Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

Não são aceitas respostas sem justificativa. Explique tudo o que você fizer.

Linguagens Formais—1º semestre de 1999 –Primeira Prova

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_4\}$, estado inicial q_1, estados finais $\{q_1\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_1	q_3
q_2	q_1	q_3
q_3	q_2	q_3
q_4	q_4	q_5

(a) Esboce o grafo de M.

(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(2) Considere as seguintes linguagens no alfabeto $\{0, 1\}$:

- L_1 é o conjunto das palavras em $\{0, 1\}^*$ nas quais a seqüência 00 aparece em algum lugar da palavra.
- L_2 é o conjunto das palavras em $\{0, 1\}^*$ que contêm um único 1 (mas podem conter um número qualquer de zeros!).

(a) Dê exemplos de um autômato finito determinístico que aceite L_1 e de um autômato finito determinístico que aceite L_2.

(b) Use os algoritmos dados em aula para construir um autômato finito não determinístico que aceite $L_1 \cup L_2$.

(c) Use os algoritmos dados em aula para construir um autômato finito não determinístico que aceite $L_1 \cdot L_2$.

(3) Considere as seguintes linguagens no alfabeto $\{0, 1\}$:

- L_1 é a linguagem que consiste das palavras em $\{0, 1\}^*$ em que o número de 0s é maior que o número de 1s;
- L_2 é a linguagem que consiste das palavras em $\{0, 1\}^*$ cujo número de 0s é múltiplo de 3.

Uma destas linguagens é regular, a outra não. Dê uma expressão regular que denote a linguagem regular, e mostre que a outra linguagem não é regular usando o lema do bombeamento.

(4) Seja $w = 0^n$ uma palavra em Σ^*, constituída de n 0s consecutivos. Vamos definir um autