{X \rightarrow v^R \,|\, X \rightarrow v \in R\}$ reconhece L^R. Isto segue do fato que para todo $n \geq 0$ e toda forma sentencial $u \in (V \cup \Sigma)^*$, $P \overset{n}{\Rightarrow}_G u$ se, e somente se, $P \overset{n}{\Rightarrow}_{G'} u^R$. Será mostrado por indução sobre n, que se $P \overset{n}{\Rightarrow}_G u$ então $P \overset{n}{\Rightarrow}_{G'} u^R$ para todo $u \in (V \cup \Sigma)^*$. A demonstração da recíproca pode ser feita de forma análoga. Para $n = 0$, tem-se, por definição, que o único u tal que $P \overset{0}{\Rightarrow}_G u$ é $u = P$; e $P \overset{0}{\Rightarrow}_{G'} P = P^R$. Seja n um número natural arbitrário e suponha, como hipótese de indução, que se $P \overset{n}{\Rightarrow}_G u$ então $P \overset{n}{\Rightarrow}_{G'} u^R$ para todo $u \in (V \cup \Sigma)^*$. Suponha que $P \overset{n+1}{\Rightarrow}_G z$. Basta, então, mostrar que $P \overset{n+1}{\Rightarrow}_{G'} z^R$. Por defincção, existem $x, y \in (V \cup \Sigma)^*$ e $Y \rightarrow s \in R$ tais que:

$$P \overset{n}{\Rightarrow}_G xYy \Rightarrow_G xsy.$$

   Pela hipótese de indução, $P \overset{n}{\Rightarrow}_{G'} (xYy)^R = y^RYx^R$. E como, por definição de G', $Y \rightarrow s^R \in R'$, segue-se que:

$$P \overset{n}{\Rightarrow}_{G'} y^RYx^R \Rightarrow_{G'} y^Rs^Rx^R = (xsy)^R = z^R.$$

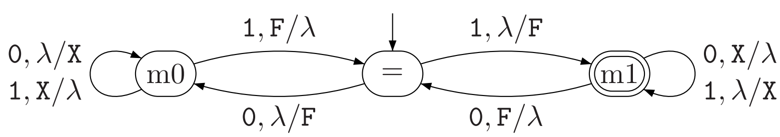
   Portanto, $P \overset{n+1}{\Rightarrow}_{G'} z^R$, como requerido.

7. Suponha que L é uma LLC e R uma linguagem regular. Como as linguagens regulares são fechadas sob complementação, $\overline{R}$ é regular; e como as LLCs são fechadas sob interseção com linguagens regulares, $L \cap \overline{R}$ é LLC. Mas $L \cap \overline{R} = L - R$. Logo, se L é uma LLC e R uma linguagem regular, então $L - R$ é uma LLC.

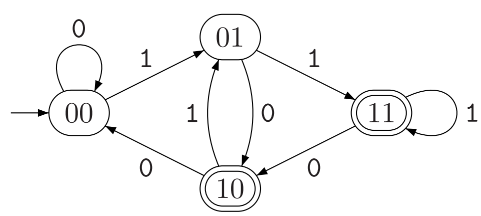
8. Seja L uma LLC arbitrária de alfabeto Σ. Como as LLCs não são fechadas sob complementação, $\overline{L} = \Sigma^* - L$ pode não ser LLC. Mas Σ é LLC! Portanto, as LLCs não são fechadas sob diferença.

9.
   - Seja $h : \Sigma \to \Delta^*$ um homomorfismo e G uma GLC na FNC para uma certa LLC L sobre Σ. Para obter uma GLC G' que aceite $h(L)$ sobre Δ, basta substituir cada regra de G da forma $X \to a$, onde a é um terminal, por $X \to h(a)$. Pode-se provar por indução sobre o tamanho das derivações que w é derivável em G se, e somente se, $h(w)$ é derivável em G'.
   - Seja $s : \Sigma \to \mathcal{P}(\Delta^*) - \{\emptyset\}$ uma substituição e G uma GLC na FNC para uma certa LLC L sobre Σ. Como $s(a)$, para $a \in \Sigma$, é uma linguagem regular, seja $Gs(a)$ uma gramática regular que gere $s(a)$ e que não contenha variáveis em comum com G nem com $Gs(b)$ para $b \neq a$. Para obter uma GLC G' que aceite $s(L)$ sobre Δ, basta substituir cada regra de G da forma $X \to a$, onde a é um terminal, por $X \to P_a$, onde P_a é o símbolo de partida de $Gs(a)$, e acrescentar às suas regras as de todas as gramáticas $Gs(a)$. Pode-se provar por indução sobre o tamanho das derivações que w é derivável em G se, e somente se, as palavras em $s(w)$ são deriváveis em G'.

10. APD para $L_1 = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid w \text{ tem mais } \mathtt{1}\text{s que } \mathtt{0}\text{s}\}$ com reconhecimento por estado final:



AFD para $L_2 = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid |w| \geq 2 \text{ e o penúltimo símbolo de } w \text{ é } \mathtt{1}\}$:



Um APD para $L_1 \cap L_2$ que reconhece por estado final usando o método requerido seria

$$(\{=, \mathrm{m0}, \mathrm{m1}\} \times \{00, 01, 10, 11\}, \{\mathtt{0}, \mathtt{1}\}, \{\mathtt{F}, \mathtt{X}\}, \delta, \{[=, 00]\}, \{[\mathrm{m1}, 10], [\mathrm{m1}, 11]\})$$

em que δ é dada por:

$\delta([=, 00], 0, \lambda) = [[\mathrm{m0}, 00], \mathtt{F}]$,
$\delta([=, 00], 1, \lambda) = [[\mathrm{m1}, 01], \mathtt{F}]$,
$\delta([=, 01], 0, \lambda) = [[\mathrm{m0}, 10], \mathtt{F}]$,
$\delta([=, 01], 1, \lambda) = [[\mathrm{m1}, 11], \mathtt{F}]$,
$\delta([=, 10], 0, \lambda) = [[\mathrm{m0}, 00], \mathtt{F}]$,

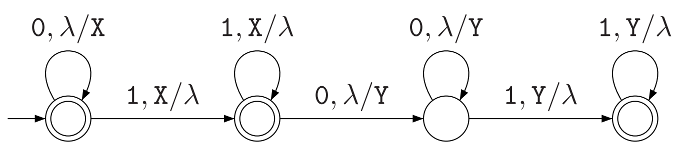
$\delta([=,10],1,\lambda) = [[\text{m1},01],\mathtt{F}]$,
$\delta([=,11],0,\lambda) = [[\text{m0},10],\mathtt{F}]$,
$\delta([=,11],1,\lambda) = [[\text{m1},11],\mathtt{F}]$,

$\delta([\text{m0},00],0,\lambda) = [[\text{m0},00],\mathtt{X}]$,
$\delta([\text{m0},00],1,\mathtt{X}) = [[\text{m0},01],\lambda]$,
$\delta([\text{m0},00],1,\mathtt{F}) = [[=,01],\lambda]$,
e assim por diante. . .

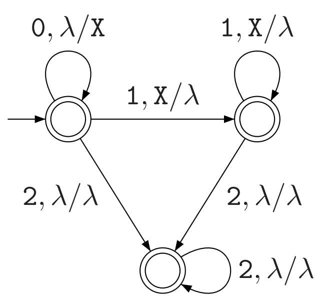
11. Seja k a constante do LB. Será mostrado que $L(M)$ é infinita se, e somente se, M aceita alguma palavra de tamanho k a $2k-1$. Assim, para testar se $L(M)$ é infinita, basta verificar se M aceita alguma palavra de tamanho k a $2k-1$; se aceita, $L(M)$ é infinita, caso contrário é finita.
($\leftarrow$) Se M aceita alguma palavra z de tamanho k a $2k-1$, pelo LB existem u, v, w, x e y tais $z = uvwxy$, $|vx| > 0$, $|vwx| \leq k$ e $uv^iwx^iy \in L(M)$ $\forall i \geq 0$. Portanto, $L(M)$ é infinita.
($\rightarrow$) Suponha que $L(M)$ é infinita. Seja z a *menor* palavra tal que $|z| \geq k$ em $L(M)$. Suponha que $|z| \geq 2k$. Pelo LB, existem u, v, w, x e y tais $w = uvwxy$, $|vx| > 0$, $|vwx| \leq k$ e $uv^iwx^iy \in L(M)$ $\forall i \geq 0$. Neste caso, $uwy \in L(M)$; mas, como $|z| \geq 2k$, $|uwy| \geq k$, o que contradiz a suposição de que z é a *menor* palavra de $L(M)$ tal que $|z| \geq k$. Portanto, $|z| < 2k$ e, assim, M aceita alguma palavra de tamanho k a $2k-1$.

3.6 Exercícios

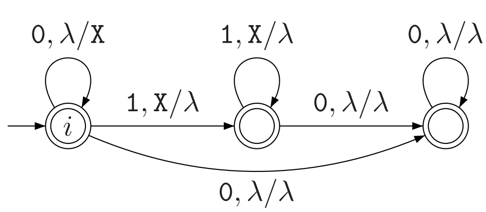
1. **a)** Um APD para $\{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}\{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}$:



b) Um APD para $\{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^k \mid n, k \geq 0\}$:

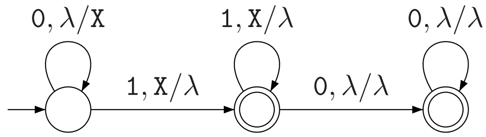


c) Um APN para $\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^k \mid n, k \geq 0\}$:

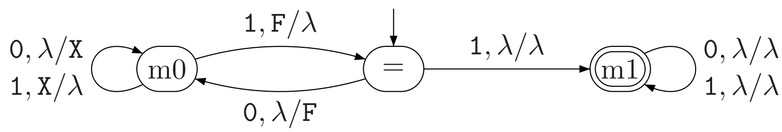


Não há como saber, no estado i, se uma seqüência de `0`s será seguida de `1`s (caso em que $n > 0$ e se deve contar os `0`s) ou não (caso em que $n = 0$ e $k > 0$ e não se deve contar os `0`s). No entanto, existe APD para esta linguagem que reconhece *por estado final*.

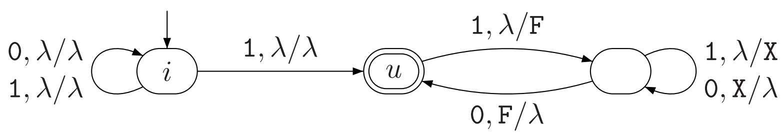
d) Um APD para $\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^k \mid n \geq 1 \text{ e } k \geq 0\}$:



e) Um APD para $\{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w$ tem algum prefixo com mais `1`s que `0`s$\}$:

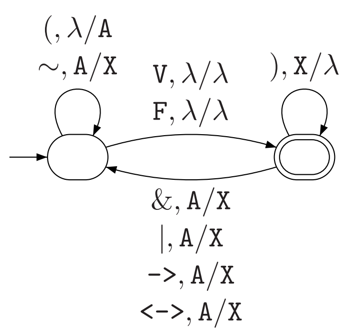


f) Um APN para $\{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w$ tem algum sufixo com mais `1`s que `0`s$\}$:

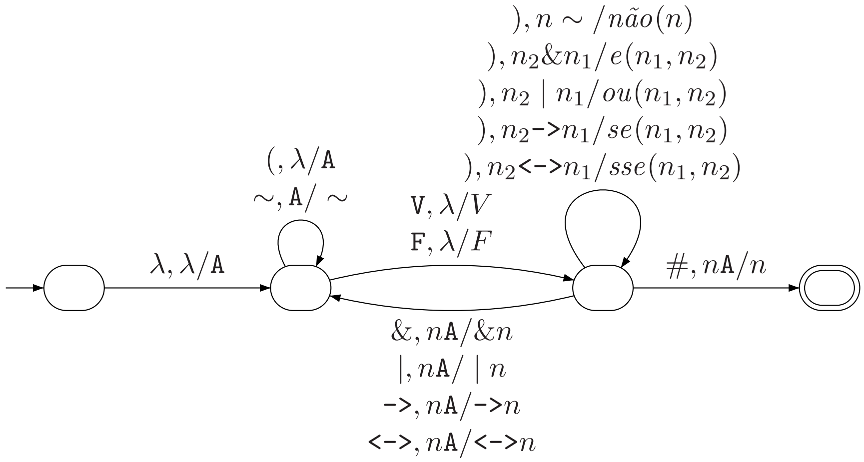


Não há como saber, no estado i, se um `1` lido é o *último* a partir do qual o número de `0`s e `1`s é sempre o mesmo (caso em que deve ser tomada a transição para o estado u) ou não (caso em que deve ser tomado o *loop*).

2. **a)** Um APD para EB:



b) Segue um avaliador de EBs baseado no APD anterior. Supõe-se a existência de funções *não*, *e*, *ou*, *se* e *sse* para as operações lógicas óbvias.



3. Sejam dois APNs $M_1 = (E_1, \Sigma_1, \Gamma_1, \delta_1, I_1, F_1)$ e $M_1 = (E_2, \Sigma_2, \Gamma_2, \delta_2, I_2, F_2)$ tais que $E_1 \cap E_2 = \emptyset$ e $\Gamma_1 \cup \Gamma_2 = \emptyset$, sem perda de generalidade, visto que estados podem ser renomeados sem alterar a linguagem aceita.

   a) APN para $L(M_1) \cup L(M_2)$: $(E_1 \cup E_2, \Sigma_1 \cup \Sigma_2, \Gamma_1 \cup \Gamma_2, \delta_1 \cup \delta_2, I_1 \cup I_2, F_1 \cup F_2)$. Neste caso, o fato que $\Gamma_1 \cup \Gamma_2 = \emptyset$ não é relevante.

   b) APN para $L(M_1)L(M_2)$: $(E_1 \cup E_2, \Sigma_1 \cup \Sigma_2, \Gamma_1 \cup \Gamma_2, \delta_3, I_1, F_2)$, em que δ_3 consta das transições δ_1 e δ_2 mais as transições $\delta_3(e, \lambda, \lambda) = \{[e', \lambda] \mid e' \in I_2\}$ para cada $e \in F_1$. Neste caso, o fato que $\Gamma_1 \cup \Gamma_2 = \emptyset$ é fundamental.

   c) APN para $L(M_1)^*$: $(E_1 \cup \{i\}, \Sigma_1, \Gamma_1 \cup \{\#\}, \delta_*, \{i\}, \{i\})$, em que $i \notin E_1$, $\# \notin \Gamma_1$ e δ_* consta de δ_1 mais as transições:

   - $\delta_*(i, \lambda, \lambda) = \{[e, \#] \mid e \in I_1\}$, e
   - $\delta_*(e, \lambda, \#) = \{[i, \lambda]\}$ para cada $e \in F_1$.

4. Seja um AP $M = (E, \Sigma, \Gamma, \delta, I, F)$ cuja pilha pode conter, no máximo, n símbolos. Um AFN que reconhece $L(M)$ é $M' = (E', \Sigma, \delta', I', F')$, onde:

   - $E' = E \times \{z \in \Gamma^* \mid |z| \leq n\}$;
   - $I' = I \times \{\lambda\}$;
   - $F' = F \times \{\lambda\}$; e
   - para cada transição de M, $[e', z] \in \delta(e, a, X)$, tem-se transições em M' da forma $[e', zy] \in \delta'([e, Xy], a)$ para cada $y \in \Gamma^*$ tal que $|Xy| \leq n$ e $|zy| \leq n$; estas são as únicas transições de M'.

5. Seja um AFNλ $M = (E, \Sigma, \delta, I, F)$. Um APN de dois estados que reconhece $L(M)$ seria $P = (\{i, f\}, \Sigma, E, \delta_P, \{i\}, \{f\})$, sendo as transições constituídas de:

   - $\delta_P(i, \lambda, \lambda) = \{[e, f] \mid e \in I\}$;
   - $[f, e'] \in \delta_P(f, a, e)$ para todo $e' \in \delta(e, a)$; e
   - $[f, \lambda] \in \delta_P(f, \lambda, e)$ para todo $e \in F$.

6. Só há fechamento com relação a complementação. Seguem ontra-exemplos para os outros casos:

a) União: $\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ e $\{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$ são reconhecíveis por APDs por estado final, mas sua união não é.

b) Interseção: $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^k \mid n, k \geq 0\}$ e $\{\mathtt{a}^n\mathtt{b}^k\mathtt{c}^k \mid n, k \geq 0\}$ são reconhecíveis por APDs por estado final, mas sua interseção não é.

7. Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$. Um APN M' tal que $L_V(M') = L(M)$ é $M' = (E \cup \{i, f\}, \Sigma, \Gamma \cup \{\#\}, \delta', \{i\})$, em que $i, f \notin E$, $\# \notin \Gamma$ e δ' consta das transição de δ mais:

- $\delta'(i, \lambda, \lambda) = \{[e, \#] \mid e \in I\}$;
- $\delta'(e, \lambda, \#) = \{[f, \lambda]\}$ para todo $e \in F$.

Agora, seja um APN $M = (E, \Sigma, \Gamma, \delta, I)$. Um APN M' tal que $L(M') = L_V(M)$ é $M' = (E, \Sigma, \Gamma, \delta, I, E)$.

8. Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$. Um APN M' tal que $L_F(M') = L(M)$ é $M' = (E \cup \{i, f\}, \Sigma, \Gamma \cup \{\#\}, \delta', \{i\}, \{f\})$, em que $i, f \notin E$, $\# \notin \Gamma$ e δ' consta das transição de δ mais:

- $\delta'(i, \lambda, \lambda) = \{[e, \#] \mid e \in I\}$;
- $\delta'(e, \lambda, \#) = \{[f, \lambda]\}$ para todo $e \in F$.

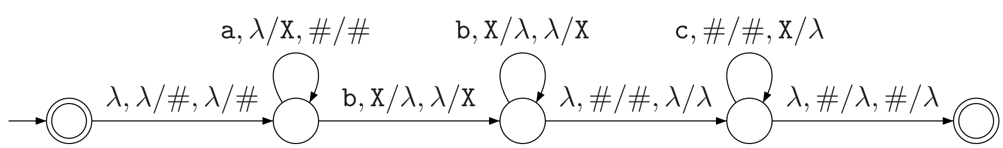
9. APD sem transições λ.

10. Seja $M = (E, \Sigma, \Gamma, \delta, i, F)$ um APD sem transições λ. Um APD M' que aceita $L_F(M)\{\#\}$ por pilha vazia é $M' = (E \cup \{i', v\}, \Sigma \cup \{\#\}, \Gamma \cup \{\$\}, \delta', i')$, em que $\$ \notin \Gamma$, $i' \notin E$ e δ' consta das transição de δ mais:

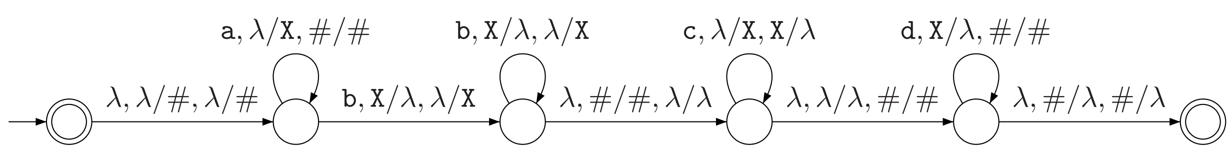
- $\delta'(i', \lambda, \lambda) = [i, \$]$;
- para cada $e \in F$, $\delta'(e, \#, \lambda) = [v, \lambda]$;
- $\delta'(v, \lambda, X) = [v, \lambda]$ para todo $X \in \gamma$;
- $\delta'(v, \lambda, \$) = [v, \lambda]$.

Observe que, como M não tem trasições λ, uma transição $\delta'(e, \#, \lambda) = [v, \lambda]$ não é compatível com nenhuma outra que emane de e.

11. a) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$.



b) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n\mathtt{d}^n \mid n \geq 0\}$.

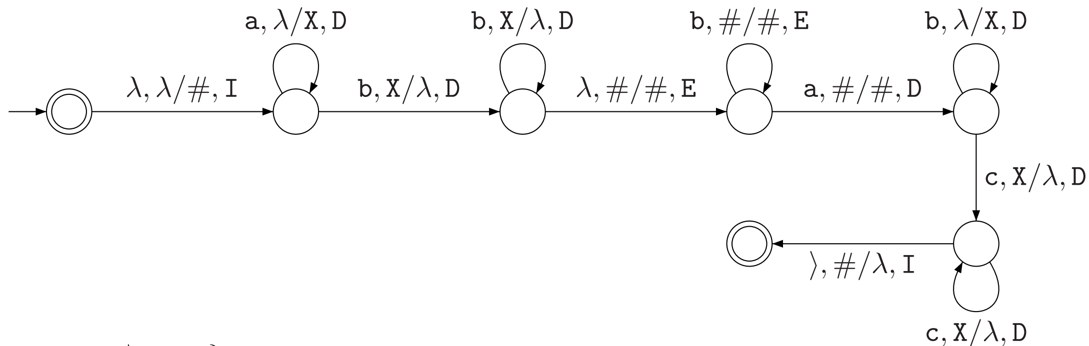


12. Um autômato de pilha com fita bidirecional é uma óctupla $(E, \Sigma, \Gamma, \langle, \rangle, \delta, I, F)$, em que:

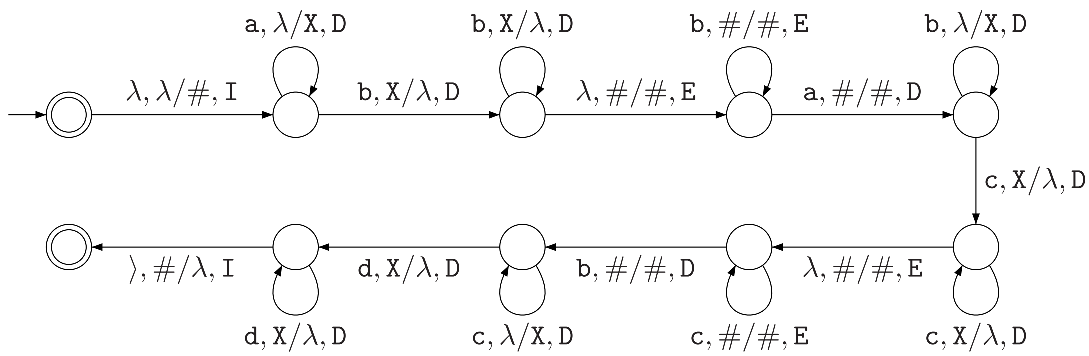
- E, Σ, Γ e F são como em APs;
- δ, a função de transição, é uma função parcial de $E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\})$ para D, sendo D constituído dos subconjuntos finitos de $E \times \Gamma^* \times \{\mathtt{D}, \mathtt{E}, \mathtt{I}\}$;
- I, um subconjunto de E, é o conjunto de estados iniciais.

As opções `D`, `E` e `I` especificam que o cabeçote de leitura é movido para a direita, esquerda ou fica imóvel, respectivamente. Elas serão escritas por último, separadas do restante do rótulo da transição por uma vírgula.

a) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$.



b) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n\mathtt{d}^n \mid n \geq 0\}$.



13. a) $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^{2(m+n)} \mid m, n \geq 0\}$.

$$P \rightarrow \mathtt{a}P\mathtt{cc} \mid B$$
$$B \rightarrow \mathtt{b}B\mathtt{cc} \mid \lambda$$

b) $\{w \in \{a, b\}^* \mid$ o número de as em w é o dobro do número de bs$\}$.

$$P \rightarrow \mathtt{a}P\mathtt{a}P\mathtt{b}P \mid \mathtt{a}P\mathtt{b}P\mathtt{a}P \mid \mathtt{b}P\mathtt{a}P\mathtt{a}P \mid \lambda$$

c) $\{w \in \{a, b\}^* \mid$ o número de as em w é diferente do número de bs$\}$.

$$P \rightarrow A \mid B$$
$$A \rightarrow \mathtt{a}A \mid IA \mid \mathtt{a}I$$
$$B \rightarrow \mathtt{b}B \mid IB \mid \mathtt{b}I$$
$$I \rightarrow \mathtt{a}I\mathtt{b}I \mid \mathtt{b}I\mathtt{a}I \mid \lambda$$

Observe que $A \stackrel{*}{\Rightarrow} (a + I)^*aI$ e $I \stackrel{*}{\Rightarrow} w$, em que w pode ser qualquer palavra com número igual de `a`s e `b`s.

d) $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid n > m$ ou $n > k\}$.

$$P \rightarrow XC \mid AY$$

$$X \to \mathtt{a}X\mathtt{b} \,|\, \mathtt{b}X \,|\, \mathtt{b}$$
$$Y \to \mathtt{b}X\mathtt{c} \,|\, X\mathtt{c} \,|\, \mathtt{c}$$
$$C \to \mathtt{c}C \,|\, \lambda$$
$$A \to \mathtt{a}A \,|\, \lambda$$

e) $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^i \,|\, m+n > i\}$.

$$P \to \mathtt{a}P\mathtt{c} \,|\, \mathtt{b}Q\mathtt{c} \,|\, \mathtt{a}A \,|\, \mathtt{b}B$$
$$Q \to \mathtt{b}Q\mathtt{c} \,|\, \mathtt{b}B$$
$$A \to \mathtt{a}A \,|\, \mathtt{b}B \,|\, \lambda$$
$$B \to \mathtt{b}B \,|\, \lambda$$

f) $\{a^m b^n c^p d^q \,|\, m+n \geq p+q\}$.

$$P \to \mathtt{a}P\mathtt{d} \,|\, \mathtt{b}Q\mathtt{d} \,|\, \mathtt{a}R\mathtt{c} \,|\, \mathtt{b}S\mathtt{c} \,|\, A \,|\, B$$
$$Q \to \mathtt{b}Q\mathtt{d} \,|\, \mathtt{b}S\mathtt{c} \,|\, B$$
$$R \to \mathtt{a}R\mathtt{c} \,|\, \mathtt{b}S\mathtt{c} \,|\, A \,|\, B$$
$$S \to \mathtt{b}S\mathtt{c} \,|\, B$$
$$A \to \mathtt{a}A \,|\, \mathtt{b}B \,|\, \lambda$$
$$B \to \mathtt{b}B \,|\, \lambda$$

g) $\{w \in \{a,b\}^* \,|\, w$ não é da forma $xx\}$.

$$P \to AB \,|\, BA \,|\, A \,|\, B$$
$$A \to XAX \,|\, \mathtt{a}$$
$$B \to XBX \,|\, \mathtt{b}$$
$$X \to \mathtt{a} \,|\, \mathtt{b}$$

14. Uma GLC tendo em vista a definição original:

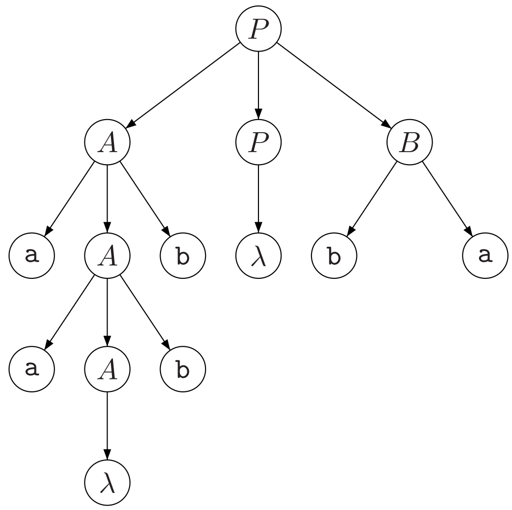
$$E \to (E+E) \,|\, (EE) \,|\, E^* \,|\, a \,|\, b$$

Uma GLC (não ambigua) para ERs com as regras de precedência usuais e parênteses à vontade:

$$E \to E+T \,|\, T$$
$$T \to EF \,|\, F$$
$$F \to I^* \,|\, I$$
$$I \to (E) \,|\, a \,|\, b$$

15. a) $P \Rightarrow APB \Rightarrow \mathtt{a}A\mathtt{b}PB \Rightarrow \mathtt{aa}A\mathtt{bb}PB \Rightarrow \mathtt{aabb}PB \Rightarrow \mathtt{aabb}B \Rightarrow \mathtt{aabbba}$

b) Árvore de derivação:



c) $L(G) = \bigcup_{k\geq 0}\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}^k\{\mathtt{b}^n\mathtt{a}^n \mid n > 0\}^k$.

16. Sejam v_1, v_2, ..., v_n os vértices internos (ou seja, não folhas) da AD, sendo v_1 a raiz da AD. Para $1 \leq i \leq n$, seja $n(v_i)$ o número de vértices internos descendentes de v_i, *incluindo o próprio* v_i, e seja $d(v_i)$ o número de derivações que levam à AD de raiz v_i. O número de derivações que levam à AD é:

$$d(v_1) = \frac{n(v_1)!}{n(v_1) \times n(v_2) \times \cdots \times n(v_n)}.$$

Assim, o número de derivações que levam à AD da Figura 3.16 é $11!/(11.10.2.1.7.6.3.2.1.2.1) = 360$.

Para provar o resultado anterior, veja que (a) se a AD tem apenas um vértice, v_1, $d(v_1) = n(v_1) = 1$, e (b) se a AD tem mais de um vértice, sendo $v_{f_1}, v_{f_2}, \ldots, v_{f_k}$ os filhos da raiz v_1, $n(v_1) = 1 + \sum_{j=1}^{k} n(v_{f_j})$, e

$$d(v_1) = \binom{\sum_{j=1}^{k} n(v_{f_j})}{n(v_{f_1}), \ldots, n(v_{f_k})} \times \prod_{j=1}^{k} d(v_{f_j}) = \frac{n(v_1)!}{n(v_1) \times n(v_{f_1})! \times \ldots \times n(v_{f_k})!} \times \prod_{j=1}^{k} d(v_{f_j}).$$

O primeiro fator é o número de intercalamentos dos vértices das subárvores, considerando, temporariamente, os vértices de uma mesma subárvore como idênticos. O segundo, dá os números de derivações possíveis levando em conta cada subárvore (para cada intercalamento possível). Agora, o resultado pode ser provado por indução forte sobre o número de vértices internos da AD.

17. a) $L(G) = \{xyx^R \mid x \in \{\mathtt{a}, \mathtt{b}\}^*$ e $y \in \{\mathtt{a}\}^+\{\mathtt{b}\}^+\}$.

b) A palavra aabb tem duas DMEs:

$P \Rightarrow \mathtt{a}A\mathtt{b} \Rightarrow \mathtt{aa}A\mathtt{b} \Rightarrow \mathtt{aa}A\mathtt{bb} \Rightarrow \mathtt{aabb}$ e

$P \Rightarrow \mathtt{a}A\mathtt{b} \Rightarrow \mathtt{a}A\mathtt{bb} \Rightarrow \mathtt{aa}A\mathtt{bb} \Rightarrow \mathtt{aabb}$.

Logo, G é ambígua.

c) Uma GLC não ambígua equivalente:

$P \rightarrow \mathtt{a}P\mathtt{a} \mid \mathtt{b}P\mathtt{b} \mid AB$

$A \rightarrow \mathtt{a}A \mid \mathtt{a}$

$B \rightarrow B\mathtt{b} \mid \mathtt{b}$

18. Será mostrado que para todo $x \in \{\mathtt{a}, \mathtt{b}\}^*$, $x \in L(G)$ se, e somente se, todo prefixo de x tem no mínimo tantos as quantos bs. Seja $x \in \{\mathtt{a}, \mathtt{b}\}^*$.

   ($\rightarrow$) Será mostrado, por indução sobre n, que para todo $n \geq 1$, se $P \stackrel{n}{\Rightarrow} x$, então todo prefixo de x tem no mínimo tantos as quantos bs. Observe, inicialmente, que $P \stackrel{1}{\Rightarrow} x$ apenas em um caso: aplicando-se a regra $P \rightarrow \lambda$ para gerar $x = \lambda$; e λ tem tantos as quantos bs: zero. Agora, seja $n \geq 1$ e suponha, como hipótese de indução, que para $k \leq n$, se $P \stackrel{k}{\Rightarrow} x$, então todo prefixo de x tem no mínimo tantos as quantos bs. Uma derivação de $n + 1$ passos só pode ter duas formas, correspondendo às aplicações das duas regras recursivas:

   1. $P \Rightarrow \mathtt{a}P \stackrel{n}{\Rightarrow} x$ (regra $P \rightarrow \mathtt{a}P$) ou
   2. $P \Rightarrow \mathtt{a}P\mathtt{b}P \stackrel{n}{\Rightarrow} x$ (regra $P \rightarrow \mathtt{a}P\mathtt{b}P$).

   No primeiro caso, $x = \mathtt{a}y$ e $P \stackrel{n}{\Rightarrow} y$. Desta última, segue-se, pela hipótese de indução, que todo prefixo de y tem no mínimo tantos as quantos bs; logo, como $x = \mathtt{a}y$, todo prefixo de x tem no mínimo tantos as quantos bs (na verdade, mais as que bs nesse caso). No segundo caso, existem y_1 e y_2 tais que $P \stackrel{n_1}{\Rightarrow} y_1$, $P \stackrel{n_2}{\Rightarrow} y_2$, $x = \mathtt{a}y_1\mathtt{b}y_2$ e $n_1 + n_2 = n$. Pela hipótese de indução, tanto em y_1, quanto em y_2, os prefixos têm no mínimo tantos as quantos bs. Segue-se que os prefixos de $\mathtt{a}y_1$ têm mais as que bs, os prefixos de $\mathtt{a}y_1\mathtt{b}$ têm no mínimo tantos as quantos bs, e os prefixos de $\mathtt{a}y_1\mathtt{b}y_2 = x$ têm no mínimo tantos as quantos bs.

   ($\leftarrow$) Será mostrado, por indução sobre n, que para todo n e x tais que $|x| = n$, se todo prefixo de x tem no mínimo tantos as quantos bs, então $x \in L(G)$. Para $n = 0$, apenas $x = \lambda$ é tal que $|x| = 0$; e $\lambda \in L(G)$, pois $P \Rightarrow \lambda$. Seja $n \geq 0$ e suponha que para todo x tal que $|x| = k$, para $0 \leq k \leq n$, se todo prefixo de x tem no mínimo tantos as quantos bs, então $x \in L(G)$, ou seja, $P \stackrel{*}{\Rightarrow} x$. Seja, então, um w arbitrário tal que $|w| = n + 1$ e todo prefixo de w tenha no mínimo tantos as quantos bs. Basta mostrar que $w \in L(G)$. Como $|w| > 0$, e todo prefixo de w tem no mínimo tantos as quantos bs, $w = ay$ para algum y de n símbolos. Dois casos:

   1. Todo prefixo de y tem no mínimo tantos as quantos bs.
      Pela hipótese de indução, $P \stackrel{*}{\Rightarrow} y$. Como G tem a regra $P \rightarrow \mathtt{a}P$, segue-se que $P \Rightarrow \mathtt{a}P \stackrel{*}{\Rightarrow} \mathtt{a}y = w$. Logo, $w \in L(G)$.
   2. Existe prefixo de y com menos as que bs.
      Como todo prefixo de $w = \mathtt{a}y$ tem no mínimo tantos as quantos bs, deve existir b tal que $y = x_1\mathtt{b}x_2$, sendo que todo prefixo de x_1 tem no mínimo tantos as quantos bs, $x_1\mathtt{b}$ tem tantos as quantos bs e todo prefixo de x_2 tem no mínimo tantos as quantos bs. Pela hipótese de indução, $P \stackrel{*}{\Rightarrow} x_1$ e $P \stackrel{*}{\Rightarrow} x_2$. Como G tem a regra $P \rightarrow \mathtt{a}P\mathtt{b}P$, segue-se que $P \Rightarrow \mathtt{a}P\mathtt{b}P \stackrel{*}{\Rightarrow} \mathtt{a}x_1\mathtt{b}P \stackrel{*}{\Rightarrow} \mathtt{a}x_1\mathtt{b}x_2 = w$. Logo, $w \in L(G)$.

19. Seja $G = (V, \Sigma, R, P)$ uma GLC sem símbolo de partida recursivo, na FNG, que gere a linguagem. Um APN sem transições λ que aceita $L(G)$ é $M = (\{i, f\}, \Sigma, V, \delta, \{i\}, F)$, em que:

   - $F = \{i, f\}$, se $P \rightarrow \lambda \in R$, e $F = \{f\}$, se $P \rightarrow \lambda \notin R$;
   - $\delta(i, a, \lambda) = \{[f, y] \mid P \rightarrow ay \in R\}\}$; e
   - para cada $a \in \Sigma$ e cada $X \in V - \{P\}$, $\delta(f, a, X) = \{[f, y] \mid X \rightarrow ay \in R\}$.

20. Seja uma GLC (V, Σ, R, P) na FNG. Se não existir alguma regra da forma $X \rightarrow aY_1Y_2 \ldots Y_n$ com $n > 2$, G já está na forma pretendida. Se existir, basta fazer o seguinte:

1. Obter $G' = (V \cup V', \Sigma, R', P)$, em que:
   - para cada regra da forma $X \to aY_1Y_2 \ldots Y_n$ com $n \geq 2$, V' contém $n-1$ variáveis *novas*;
   - R' contém todas as regras de R com o lado direito com, no máximo, uma variável mais, para cada regra da forma $X \to aY_1Y_2 \ldots Y_n$ com $n \geq 2$, n regras do tipo:

     $X \to aZ_1$
     $Z_1 \to Y_1Z_2$
     $Z_2 \to Y_2Z_3$
     $\vdots$
     $Z_{n-1} \to Y_{n-1}Y_n$

     em que $Z_1, \ldots, Z_{n-1}$ são as $n-1$ variáveis novas correspondentes à regra original.
2. Por construção, toda regra de G' é de uma das formas:
   - $P \to \lambda$,
   - $X \to a$, com $X \in V$,
   - $X \to aY$, com $X \in V$ e $Y \in V \cup V'$, ou
   - $X \to YZ$, com $X \in V'$ e $Y \in V$ e $Z \in V \cup V'$.

   Para cada regra desta última forma, aplicar o Teorema 24: substituir $X \to YZ$ pelas regras

   $X \to a_1W_1Z \,|\, a_2W_2Z \,|\, \ldots \,|\, a_kW_kZ$

   sendo a_1W_1, a_2W_2, $\ldots$, a_kW_k os lados direitos de todas as regras Y. Observe que, como $Y \in V$, $a_i \in \Sigma$ e $W_i \in V \cup V' \cup \{\lambda\}$.

21. Para a variável de partida, o processo termina imediatamente, já que ela tem o menor número. Assim, basta mostrar que o processo termina para uma variável X de número n ($\#X = n$), supondo, como hipótese de indução, que termina para variáveis de número menor que n. Seja, então, uma variável X tal que $\#X = n$. Se não existe regra cujo lado direito começa com uma variável de número menor que ou igual a n, o processo termina imediatamente. Caso haja regra da forma $X \to Yw$ com $\#Y < n$, ela é substituída, aplicando-se o Teorema 24, por regras da forma $X \to zw$ para cada regra $Y \to z$. Se z começa com variável, o número dela é maior que o de Y (pois $\#Y < n$). Assim, repetindo-se a aplicação do Teorema 24, sempre que preciso, obtém-se lados direitos de regras X começando com variáveis de *número cada vez maior*, mas *menor ou igual* a n. Com isso, só irão restar, se restar, regras $X \to Yw$ com $\#Y = n$ (ou seja $X = Y$) ou $\#Y > n$. Se houver regra da forma $X \to Yw$ com $X = Y$, aplica-se o Teorema 23. E o processo termina, de qualquer maneira.

22. a) Seja uma GLC $G = (V, \Sigma, R, P)$. Para verificar se $L(G) = \emptyset$:

    determine $\mathcal{I}_1 = \{X \in V \,|\, X \stackrel{*}{\Rightarrow} w$ para algum $w \in \Sigma^*\}$ usando o algoritmo da Figura 3.18(a);
    se $P \notin \mathcal{I}_1$ **então**
    retorne *sim* /* $L(G) = \emptyset$ */
    senão
    retorne *não* /* $L(G) \neq \emptyset$ */
    fimse.

    b) Seja uma GLC $G = (V, \Sigma, R, P)$. Para verificar se $L(G)$ é finita:

elimine toda variável inútil de G, obtendo $G'' = (V'', \Sigma, R'', P)$ como mostrado na prova do Teorema 17;
se R'' tem regra recursiva (da forma $X \rightarrow xXy$) **então**
retorne *não* /* $L(G)$ é infinita */
senão
retorne *sim* /* $L(G)$ é finita*/
fimse.

23. São dadas uma GLC G e uma palavra w. Sega G' uma GLC equivalente a G em que o símbolo de partida seja P, a única transição λ, se houver, seja $P \rightarrow \lambda$, e P não seja recursivo. Como cada regra de G' (com exceção de $P \rightarrow \lambda$) tem o lado direito maior ou igual ao lado esquerdo e como P não é recursivo, se $u \Rightarrow v$ então $|u| \leq |v|$ (a menos que $u = P$ e $v = \lambda$). Com isto, o número de derivações da forma $P \Rightarrow \cdots \Rightarrow z$, em que $|z| = |w|$, e nas quais *não ocorrem formas sentencias idênticas* é finito. Evidentemente, se alguma dessas formas sentenciais z for w, $w \in L(G)$; caso contrário, $w \notin L(G)$.

24. a) Uma GLC que gera $L = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid m \neq n \text{ ou } n \neq k\}$:

$$
\begin{aligned}
P &\rightarrow XC \mid AY \\
X &\rightarrow \mathtt{a}X\mathtt{b} \mid \mathtt{a}A \mid \mathtt{b}B \\
Y &\rightarrow \mathtt{b}Y\mathtt{c} \mid \mathtt{b}B \mid \mathtt{c}C \\
A &\rightarrow \mathtt{a}A \mid \lambda \\
B &\rightarrow \mathtt{b}B \mid \lambda \\
C &\rightarrow \mathtt{c}C \mid \lambda
\end{aligned}
$$

b) Suponha que $\overline{L}$ é uma LLC. Então $\overline{L} \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^*$ deve ser LLC. Mas $\overline{L} \cap \{\mathtt{a}\}^*\{\mathtt{b}\}^*\{\mathtt{c}\}^* = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$, que não é LLC. Logo, $\overline{L}$ não é uma LLC.

25. **a)** $L = \{\mathtt{0}^m\mathtt{1}^n\mathtt{2}^k \mid m < n < k\}$. Suponha que L é LLC e seja k a constante do LB. Seja $z = \mathtt{0}^k\mathtt{1}^{k+1}\mathtt{2}^{k+2}$. Como $|z| > k$, o LB diz que existem u, v, w, x e y tais que $z = uvwxy$, $|vwx| \leq k$, $vx \neq \lambda$ e $uv^iwx^iy \in L$ para todo $i \geq 0$. Tem-se dois casos:

Caso 1: vx contém $\mathtt{0}$. Sendo $|vwx| \leq k$, vx não pode conter $\mathtt{2}$s; e sendo $vx \neq \lambda$, o número de $\mathtt{0}$s em uv^3wx^3y será maior ou igual ao de $\mathtt{2}$s. Logo, $uv^3wx^3y \notin L$.

Caso 2: vx não contém $\mathtt{0}$. Sendo $vx \neq \lambda$, vx contém $\mathtt{1}$ ou $\mathtt{2}$. Se vx contiver $\mathtt{1s}$, o número de $\mathtt{0}$s em uv^0wx^0y será maior ou igual ao de $\mathtt{1}$s; e se vx não contiver $\mathtt{1s}$, o número de $\mathtt{1}$s em uv^0wx^0y será maior ou igual ao de $\mathtt{2}$s. Logo, $uv^0wx^0y \notin L$.

Em qualquer caso, entra-se em contradição com o LB. Portanto, L não é LLC.

b) $L = \{\mathtt{0}^n\mathtt{1}^{n^2} \mid n \geq 0\}$. Suponha que L é LLC e seja k a constante do LB. Seja $z = \mathtt{0}^k\mathtt{1}^{k^2}$. Como $|z| > k$, o LB diz que existem u, v, w, x e y tais que $z = uvwxy$, $|vwx| \leq k$, $vx \neq \lambda$ e $uv^iwx^iy \in L$ para todo $i \geq 0$. Se $vx \neq \lambda$, $|uv^2wx^2y| > k + k^2$. E se $|vwx| \leq k$, $|vx| \leq k$ e, assim, $|uv^2wx^2y| \leq k + k^2 + k = 2k + k^2 < 2 + 3k + k^2 = (k+1) + (k+1)^2$. Portanto, $k + k^2 < |uv^2wx^2y| < (k+1) + (k+1)^2$ e, logo, não existe n tal que $|uv^2wx^2y| = n + n^2$. Assim, não é possível que uv^2wx^2y seja da forma $\mathtt{0}^n\mathtt{1}^{n^2}$, o que contradiz o LB.

c) $L = \{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^k \mid n \leq k \leq 2n\}$. Suponha que L seja LLC e seja k a constante do LB. Seja $z = \mathtt{0}^k\mathtt{1}^k\mathtt{2}^{2k}$. Como $|z| > k$, o LB diz que existem u, v, w, x e y tais que $z = uvwxy$, $|vwx| \leq k$, $vx \neq \lambda$ e $uv^iwx^iy \in L$ para todo $i \geq 0$. Tem-se dois casos:

Caso 1: vx contém $\mathtt{0}$. Sendo $|vwx| \leq k$, vx não pode conter $\mathtt{2}$s; e sendo $vx \neq \lambda$, o número de $\mathtt{2}$s em uv^0wx^0y será mais que o dobro do de $\mathtt{0}$s. Logo, $uv^0wx^0y \notin L$.

Caso 2: vx não contém `0`. Sendo $vx \neq \lambda$, vx contém `1` ou `2`. Se vx contiver `1s`, o número de `1s` em uv^2wx^2y será maior que de `0s`; e se vx não contiver `1`, o número de `2s` em uv^2wx^2y será mais que o dobro do de `0s`. Logo, $uv^2wx^2y \notin L$.

Em qualquer caso, entra-se em contradição com o LB. Portanto, L não é LLC.

d) $L = \{ww \mid w \in \{\mathtt{0},\mathtt{1}\}^*\}$. Suponha que L seja LLC e seja k a constante do LB. Seja $z = \mathtt{0}^k\mathtt{1}\mathtt{0}^k\mathtt{1}$. Como $|z| > k$, o LB diz que existem u, v, w, x e y tais que $z = uvwxy$, $|vwx| \leq k$, $vx \neq \lambda$ e $uv^iwx^iy \in L$ para todo $i \geq 0$. Tem-se três casos:

Caso 1: vx contém apenas `0s` do prefixo de k `0s`. Sendo $vx \neq \lambda$, se uv^2wx^2y tem número par de símbolos, a primeira metade tem `0` no final e a segunda metade tem `1`. Logo, uv^2wx^2y não é da forma ww e, assim, $uv^2wx^2y \notin L$.

Caso 2: vx contém `1`. Como $|vwx| \leq k$, uv^0wx^0y conterá apenas um dos dois `1s` e, portanto, $uv^0wx^0y \notin L$ (uma palavra com apenas um `1` nunca pode ser da forma ww).

Caso 3: vx contém apenas `0s` do sufixo de k `0s`. Sendo $vx \neq \lambda$, se uv^2wx^2y tem número par de símbolos, a primeira metade tem `0` no final e a segunda metade tem `1`. Logo, uv^2wx^2y não é da forma ww e, assim, $uv^2wx^2y \notin L$.

Em qualquer caso, entra-se em contradição com o LB. Portanto, L não é LLC.

e) $L = \{ww^Rw \mid w \in \{\mathtt{0},\mathtt{1}\}^*\}$. Suponha que L seja LLC e seja k a constante do LB. Seja $z = \mathtt{0}^k\mathtt{110}^{2k}\mathtt{1}$. Como $|z| > k$, o LB diz que existem u, v, w, x e y tais que $z = uvwxy$, $|vwx| \leq k$, $vx \neq \lambda$ e $uv^iwx^iy \in L$ para todo $i \geq 0$. Tem-se três casos:

Caso 1: vx contém apenas `0s` do prefixo de k `0s`. Sendo $vx \neq \lambda$, se uv^2wx^2y tem um número de símbolos múltiplo de 3, a primeira terça parte tem `0` no final e a terceira tem `1` no final. Logo, $uv^2wx^2y \notin L$.

Caso 2: vx contém `1`. Como $|vwx| \leq k$, uv^0wx^0y conterá apenas `1s` dos primeiros dois terços ou então o `1` do final; logo, uv^0wx^0y tem um ou dois `1s` e, assim, $uv^0wx^0y \notin L$.

Caso 3: vx contém apenas `0s` do terço final de z. Sendo $vx \neq \lambda$, se uv^2wx^2y tem um número de símbolos múltiplo de 3, a segunda terça parte começa com `0` e a terceira tem `1` no final. Logo, $uv^2wx^2y \notin L$.

Em qualquer caso, entra-se em contradição com o LB. Portanto, L não é LLC.

26. Igual ao Exercício 13(g).

27. **a)** Suponha que L é LLC e F é finita. Como F é finita, é regular. Como as linguagens regulares são fechadas sob complementação, $\overline{F}$ é regular. A interseção de LLC com linguagem regular é LLC; assim, $L \cap \overline{F} = L - F$ é LLC. Conclusão: se L é uma LLC e F é finita, então $L - F$ é LLC.

b) Suponha que L é LLC e R é regular. Como as linguagens regulares são fechadas sob complementação, $\overline{R}$ é regular. A interseção de LLC com linguagem regular é LLC; assim, se L é uma LLC e R é regular, então $L - R$ é LLC. Conclusão: se L é uma LLC e R é regular, então $L - R$ é LLC.

c) Suponha que L não é LLC e F é finita. Suponha que $L - F$ é LLC. Como $L \cap F$ é finita, é LLC. Como as LLCs são fechadas sob união, $(L - F) \cup (L \cap F)$ é LLC. Mas $(L - F) \cup (L \cap F) = L$, que não é LLC. Contradição! Logo se L não é uma LLC e F é finita, então $L - F$ não é LLC.

d) Seja L não LLC sobre o alfabeto Σ. Σ^* é regular e $L - \Sigma^* = \emptyset$, uma linguagem regular. Assim, se L não é uma LLC e R é regular, $L - R$ pode ser LLC.

e) Suponha que L não é LLC e F é finita. Suponha que $L \cup F$ é LLC. Como $F - L$ é finita, é regular e, assim, $\overline{F - L}$ também é regular. Mas, $(L \cup F) - (F - L) = (L \cup F) \cap \overline{F - L} = L$ e esta não é LLC. Contradição! Logo se L não é uma LLC e F é finita, então $L \cup F$ não é LLC.

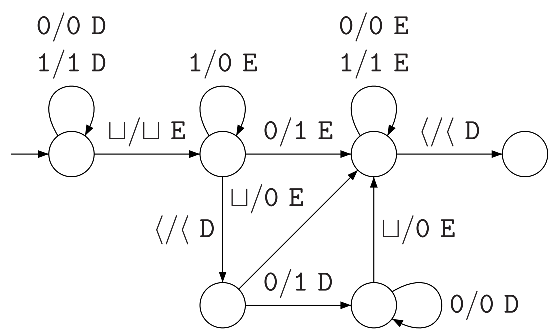
f) Seja L não LLC sobre o alfabeto Σ. Σ^* é uma linguagem regular e $L \cup \Sigma^* = \Sigma^*$. Portanto, se L não é uma LLC e R é regular, $L \cup R$ pode ser LLC.

Capítulo 4

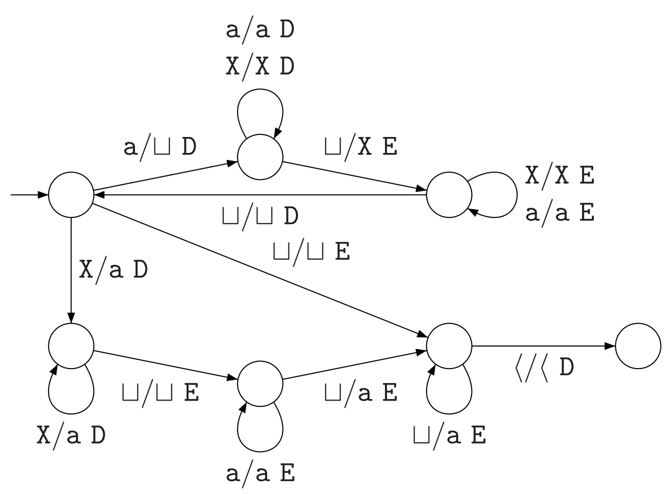
Máquinas de Turing

4.1 O que É Máquina de Turing

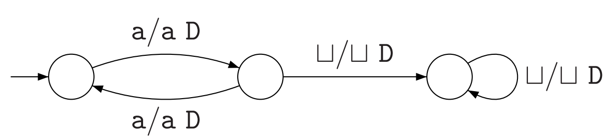
1.



2.



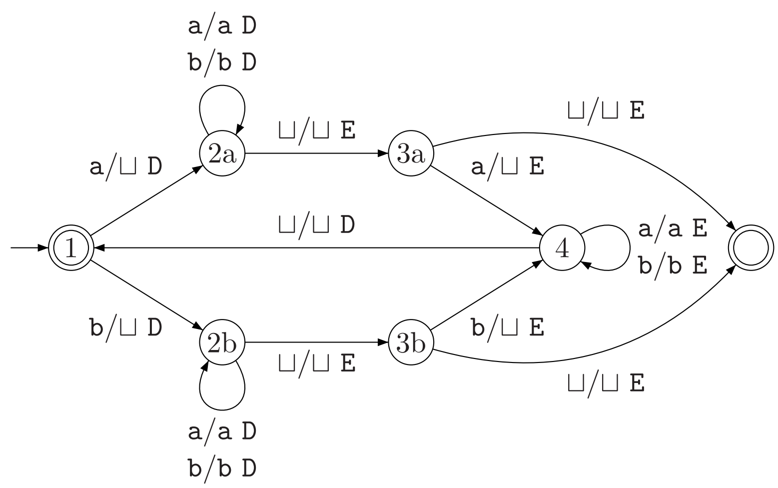
3.



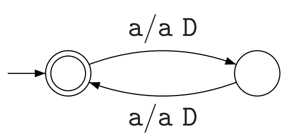
4. a) MT de 1 estado: $(\{0\}, \{\mathtt{a}, \mathtt{b}\}, \{\langle, \sqcup, \mathtt{a}, \mathtt{b}\}, \langle, \sqcup, \delta, 0, \{0\})$, onde δ consta de: $\delta(0, \mathtt{b}) = [0, \mathtt{b}, \mathtt{E}]$, $\delta(0, \sqcup) = [0, \sqcup, \mathtt{E}]$, $\delta(0, \langle) = [0, \langle, \mathtt{D}]$.

b) MT de 1 transição: $(\{0,1\}, \{\texttt{a}, \texttt{b}\}, \{\langle, \sqcup, \texttt{a}, \texttt{b}\}, \langle, \sqcup, \delta, 0, \{1\})$, onde δ consta de: $\delta(0, \texttt{a}) = [1, \texttt{a}, \texttt{E}]$.

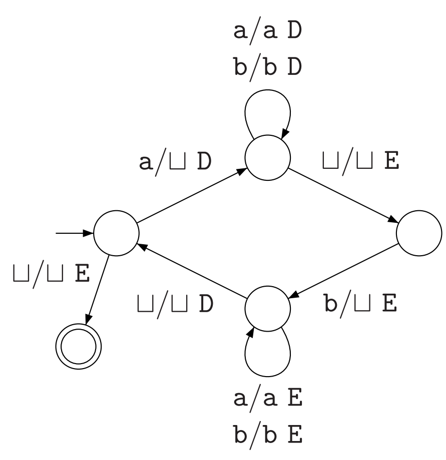
5.



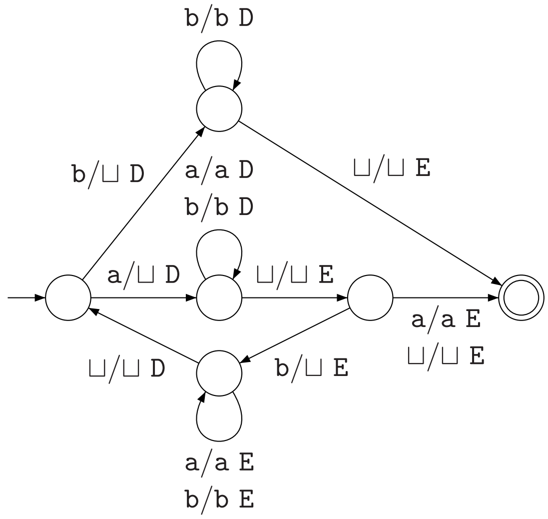
6. **a)**



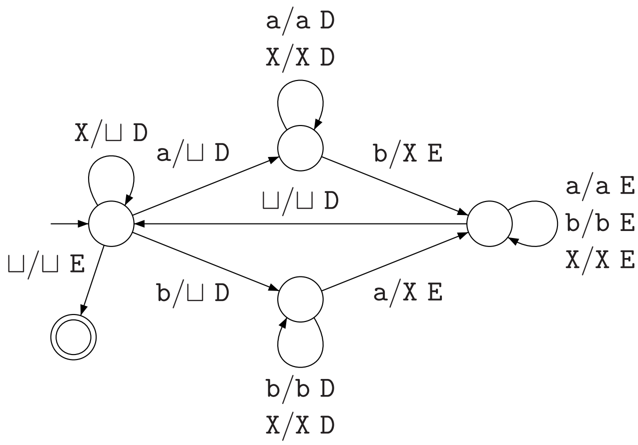
b)



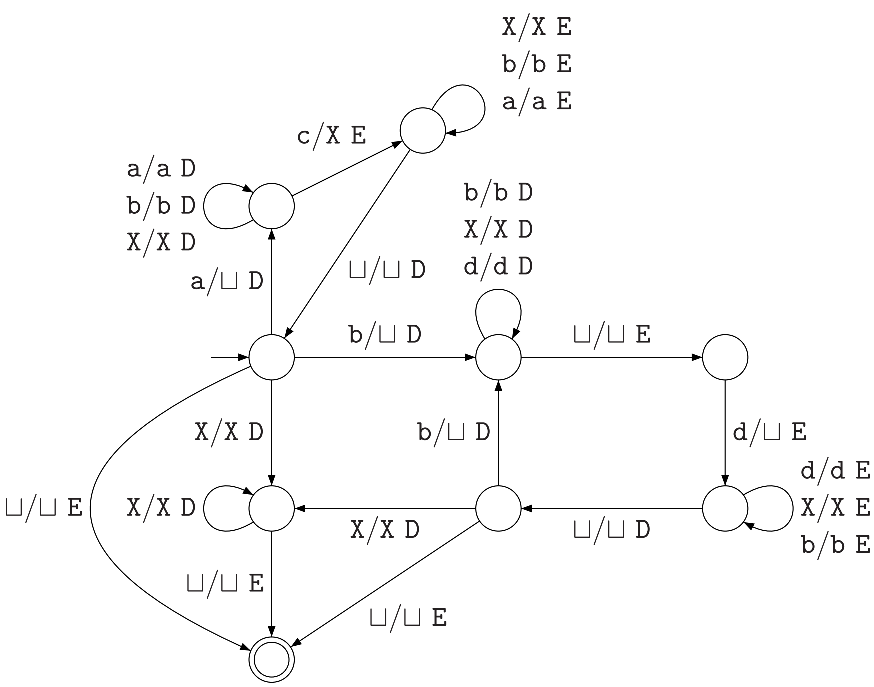
c)


b/b D
b/⊔ D
a/a D
b/b D
⊔/⊔ E
a/⊔ D
⊔/⊔ E
a/a E
⊔/⊔ E
⊔/⊔ D
b/⊔ E
a/a E
b/b E

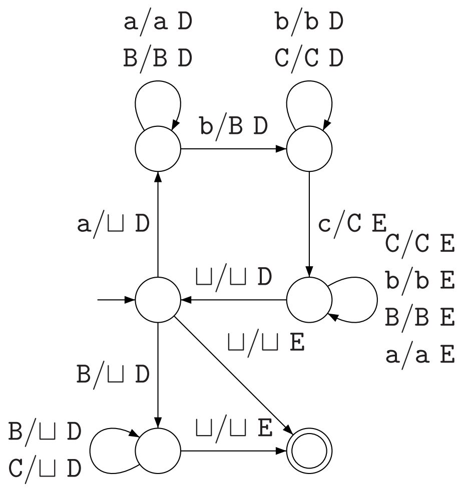
d)


a/a D
X/X D
X/⊔ D
a/⊔ D
b/X E
⊔/⊔ D
a/a E
b/b E
X/X E
⊔/⊔ E
b/⊔ D
a/X E
b/b D
X/X D

e)

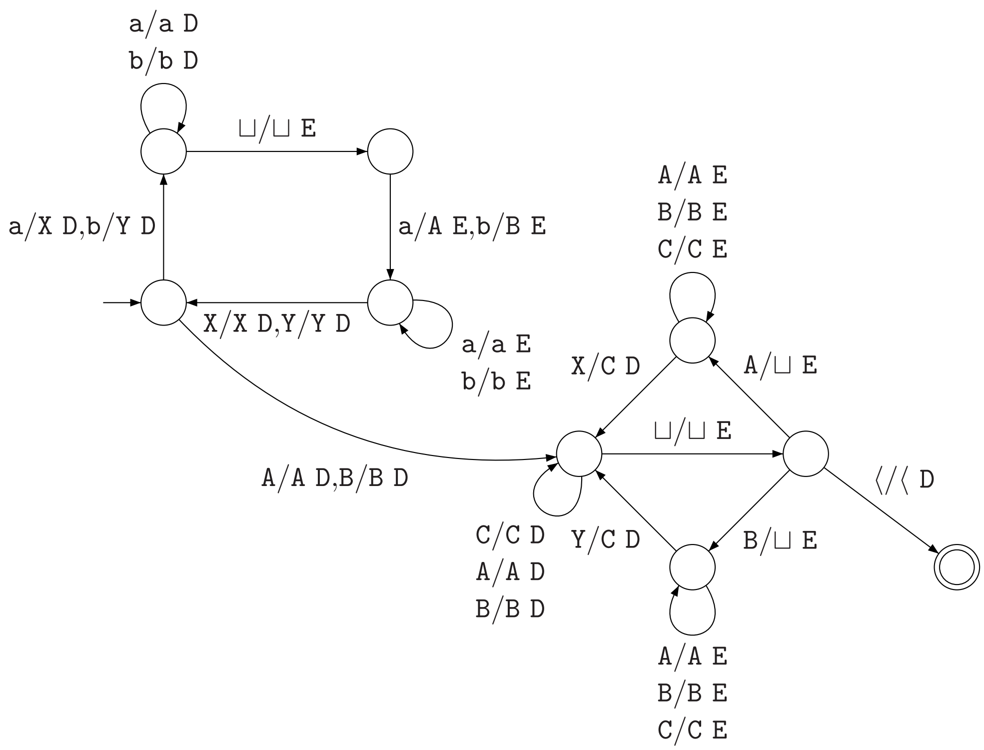

X/X E
b/b E
a/a E
c/X E
a/a D
b/b D
X/X D
b/b D
X/X D
d/d D
⊔/⊔ D
a/⊔ D
b/⊔ D
⊔/⊔ E
X/X D
b/⊔ D
d/⊔ E
d/d E
X/X E
b/b E
⊔/⊔ E
X/X D
X/X D
⊔/⊔ D
⊔/⊔ E
⊔/⊔ E

f)



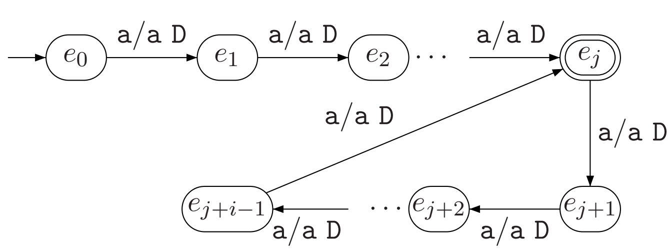
a/a D
B/B D
b/b D
C/C D
b/B D
a/⊔ D
c/C E
C/C E
b/b E
B/B E
a/a E
⊔/⊔ D
⊔/⊔ E
B/⊔ D
B/⊔ D
C/⊔ D
⊔/⊔ E

g)



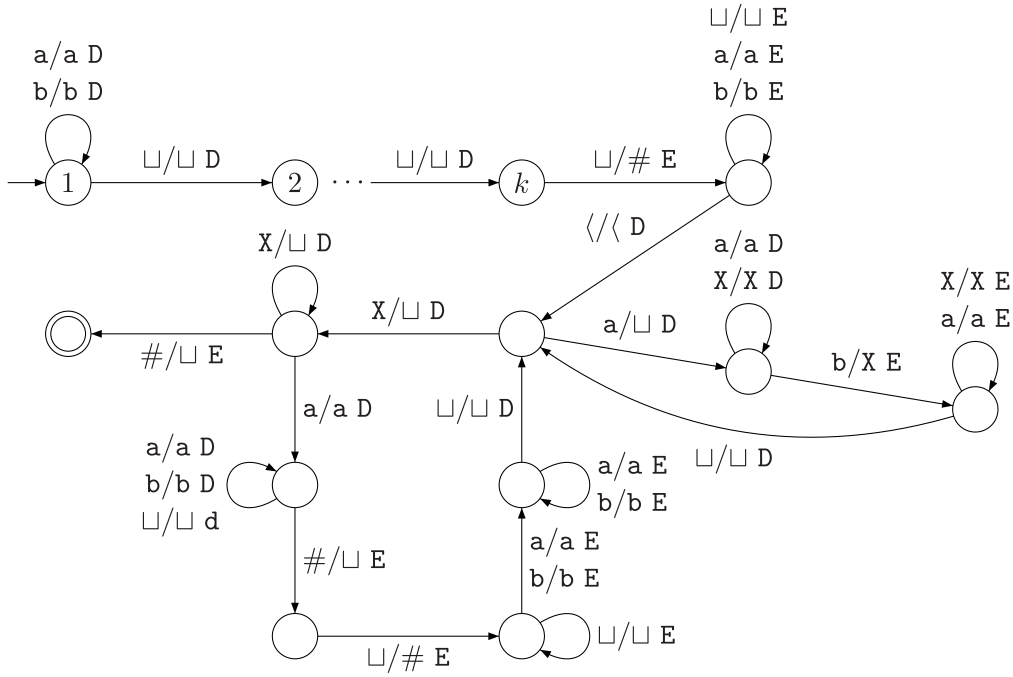
a/a D
b/b D
⊔/⊔ E
a/X D,b/Y D
a/A E,b/B E
X/X D,Y/Y D
a/a E
b/b E
A/A E
B/B E
C/C E
X/C D
A/⊔ E
⊔/⊔ E
A/A D,B/B D
⟨/⟨ D
C/C D
A/A D
B/B D
Y/C D
B/⊔ E
A/A E
B/B E
C/C E

7.

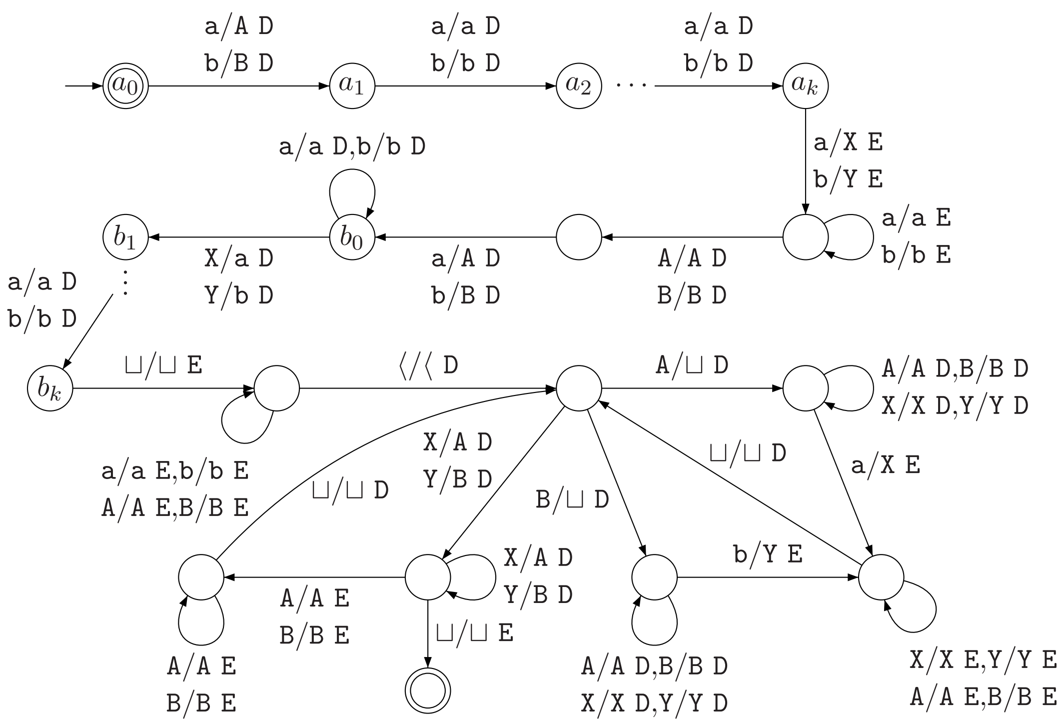


e_0
a/a D
e_1
a/a D
e_2 ···
a/a D
e_j
a/a D
a/a D
e_{j+i-1}
a/a D
··· e_{j+2}
a/a D
e_{j+1}

8. **a)**



b) Se $k = 1$, a MT é $(\{0\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a}, \mathtt{b}, \langle, \sqcup\}, \langle, \sqcup, \{\}, 0, \{0\})$, que reconhece $\{\mathtt{a}, \mathtt{b}\}^*$. MTs para $k \geq 2$ podem ser assim construídas:

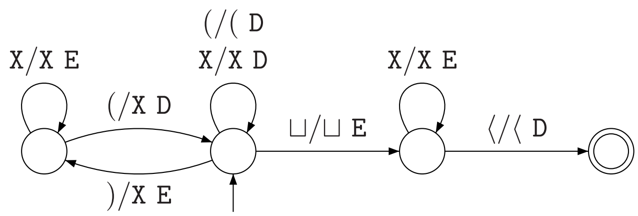


9. Seja um AFD $M = (E, \Sigma, \delta, i, F)$. Uma MT que reconhece $L(M)$ seria

$$M' = (E, \Sigma, \Sigma \cup \{\langle, \sqcup\}, \langle, \sqcup, \delta', i, F),$$

em que para todo par $(e, a) \in E \times \Sigma$, $\delta'(e, a) = [\delta(e, a), a, \mathtt{D}]$.

10. Diagrama de estados de uma máquina de Turing para parênteses balanceados:



11. Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ a MT original.

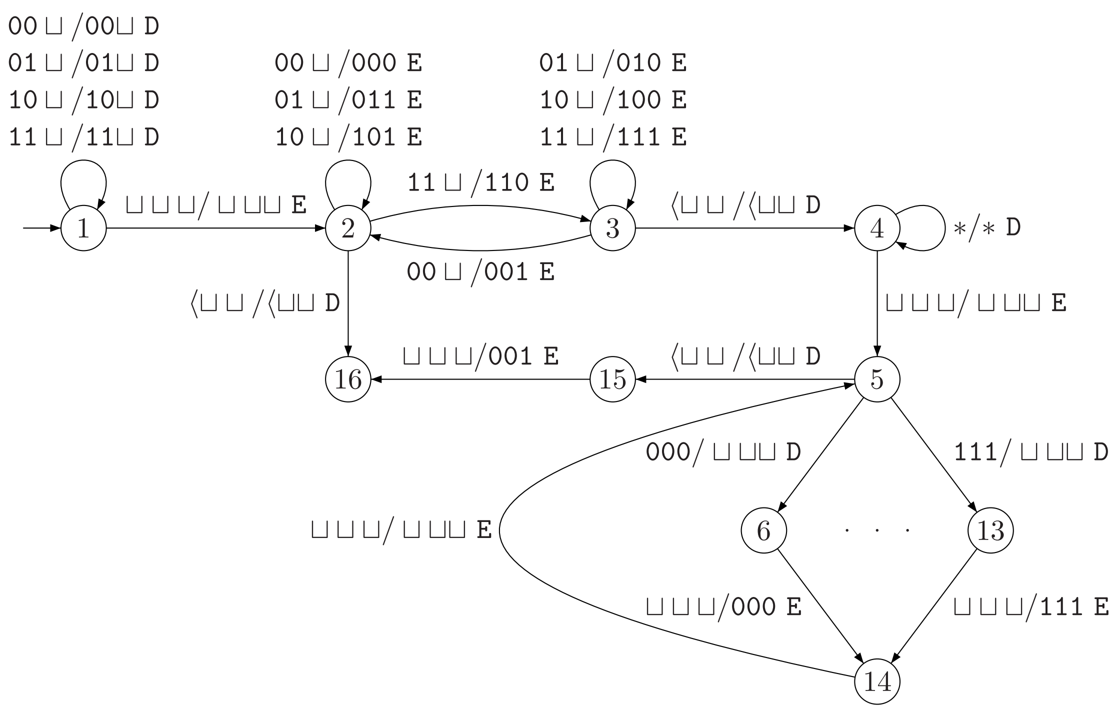
    a) Se M reconhece por estado final, uma MT que reconhece $L(M)$ por parada em estado final é: $(E, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$, sendo δ' como δ, mas com $\delta'(e, a)$ indefinido para todo $(e, a) \in F \times \Gamma$.

    b) Se M reconhece por parada, uma MT que reconhece $L(M)$ por estado final é: $(E \cup \{n\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i, \{n\})$, sendo que $n \notin E$ e δ' é como δ, mas com o acréscimo das transições $\delta'(e, a) = [n, a, \mathtt{D}]$, para cada (e, a) tal que $\delta(e, a)$ é indefinido.

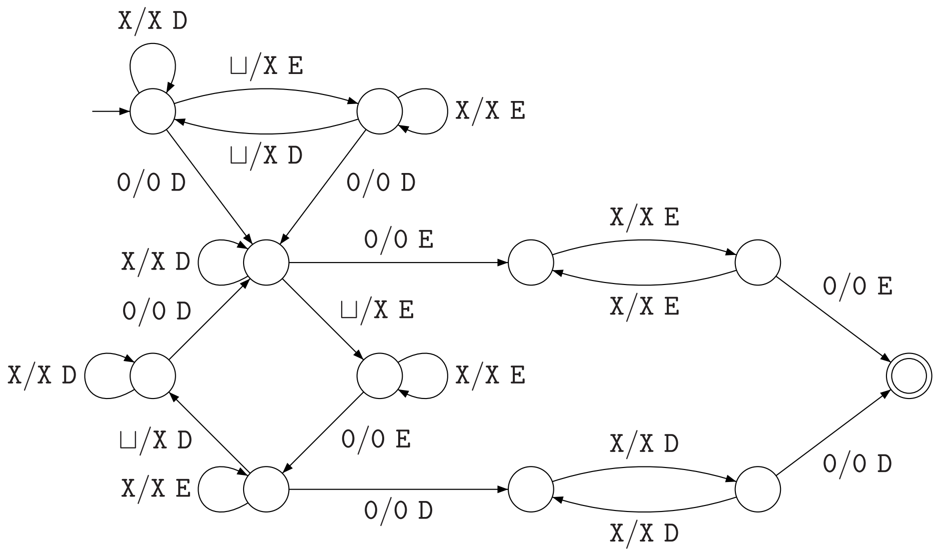
    c) Se M reconhece por parada em estado final, uma MT que reconhece $L(M)$ por parada é: $(E \cup \{n\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i)$, sendo que $n \notin E$ e δ' é como δ, mas com o acréscimo das transições $\delta'(e, a) = [n, a, \mathtt{D}]$, para cada $(e, a) \in (E - F) \times \Gamma$ tal que $\delta(e, a)$ é indefinido e $\delta'(n, a) = [n, a, \mathtt{D}]$, para cada $a \in \Gamma$.

4.2 Algumas Variações de Máquinas de Turing

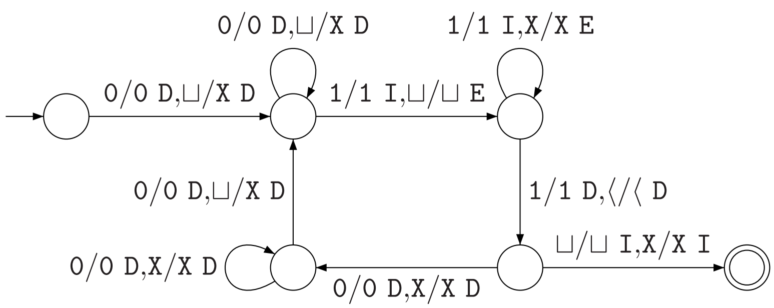
1. O *loop* com rótulo $*/*$ D (estado 4) representa 8 transições: 000/000 D, 001/001 D, 010/010 D, 011/011 D, 100/100 D, 101/101 D, 110/110 D e 111/111 D.



2.

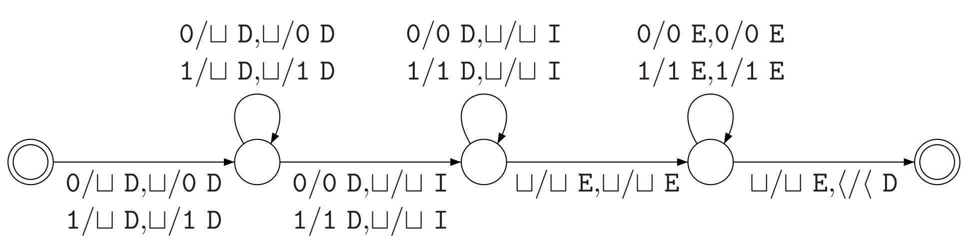


3.

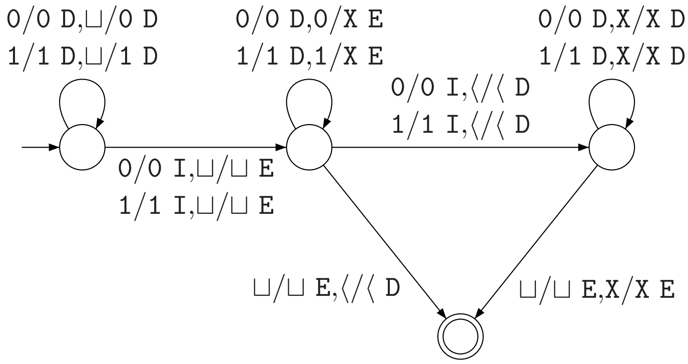


4. Começando pela transição $[b, \mathtt{0}, \mathtt{D}] \in \delta(a, \mathtt{0})$, M reconhece as palavras de $0 + 01(0+1)^*$, já que a ocorrência de ⊔ ou `1` faz M parar no estado final b, e a ocorrência de `0` faz M entrar em *loop*. Por outro lado, começando pela transição $[d, \mathtt{0}, \mathtt{D}] \in \delta(a, \mathtt{0})$, M reconhece as palavras de 00^*, já que no estado d só são admitidos `0`s até o final da palavra de entrada, quando ocorre a transição para o estado final b (avançando-se o para cabeçote para a direita, onde está um ⊔: assim, M pára em b). Portanto, M reconhece $0^+ + 01(0+1)^*$.

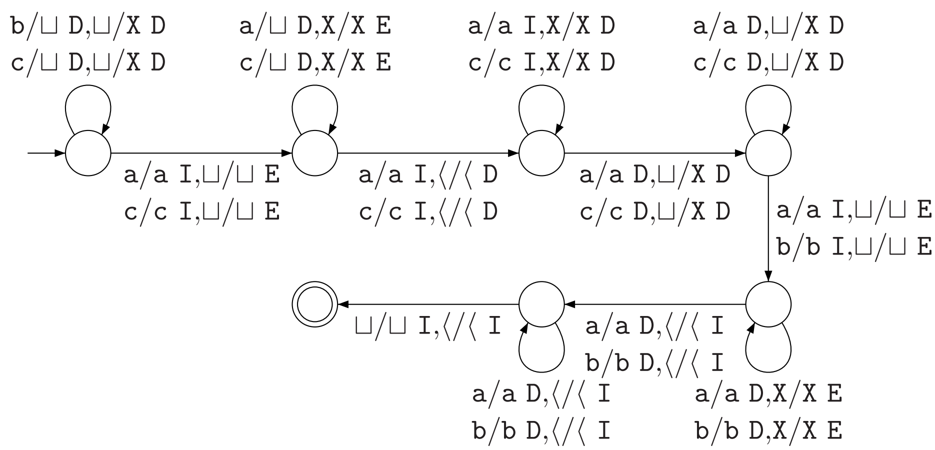
5. **a)**



b)



c)



6. Seja uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$. Uma MT com um único estado final que reconhece $L(M)$ por estado final é $M' = (E \cup \{f\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i, \{f\})$, em que $f \notin E$ e δ' é tal que:

   - $\delta'(e, a) = \delta(e, a)$ para todo (e, a) tal que $\delta(e, a)$ é definido;
   - $\delta'(e, a) = [f, a, \mathtt{D}]$ para todo $(e, a) \in F \times \Gamma$ tal que $\delta(e, a)$ é indefinido;
   - $\delta'(e, a)$ é indefinido nos casos restantes.

7. Seja $\{i_1, i_2, \ldots, i_n\}$ o conjunto de estados iniciais. Basta acrescentar um *novo estado* i como estado inicial e acrescentar as transições $\delta(i, a) = \{[i_k, a, I] \mid 1 \leq k \leq n\}$, para cada $a \in \Sigma \cup \{\sqcup\}$.

8. A máquina de Turing em questão é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ em que E, Σ, $\langle$, $\sqcup$ i e F são como em MTs padrão, cuidando que $\mathtt{E}, \mathtt{D} \notin \Gamma$, e $\delta : E \times \Gamma \to E \times (\Gamma \cup \{\mathtt{E}, \mathtt{D}\})$ é a função de transição, uma função parcial. O reconhecimento por esse tipo de MT é definido da mesma forma que para MT padrão.

   Será mostrado que esse tipo de MT reconhece uma linguagem L se, e somente se, L é LRE:

   $(\rightarrow)$ Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT como definido acima. Uma MT padrão que reconhece $L(M)$ seria $(E \cup N, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$, onde N é o conjunto de novos estados obtido como a seguir e δ' é assim obtida a partir de δ:

   - para cada transição da forma $\delta(e, a) = [e', b]$, $b \in \Gamma$:
     - criar um *novo* estado e_1;
     - criar uma transição $\delta'(e, a) = [e_1, b, \mathtt{D}]$; e

- criar transições $\delta'(e_1, c) = [e', c, \mathtt{E}]$ para cada $c \in \Gamma - \{\langle\}$.

- para cada transição da forma $\delta(e, a) = [e', d]$, $d \in \{\mathtt{E}, \mathtt{D}\}$:
  - criar uma transição $\delta'(e, a) = [e', a, d]$.

($\leftarrow$) Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT padrão. Uma MT como definido acima que reconhece $L(M)$ seria $(E \cup N, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$, onde N é o conjunto de novos estados obtido como a seguir e δ' é assim obtida a partir de δ:

- para cada transição da forma $\delta(e, a) = [e', b, d]$, $b \in \Gamma$ e $d \in \{\mathtt{E}, \mathtt{D}\}$:
  - criar um *novo* estado e_1;
  - criar uma transição $\delta'(e, a) = [e_1, b]$; e
  - criar uma transição $\delta'(e_1, b) = [e', d]$.

9. Basta mostrar como obter uma MT com a restrição em questão que seja equivalente a uma MT padrão. Para isso, basta criar um *novo* símbolo de fita, B, e substituir a função de transição δ da MT padrão pela função de transição δ' assim obtida:

   - se $\delta(e, a) = [e', b, d]$ e $a \neq \sqcup$ e $b \neq \sqcup$, então $\delta'(e, a) = \delta(e, a)$;
   - se $\delta(e, a) = [e', \sqcup, d]$ e $a \neq \sqcup$, então $\delta'(e, a) = [e', B, d]$;
   - se $\delta(e, \sqcup) = [e', a, d]$ e $a \neq \sqcup$, então $\delta'(e, \sqcup) = [e', a, d]$ e $\delta'(e, B) = [e', a, d]$;
   - se $\delta(e, \sqcup) = [e', \sqcup, d]$, então $\delta'(e, \sqcup) = [e', B, d]$ e $\delta'(e, B) = [e', B, d]$.

   Apenas essas transições fazem parte da máquina.

10. A máquina de Turing em questão é uma óctupla $(E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ em que E, Σ, $\langle$, $\sqcup$ i e F são como em MTs padrão e $\delta : E \times \Gamma \times \Gamma \rightarrow E \times \Gamma \times \{\mathtt{E}, \mathtt{D}\}$ é a função de transição, uma função parcial. O reconhecimento por esse tipo de MT é definido da mesma forma que para MT padrão.

    Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT como definido acima. Uma MT padrão que reconhece $L(M)$ seria $(E \cup N, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$, onde N é o conjunto de novos estados obtido como a seguir e δ' é obtida a partir de δ:

    - para cada transição da forma $\delta(e, a, b) = [e', c, d]$:
      - criar dois *novos* estados e_1 e e_2;
      - criar uma transição $\delta'(e, a) = [e_1, a, \mathtt{D}]$;
      - criar uma transição $\delta'(e_1, b) = [e_2, b, \mathtt{E}]$; e
      - criar uma transição $\delta'(e_2, a) = [e', c, d]$.

    Por outro lado, dada uma MT padrão $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT como definido acima equivalente seria $(E, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$ em que δ' é tal que, sendo $\delta(e, a) = [e', b, d]$, $\delta'(e, a, c) = [e, b, d]$ para cada $c \in \Gamma - \{\rangle\}$.

11. Seja $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT desse tipo. Ela reconhece, necessariamente, uma linguagem regular. Lembrando que M é determinística, obtém-se um AFD equivalente $(E \cup \{x, f\}, \Sigma, \delta', i, F \cup \{f\})$, em que $x, f \notin E$ e δ' é obtida de δ assim:

    - para cada $(e, a) \in E \times \Sigma$ tal que $\delta(e, a)$ é indefinido:
      - se $e \in F$, fazer $\delta(e, a) = f$; caso contrário, fazer $\delta(e, a) = x$;
    - fazer $\delta(f, a) = f$ para todo $a \in \Sigma$;

- fazer $\delta(x,a) = x$ para todo $a \in \Sigma$;
- para cada transição da forma $\delta(e,a) = [e', b, \mathtt{D}]$, $a \in \Sigma$:
  - fazer $\delta'(e,a) = e'$.
- para cada transição da forma $\delta(e,a) = [e', b, \mathtt{I}]$, $a \in \Sigma$:
  - seja $(d,c) \in E \times \Gamma$ tal que $[e,\underline{a}] \overset{*}{\vdash} [d,\underline{c}]$ e $\delta(d,c) \neq [d', c', \mathtt{I}]$ para quaisquer d' e c'; se $\delta(d,c)$ for definido, fazer $\delta'(e,a) = \delta(d,c)$; se não for, fazer $\delta'(e,a) = f$ se $e \in F$, ou $\delta'(e,a) = x$ se $e \notin F$.

12. Seja uma máquina de Turing $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$. Um AP de duas pilhas para $L(M)$ é $(E \cup \{i', j_1, j_2, f\}, \Sigma, \Gamma \cup \{\mathtt{F}\}, \delta', i', \{f\})$, em que δ' é contém apenas as transições:
    - $\delta'(i', \lambda, \lambda, \lambda) = [j_1, \langle, \lambda]$
      (empilha $\langle$ na pilha 1 e vai para o estado j_1);
    - para cada $a \in \Sigma$: $\delta'(j_1, a, \lambda, \lambda) = [j_1, a, \lambda]$
      (lê a palavra de entrada e empilha na pilha 1);
    - $\delta'(j_1, \lambda, \lambda, \lambda) = [j_2, \lambda, \mathtt{F}]$
      (acaba a leitura da palavra de entrada e empilha $\mathtt{F}$ na pilha 2);
    - para cada $a \in \Sigma$: $\delta'(j_2, \lambda, a, \lambda) = [j_2, \lambda, a]$
      (transfere a palavra de entrada da pilha 1 para a pilha 2);
    - $\delta'(j_2, \lambda, \langle, \lambda) = [i, \langle, \lambda]$
      (acaba a transferência da palavra de entrada para a pilha 2 e vai para o estado i);
    - para cada $\delta(e,a) = [e', b, \mathtt{E}]$, para cada $c \in \Gamma - \{\langle\}$: $\delta'(e, \lambda, c, a) = [e', \lambda, cb]$
      (simula uma transição com movimento para a esquerda);
    - para cada $\delta(e,a) = [e', b, \mathtt{D}]$: $\delta'(e, \lambda, \lambda, a) = [e', b, \lambda]$ e, se $a = \sqcup$, também $\delta'(e, \lambda, \lambda, \mathtt{F}) = [e', b, \mathtt{F}]$
      (simula uma transição com movimento para a direita);
    - para cada $(e,a) \in E \times \Gamma$ tal que $\delta(e,a)$ é indefinido e $e \in F$: $\delta'(e, \lambda, \lambda, a) = [f, \lambda, a]$
      (simula aceitação da MT).

4.3 Gramáticas e Máquinas de Turing

1. a) Gramática para $\{\mathtt{0}^n\mathtt{1}^k\mathtt{0}^n\mathtt{1}^k \mid n,k \geq 0\}$:

    $$\begin{aligned} P &\to AB \\ A &\to \mathtt{0}AZ \mid \lambda \\ B &\to \mathtt{1}B\mathtt{1} \mid X \\ Z\mathtt{1} &\to \mathtt{1}Z \\ ZX &\to X\mathtt{0} \\ X &\to \lambda \end{aligned}$$

   b) Gramática para $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid m < n < k\}$:

    $$\begin{aligned} P &\to \mathtt{a}BP\mathtt{c} \mid \mathtt{b}X\mathtt{c} \\ B\mathtt{a} &\to \mathtt{a}B \\ B\mathtt{b} &\to \mathtt{bb} \\ X &\to \mathtt{b}X\mathtt{c} \mid X\mathtt{c} \mid \mathtt{c} \end{aligned}$$

   c) Gramática para $\{www \mid w \in \{\mathtt{0},\mathtt{1}\}^*\}$:

$$
\begin{aligned}
P &\rightarrow \mathtt{0}PZ \mid \mathtt{1}PU \mid AB \\
\mathtt{0}Z &\rightarrow Z\mathtt{0} \\
\mathtt{1}Z &\rightarrow Z\mathtt{1} \\
\mathtt{0}U &\rightarrow U\mathtt{0} \\
\mathtt{1}U &\rightarrow U\mathtt{1} \\
BZ &\rightarrow ZB\mathtt{0} \\
BU &\rightarrow UB\mathtt{1} \\
AZ &\rightarrow A\mathtt{0} \\
AU &\rightarrow A\mathtt{1} \\
A &\rightarrow \lambda \\
B &\rightarrow \lambda
\end{aligned}
$$

2. Seja uma GI $G = (V, \Sigma, R, P)$ que contenha alguma regra cujo lado esquerdo só contém terminais. Uma GI equivalente em que toda regra contém variável do lado esquerdo seria $(V \cup \{N\}, \Sigma, R' \cup \{N \rightarrow \lambda\}, P)$, em que $N \notin V$ e R' seria como R, exceto que: para cada regra de N que contenha apenas terminais do lado esquerdo, escolher um terminal a que aparece do lado esquerdo (basta um) e substituir toda ocorrência de a em todas as regras por aN.

3. Basta substituir cada regra $u \rightarrow v$ em que $|u| > |v|$, u não é uma variável e $v \neq \lambda$, pela regra $u \rightarrow vX^{|u|-|v|}$, onde X é uma *variável nova* (basta *uma* variável nova X para a gramática que está sendo criada), e acrescentar a regra $X \rightarrow \lambda$.

4.
$$
\begin{aligned}
P &\rightarrow B\rangle \\
B &\rightarrow \mathtt{a}BA_1 \mid \mathtt{b}BA_2 \\
B &\rightarrow \langle 0 \\
A_1\rangle &\rightarrow \mathtt{a}\rangle \\
A_2\rangle &\rightarrow \mathtt{b}\rangle \\
A_1\mathtt{a} &\rightarrow \mathtt{a}A_1 \\
A_1\mathtt{b} &\rightarrow \mathtt{b}A_1 \\
A_2\mathtt{a} &\rightarrow \mathtt{a}A_2 \\
A_2\mathtt{b} &\rightarrow \mathtt{b}A_2
\end{aligned}
$$

$$
\begin{aligned}
0\mathtt{a} &\rightarrow \mathtt{a}1 \\
\mathtt{a}1\mathtt{b} &\rightarrow 0\mathtt{ab} \\
\mathtt{b}1\mathtt{b} &\rightarrow 0\mathtt{bb} \\
\langle 1\mathtt{b} &\rightarrow 0\langle \mathtt{b} \\
\sqcup 1\mathtt{b} &\rightarrow 0 \sqcup \mathtt{b}
\end{aligned}
$$

$$
\begin{aligned}
0\mathtt{b} &\rightarrow \# \\
0\langle &\rightarrow \langle \# \\
0\sqcup &\rightarrow \# \\
0\rangle &\rightarrow \#\rangle \\
1\mathtt{a} &\rightarrow \# \\
1\langle &\rightarrow \langle \#
\end{aligned}
$$

$1\sqcup \rightarrow \#$

$1\rangle \rightarrow \#\rangle$

$\#a \rightarrow \#$

$\#b \rightarrow \#$

$\#\sqcup \rightarrow \#$

$a\# \rightarrow \#$

$b\# \rightarrow \#$

$\sqcup\# \rightarrow \#$

$\langle\#\rangle \rightarrow \lambda$

5. a) Uma gramática com 4 regras que gera $\{\mathtt{a}^n\mathtt{b}^{n+1}\mathtt{c}^{n+2} \mid n \geq 0\}$:

$P \rightarrow \mathtt{a}PB\mathtt{c} \mid \mathtt{bcc}$
$\mathtt{c}B \rightarrow B\mathtt{c}$
$\mathtt{b}B \rightarrow \mathtt{bb}$

b) Uma gramática com 6 regras que gera $\{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \mid m < n < k\}$:

$P \rightarrow \mathtt{a}PB\mathtt{c} \mid PB\mathtt{c} \mid P\mathtt{c} \mid \mathtt{bcc}$
$\mathtt{c}B \rightarrow B\mathtt{c}$
$\mathtt{b}B \rightarrow \mathtt{bb}$

6. Um ALL pode ser construído a partir daquele do Exemplo 130, mostrado na Figura 4.16 do livro, aproveitando-se os estados 1 a 6, com as respectivas tranisções, acrescentando-se 7 estados (numerados de 7 a 13) e as seguintes transições:

$\delta(5, \langle) = [7, \langle, \mathtt{D}]$	$\delta(7, \sqcup) = [7, \sqcup, \mathtt{D}]$	$\delta(7, \mathtt{b}) = [8, \sqcup, \mathtt{D}]$
$\delta(8, \mathtt{b}) = [8, \mathtt{b}, \mathtt{D}]$	$\delta(8, \sqcup) = [9, \sqcup, \mathtt{D}]$	$\delta(9, \sqcup) = [9, \sqcup, \mathtt{D}]$
$\delta(9, \mathtt{c}) = [10, \sqcup, \mathtt{E}]$	$\delta(10, \sqcup) = [10, \sqcup, \mathtt{E}]$	$\delta(10, \mathtt{b}) = [9, \sqcup, \mathtt{D}]$
$\delta(10, \langle) = [11, \langle, \mathtt{D}]$	$\delta(11, \sqcup) = [11, \sqcup, \mathtt{D}]$	$\delta(11, \mathtt{c}) = [12, \sqcup, \mathtt{D}]$
$\delta(12, \mathtt{c}) = [12, \sqcup, \mathtt{D}]$	$\delta(12, \langle) = [13, \langle, \mathtt{D}]$	

O ALL tem um único estado final: 13.

7. Sim, a linguagem $\{ww \mid w \in \{\mathtt{0}, \mathtt{1}\}^+\}$ é uma LSC. Uma GSC para ela seria:

$P \rightarrow \mathtt{0}PZ \mid \mathtt{1}PU \mid F_0\mathtt{0} \mid F_1\mathtt{1}$

$\mathtt{0}Z \rightarrow Z\mathtt{0}$

$\mathtt{0}U \rightarrow U\mathtt{0}$

$\mathtt{1}Z \rightarrow Z\mathtt{1}$

$\mathtt{1}U \rightarrow U\mathtt{1}$

$F_0Z \rightarrow F_0\mathtt{0}$

$F_0U \rightarrow F_0\mathtt{1}$

$F_1Z \rightarrow F_1\mathtt{0}$

$F_1U \rightarrow F_1\mathtt{1}$

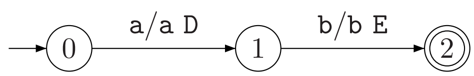
$F_0 \rightarrow \mathtt{0}$

$F_1 \rightarrow \mathtt{1}$

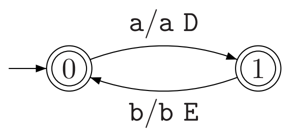
8. Seja $G = (V, \Sigma, R, P)$ uma GI que gere L. Uma GSC que gera L' é $(V \cup \{P', Z\}, \Sigma \cup \{\#\}, R', P')$, em que $P', Z \notin V$ e R' contém:
   - a regra $P' \to P\#$;
   - cada regra $u \to v$ de R tal que $|u| \leq |v|$;
   - para cada regra $u \to v$ de R tal que $|u| > |v|$, a regra $u \to vZ^{|u|-|v|}$;
   - para cada $X \in V \cup \Sigma$, a regra $ZX \to XZ$; e
   - a regra $Z\# \to \#\#$.

4.4 Propriedades das LREs e das Linguagens Recursivas

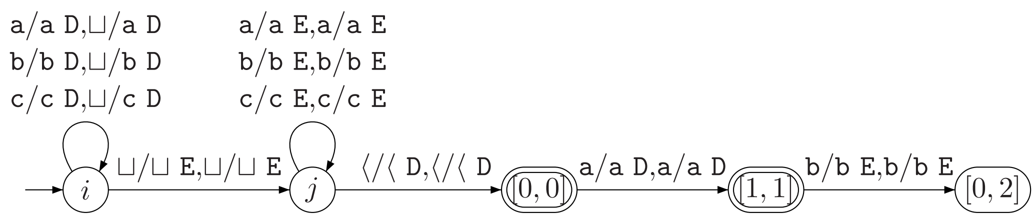
1. Uma MT para $L = \{\texttt{ab}\}\{\texttt{a}, \texttt{b}, \texttt{c}\}^*$:

   Uma MT para $\overline{L}$:

   Aplicando-se o método do Teorema 36, obtém-se a seguinte MT para $\overline{L}$:

2. Seja uma GLC $G = (V, \Sigma, R, P)$. Será mostrado como construr uma MT de duas fitas que aceita $L(G)$ e que sempre pára. O conteúdo da fita 2 será da forma:

   $$\langle P\#x_1\#x_2\#\ldots\#x_n \sqcup \ldots$$

   que codifica a uma derivação $P \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \ldots \Rightarrow x_n$. Segue o algoritmo da MT:

   Escreva P (a variável de partida) na fita 2.
   ciclo
       selecione uma posição p na última forma sentencial que está na fita 2;
       selecione uma regra $u \to v \in R$;
       se u ocorre a partir da posição p da fita 2 **então**
           copie na fita 2 a última forma sentencial substituindo u por v;
           se a última forma sentencial na fita 2 aparece anteriormente **então**
               rejeite
           senãose a última forma sentencial na fita 2 é maior que a palavra
               de entrada na fita 1 **então**
               rejeite

senãose a última forma sentencial na fita 2 é idêntica à palavra
de entrada na fita 1 **então**
aceite
fimse
senão
rejeite
fimse
fimciclo.

Como o conjunto de derivações possíveis de formas sentenciais x_n tais que $|x_n| \leq |w|$ e $x_i \neq x_j$ para $i \neq j$ é finito (já que $V \cup \Sigma$ é finito), e o algoritmo corta derivações que não satisfaçam essas condições, conclui-se que a MT sempre pára. Logo, $L(G)$ é recursiva. E como as linguagens recursivas são fechadas sob complementação, $\overline{L(G)}$ também é recursiva.

3. a) Suponha que $\overline{L}$ seja recursiva. Então, como as linguagens recursivas são fechadas sob complementação, $\overline{\overline{L}} = L$ é recursiva. Mas L não é recursiva! Logo, $\overline{L}$ não pode ser recursiva.

   b) Se L e $\overline{L}$ fossem ambas LREs, então, pelo Teorema 36, L seria recursiva. Mas L não é recursiva! Logo, se L é LRE, $\overline{L}$ não é LRE.

4. a) Como as linguagens recursivas são fechadas sob complementação, $\overline{R}$ é recursiva. Assim, $\overline{R}$ é LRE. Como as LREs são fechadas sob interseção, segue-se que $L \cap \overline{R}$ é LRE. Como $L - R = L \cap \overline{R}$, $L - R$ é uma LRE.

   b) $\emptyset$ é recursiva. E se L não é recursiva, $L - \emptyset = L$ não é recursiva.

   c) Σ^* é recursiva. E $\Sigma^* - L$ pode não ser uma LRE, pois as LREs não são fechadas sob complementação.

5. Para mostrar o fechamento sob concatenação, sejam M_1 e M_2 duas MTs. Será mostrado como construir uma MT não determinística de duas fitas que reconhece $L(M_1)L(M_2)$. A MT tem os seguintes passos:

   1. Copie na fita 2 um sufixo y da palavra de entrada w e apague tal sufixo da fita 1. Após isto, a fita 1 conterá $\langle x \sqcup \ldots$ e a fita 2 conterá $\langle y \sqcup \ldots$, sendo $xy = w$. O sufixo y é escolhido *não deterministicamente*.
   2. Simule M_1 sobre a fita 1 (entrada x) deixando o cabeçote da fita 2 imóvel. Sempre que M_1 pare em estado final, coloque uma transição para o início da simulação de M_2 (próximo passo).
   3. Simule M_2 sobre a fita 2 (entrada y) deixando o cabeçote da fita 1 imóvel. Sempre que M_1 pare em estado final, aceite.

   Para mostrar o fechamento sob fecho de Kleene, seja M uma MT. Será mostrado como construir uma MT não determinística de duas fitas que reconhece $L(M)^*$. A MT tem os seguintes passos:

   1. Se a palavra de entrada for λ, aceite.
   2. Copie na fita 2 um prefixo $x \neq \lambda$ da palavra da fita 1 e apague-o da fita 1, deixando o sufixo y (que pode ser λ) restante na fita 1. O prefixo x é escolhido *não determinísticamente*.
   3. Simule M sobre a fita 2 (entrada x) deixando o cabeçote da fita 1 imóvel. Sempre que M pare em estado final, coloque uma transição para o teste do próximo passo.

4. Se $y = \lambda$, aceite; se $y \neq \lambda$, volte ao passo 2.

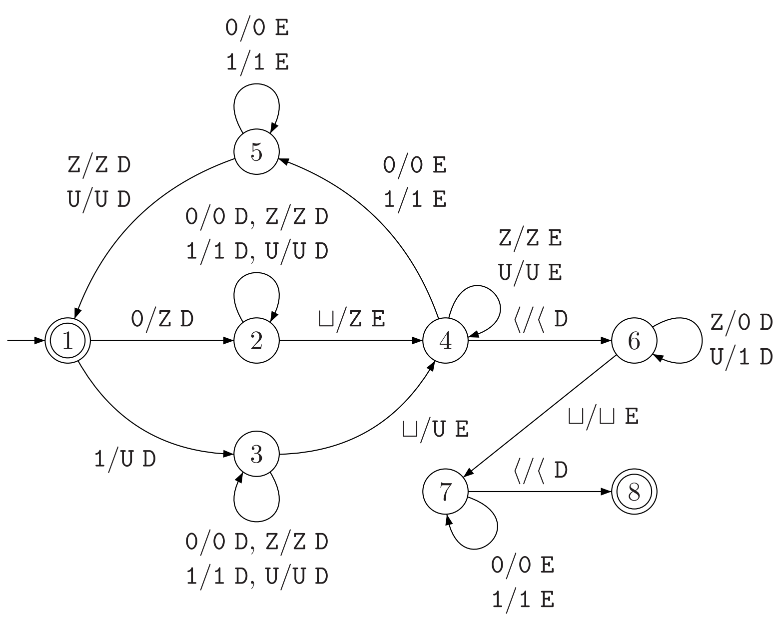
6. Suponha que o conjunto de todas as MTs cujo alfabeto de entrada é $\{0, 1\}$ é enumerável, e seja uma enumeração qualquer: M_0, M_1,... . Seja também uma enumeração das palavras de $\{0, 1\}^*$: w_0, w_1,... . Pode-se definir a linguagem D tal que:

$$w_i \in D \text{ se, e somente se, } w_i \notin L(M_i)$$

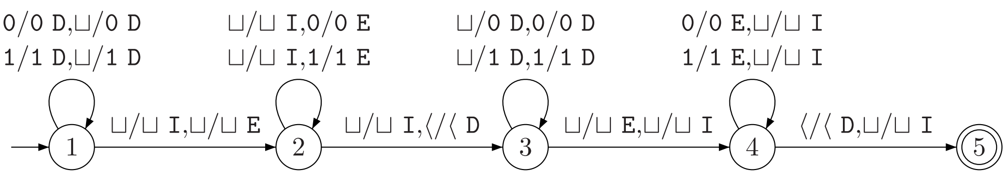
Ora, se alguma MT M_k reconhecesse D (neste caso, $L(M_k) = D$), ela seria tal que $w_k \in L(M_k)$ se, e somente se, $w_k \notin L(M_k)$! Contradição. Logo, já que o conjunto das MTs é enumerável, não há MT que reconhece D e, portanto, D não é LRE.

4.5 Exercícios

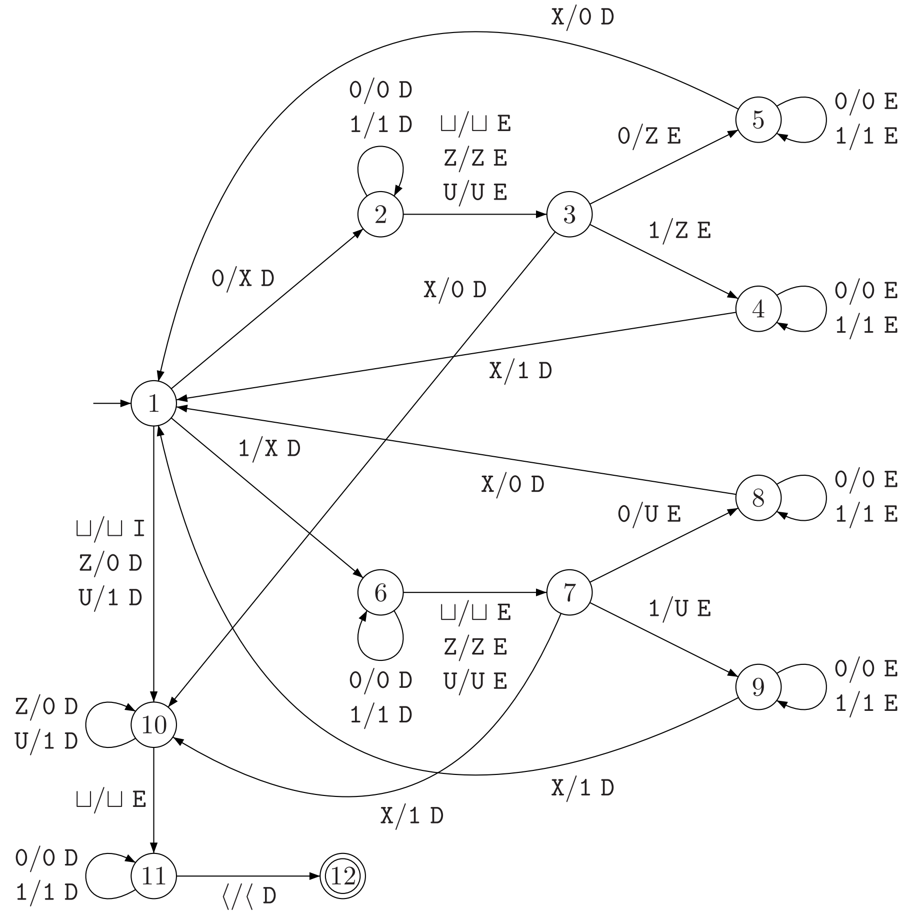
1. a) Uma MT-padrão para $f(w) = w^2$:



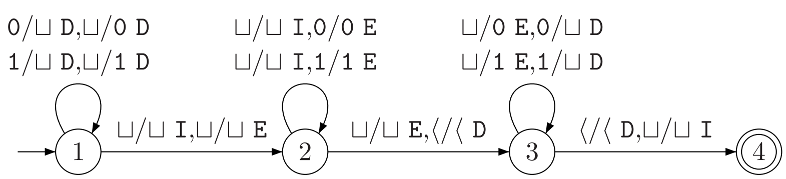
Uma MT de duas fitas para $f(w) = w^2$:



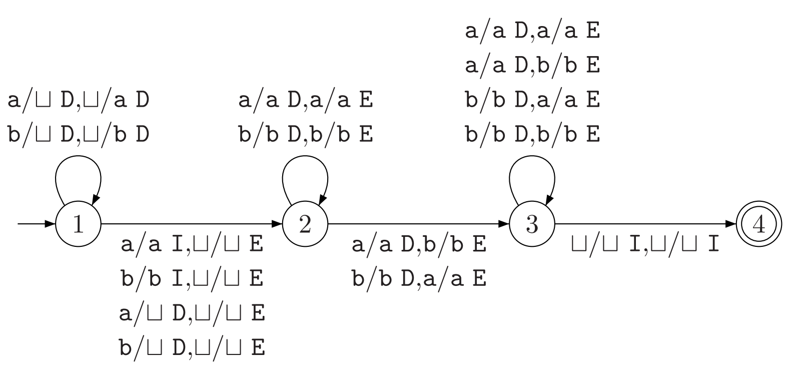
b) Uma MT-padrão para $f(w) = w^R$:



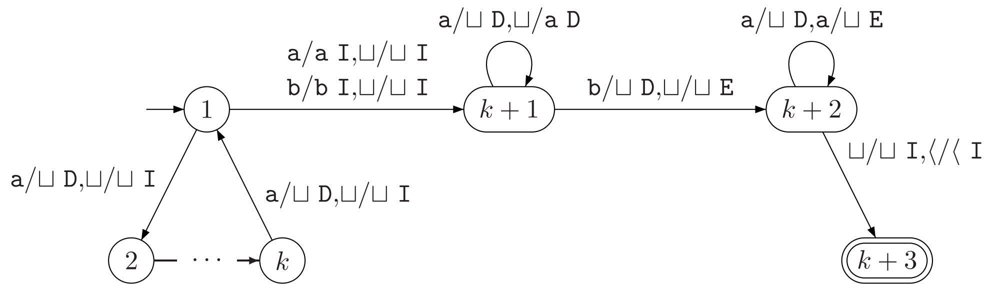
Uma MT de duas fitas para $f(w) = w^R$:



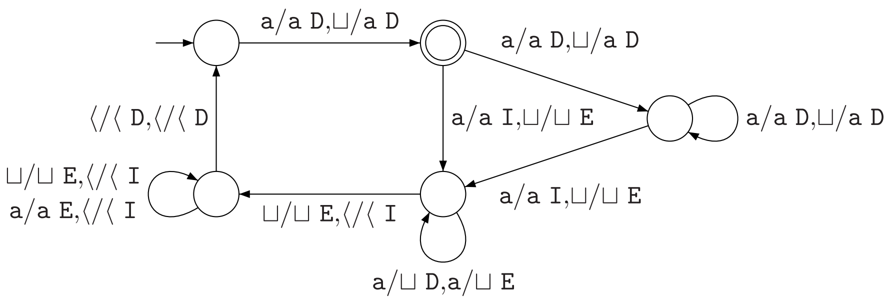
2. **a)** MT de duas fitas não determinística para $\{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \neq w^R\}$:



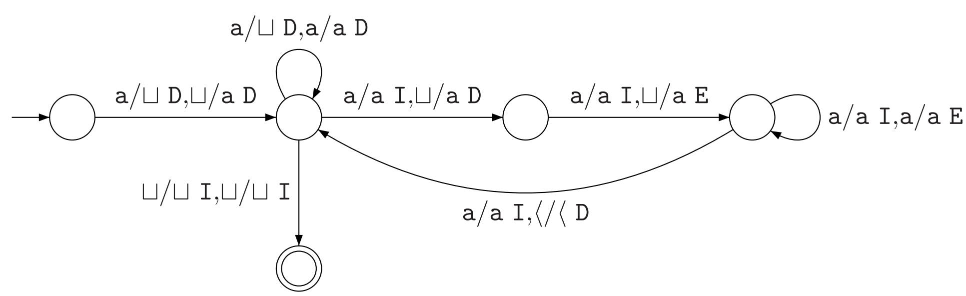
b) MT de duas fitas não determinística para $\{\mathtt{a}^m\mathtt{b}\mathtt{a}^n \mid m - n \text{ é divisível por } k\}$, onde k é uma constante positiva:



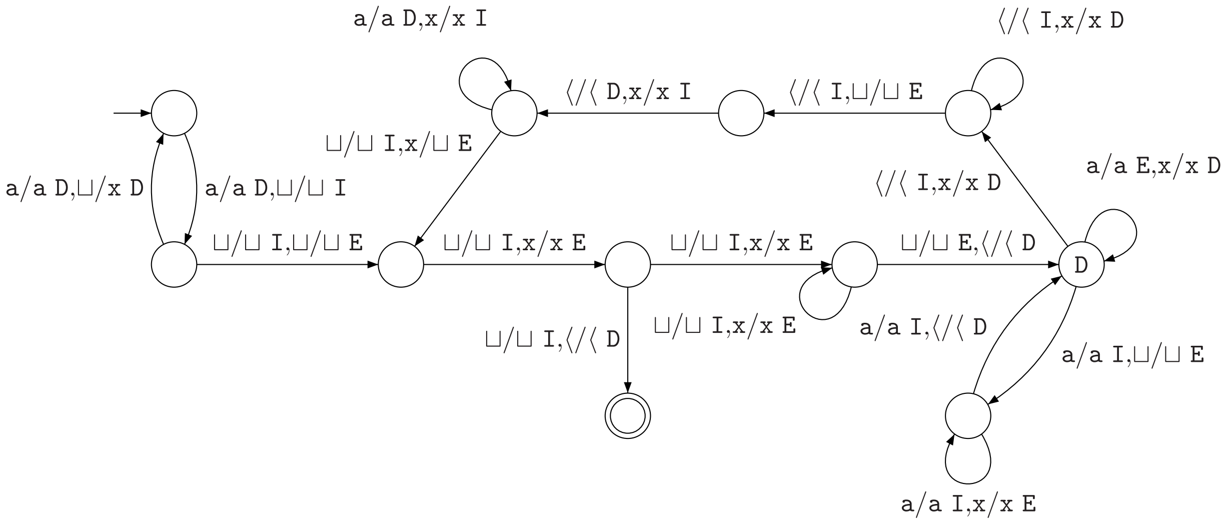
c) MT de duas fitas não determinística para $\{\mathtt{a}^{2^n} \mid n \geq 0\}$:



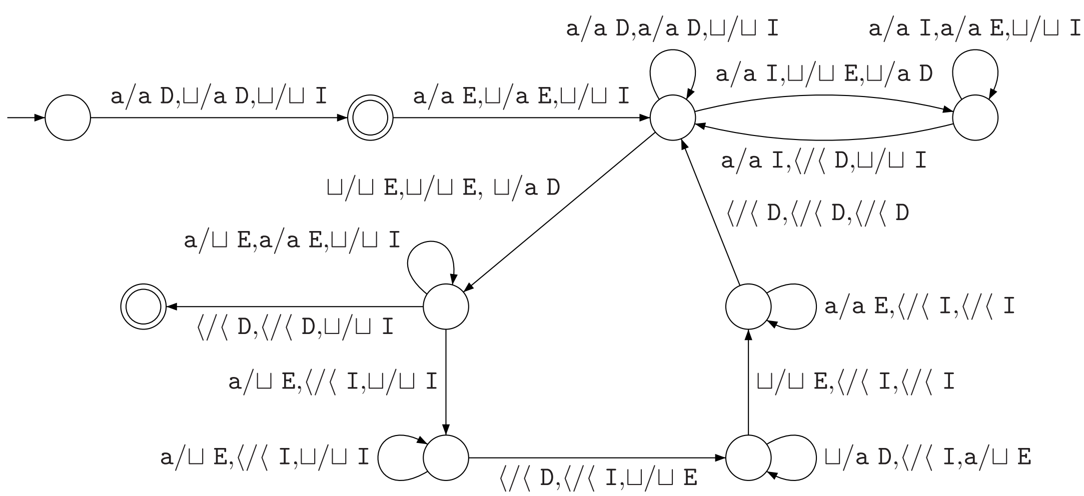
d) Segue uma MT de duas fitas para $\{\mathtt{a}^{n^2} \mid n \geq 0\}$, baseada em $\sum_{k=1}^{n}(2k-1) = n^2$. Os números ímpares são gerados sucessivamente na fita 2 (em unário) e subtraídos da palavra de entrada.



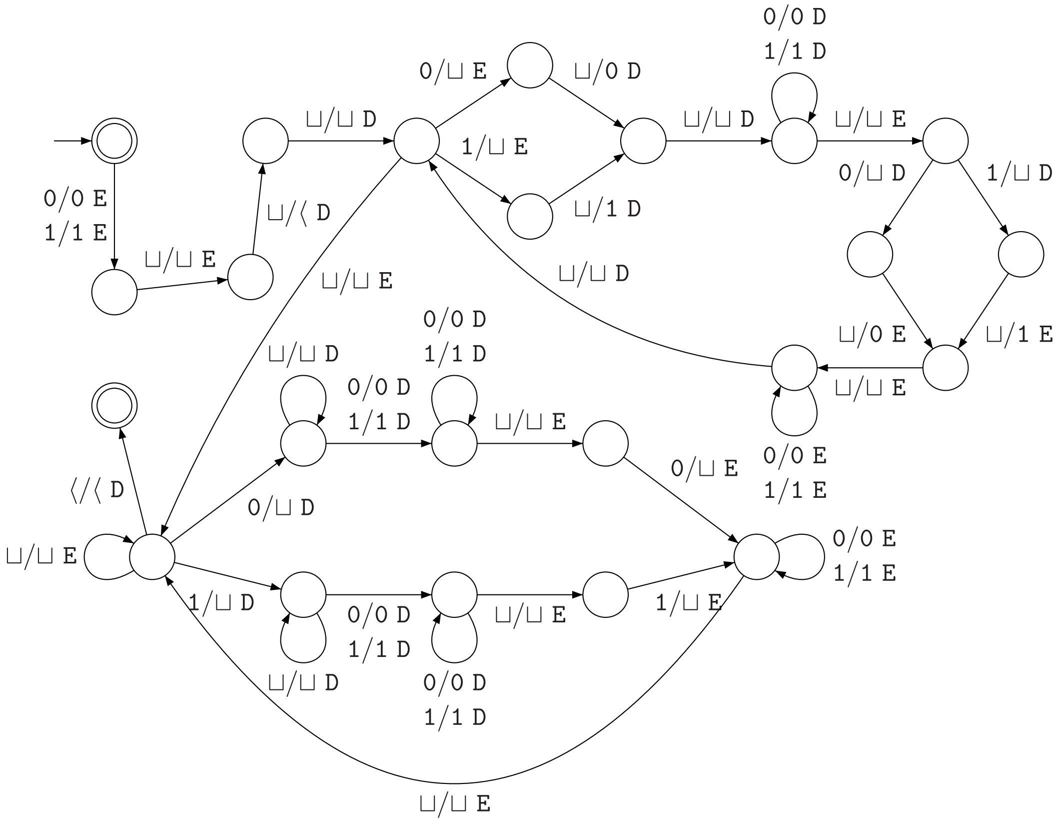
e) Segue uma MT de duas fitas para $\{\mathtt{a}^n \mid n \text{ é primo}\}$. Se o número de as da fita de entrada for ímpar, é armazenado $\lfloor n/2 \rfloor$ xs na fita 2. Enquanto o número de as não for divisível pelo número da fita 2, o da fita 2 é decrementado. Só quando o número da fita 2 atinge 1, o estado final é atingido. Se o número da fita 1 for divisível por algum número, a MT pára no estado D.



f) Uma MT de três fitas para $\{\mathtt{a}^{n!} \mid n \geq 0\}$:



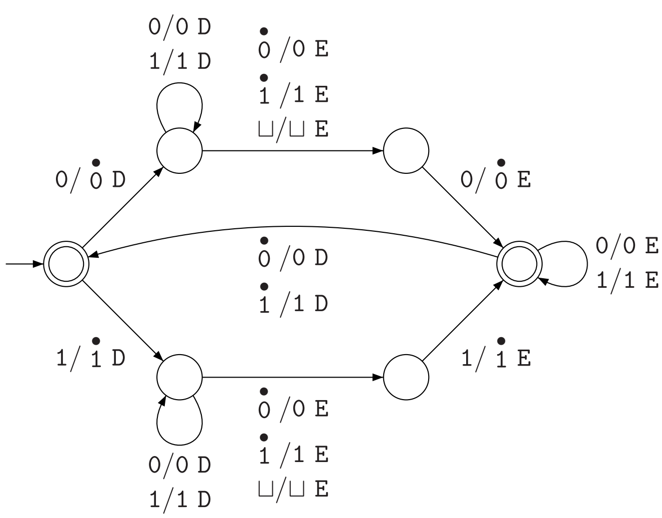
3.



4. Toda linguagem que pode ser reconhecida por MT desse tipo é, obviamente, recursiva. Por outro lado, se uma linguagem é recursiva, pode ser reconhecida por MT desse tipo: seja uma MT-padrão que sempre pare $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$; uma MT do tipo em consideração que reconhece $L(M)$ é $(E \cup \{e_{sim}, e_{não}\}, \Sigma, \Gamma, \langle, \sqcup, \delta', i)$, em que para todo $(e, a) \in E \times \Gamma$:

   - se $\delta(e, a)$ é definido, então $\delta'(e, a) = \delta(e, a)$;
   - se $\delta(e, a)$ é indefinido, então:
     - se $e \in F$, $\delta'(e, a) = e_{sim}$;
     - se $e \notin F$, $\delta'(e, a) = e_{não}$.

5. O único símbolo a ser escrito em células da segunda trilha será o "$\bullet$". O par $[a, \bullet]$ sob o cabeçote (a na primeira trilha e "$\bullet$" na segunda) será referido por $\dot{a}$, o par $[a, \sqcup]$ será referido por a e o par $[\sqcup, \sqcup]$ será referido por $\sqcup$ simplesmente, por conveniência. Segue a MT:



6. Seja um APN $M = (E, \Sigma, \Gamma, \delta, I, F)$. Uma MT não determinística de duas fitas equivalente seria $M' = (E \cup \{f\}, \Sigma, \Sigma \cup \Gamma \cup \{\langle, \sqcup\}, \langle, \sqcup, \delta', \{i\}, \{i, f\})$, em que $i, f \notin E$ e δ' é assim obtida a partir de δ;

   - $\delta'(i, c, \sqcup) = \{[e, (c, \mathtt{I}), (\sqcup, \mathtt{E})] \mid e \in I\}$ para cada $c \in \Sigma \cup \{\sqcup\}$ (pilha vazia: cabeçote da fita 2 na primeira posição);
   - para cada transição de M da forma

     $$e \xrightarrow{a,\lambda/\lambda} e'$$

     fazer:

     – se $a = \lambda$: $e \xrightarrow{c/c\ \mathtt{I},d/d\ \mathtt{I}} e'$ para cada $c \in \Sigma \cup \{\sqcup\}$ e cada $d \in \Gamma \cup \{\langle\}$.

     – se $a \neq \lambda$: $e \xrightarrow{a/a\ \mathtt{D},d/d\ \mathtt{I}} e'$ para cada $d \in \Gamma \cup \{\langle\}$.
   - para cada transição de M da forma

     $e \xrightarrow{a,\lambda/z} e'$ em que $z = z_1 \ldots z_n$

     fazer:

     – se $a = \lambda$: $e \xrightarrow{c/c\ \mathtt{I},d/d\ \mathtt{D}} e_1 \xrightarrow{c/c\ \mathtt{I},\sqcup/z_n\ \mathtt{D}} e_2 \cdots e_n \xrightarrow{c/c\ \mathtt{I},\sqcup/z_1\ \mathtt{I}} e'$
     para cada $c \in \Sigma \cup \{\sqcup\}$ e cada $d \in \Gamma \cup \{\langle\}$;

     – se $a \neq \lambda$: $e \xrightarrow{a/a\ \mathtt{I},d/d\ \mathtt{D}} e_1 \xrightarrow{a/a\ \mathtt{I},\sqcup/z_n\ \mathtt{D}} e_2 \cdots e_n \xrightarrow{a/a\ \mathtt{D},\sqcup/z_1\ \mathtt{I}} e'$
     para cada $d \in \Gamma \cup \{\langle\}$.
   - para cada transição de M da forma

     $$e \xrightarrow{a,b/\lambda} e'$$

     fazer:

     – se $a = \lambda$: $e \xrightarrow{c/c\ \mathtt{I},b/\sqcup\ \mathtt{E}} e'$ para cada $c \in \Sigma \cup \{\sqcup\}$;

     – se $a \neq \lambda$: $e \xrightarrow{a/a\ \mathtt{D},b/\sqcup\ \mathtt{E}} e'$
   - para cada transição de M da forma

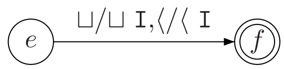
     $e \xrightarrow{a,b/z} e'$ em que $z = z_1 \ldots z_n$

     fazer:

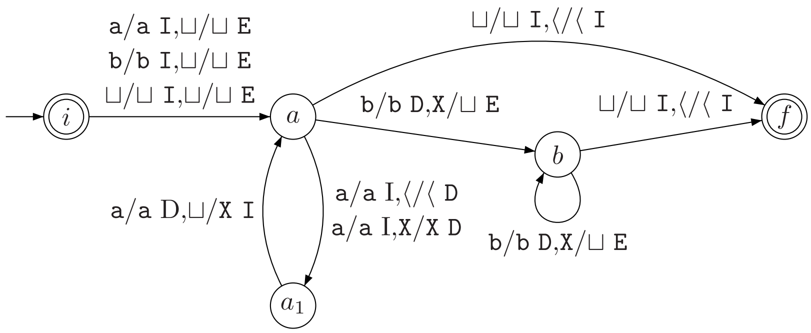
     – se $a = \lambda$: $e \xrightarrow{c/c\ \mathtt{I},b/z_n\ \mathtt{D}} e_1 \xrightarrow{c/c\ \mathtt{I},\sqcup/z_{n-1}\ \mathtt{D}} e_2 \cdots e_{n-1} \xrightarrow{c/c\ \mathtt{I},\sqcup/z_1\ \mathtt{I}} e'$
     para cada $c \in \Sigma \cup \{\sqcup\}$;

     – se $a \neq \lambda$: $e \xrightarrow{a/a\ \mathtt{I},b/z_n\ \mathtt{D}} e_1 \xrightarrow{a/a\ \mathtt{I},\sqcup/z_{n-1}\ \mathtt{D}} e_2 \cdots e_{n-1} \xrightarrow{a/a\ \mathtt{D},\sqcup/z_1\ \mathtt{I}} e'$

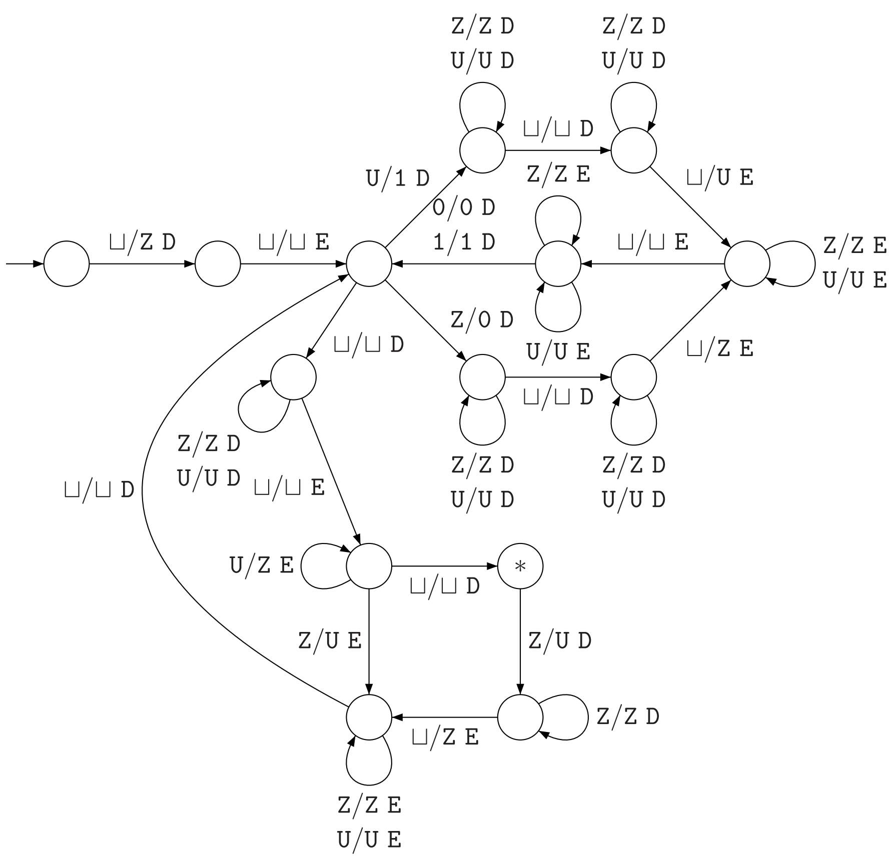
- Finalmente, para cada $e \in F$, fazer:



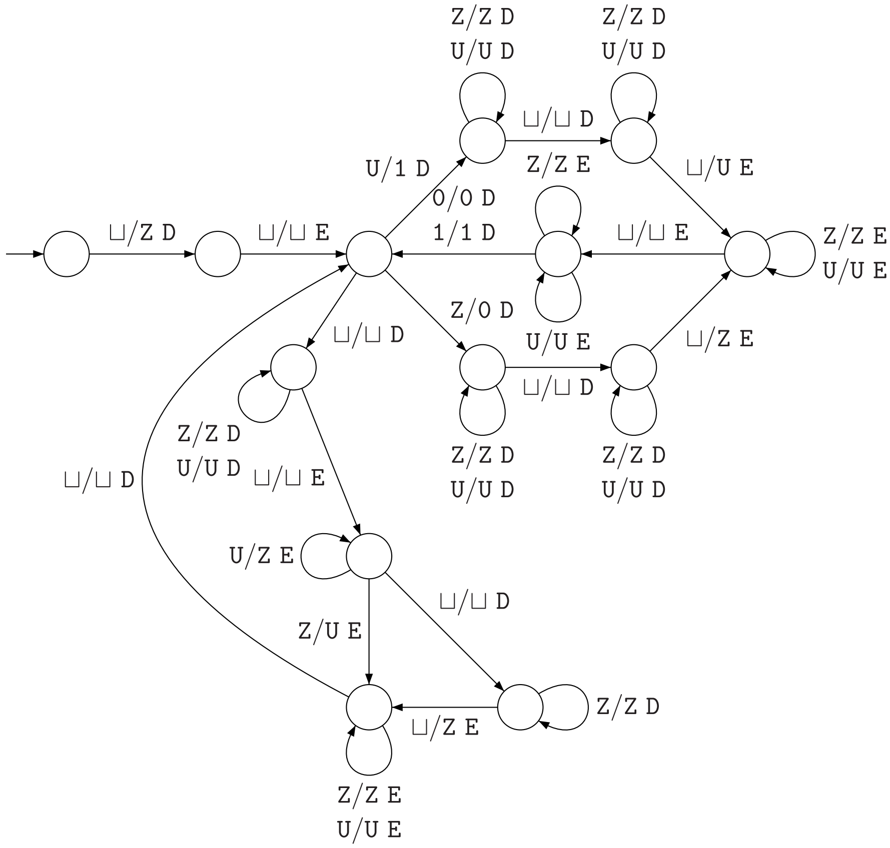
Segue a MT obtida a partir do APD da Figura 3.4, página 151 do livro:



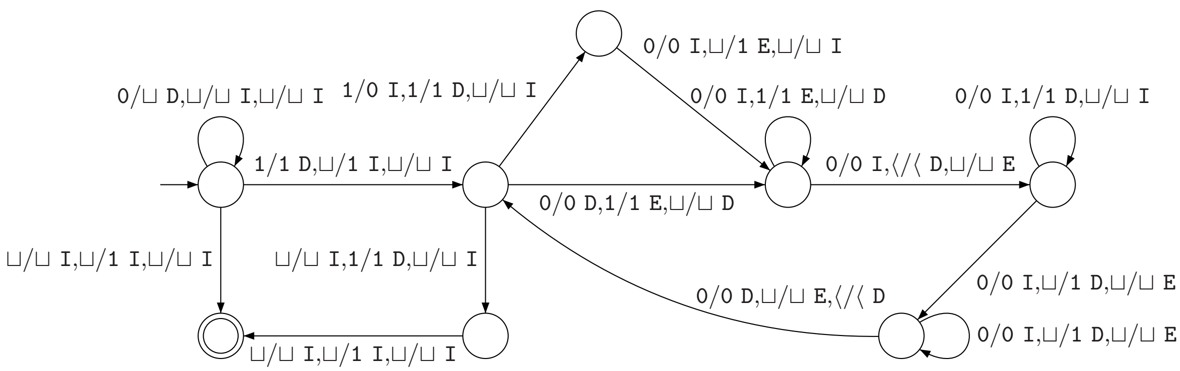
7.



8. A MT é quase idêntica à da questão anterior; basta eliminar o estado $*$, obtendo-se:



9. MT com 3 fitas. A fita 2 conterá a saída e a fita auxiliará na duplicação da palavra na fita 2. O número n é representado por $\mathtt{1}^{n+1}$.



10. Seja uma MT $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ do tipo definido. Uma MT-padrão que simula M seria $M' = (E \cup \{e_{\mathsf{x}}\} \cup N, \Sigma, \Gamma, \langle, \sqcup, \delta', i, F)$, $e_{\mathsf{x}} \notin E$, $N \cap E = \emptyset$, tal que para cada transição da forma $\delta(e, a) = [e', b, \mathtt{D}, n]$ há as transições ($e_1, \ldots, e_{n-1} \in N$ são $n-1$ estados intermediários *novos*):

- $\delta'(e, a) = [e_1, b, \mathtt{D}]$, $\delta'(e_1, c) = [e_2, c, \mathtt{D}]$ para cada $c \in \Gamma - \{\langle\}$, $\ldots$, $\delta'(e_{n-1}, c) = [e', c, \mathtt{D}]$ para cada $c \in \Gamma - \{\langle\}$;

para cada transição da forma $\delta(e, a) = [e', b, \mathtt{E}, n]$ há as transições ($e_1, \ldots, e_{n-1} \in N$ são $n-1$ estados intermediários *novos*):

- $\delta'(e, a) = [e_1, b, \mathtt{E}]$, $\delta'(e_1, c) = [e_2, c, \mathtt{E}]$ para cada $c \in \Gamma - \{\langle\}$, $\ldots$, $\delta'(e_{n-1}, c) = [e', c, \mathtt{E}]$ para cada $c \in \Gamma - \{\langle\}$;

- e para cada $i = 1, \ldots, n-1$, $\delta'(e_i, \langle) = [e_{\mathtt{x}}, \langle, \mathtt{D}]$.

Há apenas essas transições em M'.

11. Seja uma máquina de duas fitas $M = (E, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ e $\mathrm{X} \notin \Gamma$, como referidos na Seção 4.2.4. A MT M' de quatro trilhas que simula M, como lá esboçado, começa escrevendo $\langle$ no início da trilha 3, escrevendo as representações dos cabeçotes nas trilhas 2 e 4, e transitando para o estado $[i, ------]$, da seguinte forma, sendo i' o estado inicial de M' e j mais um estado auxiliar:

    - $\delta'(i', \langle, \sqcup, \sqcup, \sqcup) = [j, \langle, \sqcup, \langle, \sqcup, \mathtt{D}]$;
    - $\delta'(j, a, \sqcup, \sqcup, \sqcup) = [[i, ------], \mathtt{X}, \sqcup, \mathtt{X}, \sqcup, \mathtt{I}]$ para cada $a \in \Gamma - \{\langle\}$.

    Para cada estado da forma $[e, ------]$, M' busca os símbolos a_1 e a_2 das trilhas 1 e 3 cujas posições estão marcadas pelos símbolos $\mathtt{X}$ das trilhas 2 e 4 e transita para o estado $[e, a_1a_2----]$, assim:

    - $\delta'([e, ------], a_1, \sqcup, a_2, \sqcup) = [[e, ------], a_1, \sqcup, a_2, \sqcup, \mathtt{D}]$ para $a_1, a_2 \in \Gamma$;
    - $\delta'([e, ------], a_1, \mathtt{X}, a_2, \sqcup) = [[e, a_1-----], a_1, \mathtt{X}, a_2, \sqcup, \mathtt{D}]$ para $a_1, a_2 \in \Gamma$;
    - $\delta'([e, ------], a_1, \sqcup, a_2, \mathtt{X}) = [[e, -a_2----], a_1, \sqcup, a_2, \mathtt{X}, \mathtt{D}]$ para $a_1, a_2 \in \Gamma$;
    - $\delta'([e, ------], a_1, \mathtt{X}, a_2, \mathtt{X}) = [[e, a_1a_2----], a_1, \mathtt{X}, a_2, \mathtt{X}, \mathtt{I}]$ para $a_1, a_2 \in \Gamma$;
    - $\delta'([e, a_1-----], a_1', \sqcup, a_2, \sqcup) = [[e, a_1-----], a_1', \sqcup, a_2, \sqcup, \mathtt{D}]$ para $a_1, a_1', a_2 \in \Gamma$;
    - $\delta'([e, a_1-----], a_1', \sqcup, a_2, \mathtt{X}) = [[e, a_1a_2----], a_1', \sqcup, a_2, \mathtt{X}, \mathtt{I}]$ para $a_1, a_1', a_2 \in \Gamma$;
    - $\delta'([e, -a_2----], a_1, \sqcup, a_2', \sqcup) = [[e, -a_2----], a_1, \sqcup, a_2', \sqcup, \mathtt{D}]$ para $a_1, a_2, a_2' \in \Gamma$;
    - $\delta'([e, -a_2----], a_1, \mathtt{X}, a_2', \sqcup) = [[e, a_1a_2----], a_1, \mathtt{X}, a_2', \sqcup, \mathtt{I}]$ para $a_1, a_2, a_2' \in \Gamma$.

    Para cada estado da forma $[e, a_1a_2----]$, se $\delta(e, a_1, a_2)$ é indefinido, $\delta'([e, a_1a_2----], a_1', c_1, a_2', c_2)$ é indefinido para todo $a_1', c_1, a_2, c_2 \in \Gamma \cup \{\mathtt{X}\}$. Mas se $\delta(e, a_1, a_2) = [e', b_1, d_1, b_2, d_2]$, M' busca, movendo seu cabeçote da direita para a esquerda, os símbolos a_1 e a_2 das trilhas 1 e 3 cujas posições estão marcadas pelos símbolos $\mathtt{X}$ das trilhas 2 e 4, assim:

    - $\delta'([e, a_1a_2----], a_1', \sqcup, a_2', \sqcup) = [[e, a_1a_2----], a_1', \sqcup, a_2', \sqcup, \mathtt{E}]$ para $a_1', a_2' \in \Gamma$;
    - $\delta'([e, a_1a_2----], a_1, \mathtt{X}, a_2', \sqcup) = [[e, a_1a_2b_1---], b_1, \sqcup, a_2', \sqcup, d_1]$ para $a_2' \in \Gamma$;
    - $\delta'([e, a_1a_2b_1---], a_1', \sqcup, a_2', c) = [[e, a_1a_2b_1d_1--], a_1', \mathtt{X}, a_2', c, \mathtt{I}]$ para $a_1', a_2' \in \Gamma$ e $c \in \{\sqcup, \mathtt{X}\}$;
    - $\delta'([e, a_1a_2b_1d_1--], a_1', \sqcup, a_2', \sqcup) = [[e, a_1a_2b_1d_1--], a_1', \sqcup, a_2', \sqcup, \mathtt{E}]$ para $a_1', a_2' \in \Gamma$;
    - $\delta'([e, a_1a_2b_1d_1--], a_1', \sqcup, a_2, \mathtt{X}) = [[e, a_1a_2b_1d_1b_2-], a_1', \sqcup, a_2, \sqcup, d_2]$ para $a_1' \in \Gamma$;
    - $\delta'([e, a_1a_2b_1d_1b_2-], a_1', \sqcup, a_2', \sqcup) = [[e, a_1a_2b_1d_1b_2d_2], a_1', \sqcup, a_2', \mathtt{X}, \mathtt{E}]$ para $a_1', a_2' \in \Gamma$;
    - $\delta'([e, a_1a_2b_1d_1b_2d_2], a_1', \sqcup, a_2', \sqcup) = [[e, a_1a_2b_1d_1b_2d_2], a_1', \sqcup, a_2', \mathtt{X}, \mathtt{E}]$ para $a_1', a_2' \in \Gamma$;
    - $\delta'([e, a_1a_2b_1d_1b_2d_2], \langle, \sqcup, \langle, \sqcup) = [[e', ------], \langle, \sqcup, \langle, \sqcup, \mathtt{D}]$.

    Substitui a_1 por b_1 e a_2 por b_2, e move os símbolos $\mathtt{X}$ das trilhas 2 e 4 nas direções d_1 e d_2. Feito isso, M' transita para o estado $[e, a_1a_2b_1d_1b_2d_2]$. Nesse estado, M' volta ao início da fita e transita para o estado $[e', ------]$.

    Os estados finais de M' são os estados $[e, a_1a_2----]$ para $a_1, a_2 \in \Gamma$ e $e \in F$.

    Para concretizar a MT M' esboçada anteriormente, basta acrescentar novos estados da forma $[e, x_1x_2y_1d_1y_2d_2]$, onde $x_1, x_2, y_1, y_2 \in \Gamma \cup \{-\}$ e $d_1, d_2 \in \{\mathtt{D}, \mathtt{E}, \mathtt{I}\}$, à medida que forem necessários. Por exemplo, se a representação `X` do cabeçote da fita 1 for encontrada antes daquela da fita 2 quando M' procura por a_1 e a_2 da esquerda para a direita, pode-se fazer que M' transite para $[e, a_1 - - - - -]$. Nesse estado, M' procura por a_2; e, ao achá-lo, transita para o estado referido no parágrafo anterior, $[e, a_1 a_2 - - - -]$. Entretanto, se a representação do cabeçote da fita 2 for encontrada antes daquela da fita 1, M' transita para $[e, -a_2 - - - -]$; depois, ao encontrar a_1, transita para $[e, a_1 a_2 - - - -]$ e assim por diante.

Só existem as transições explicitas acima.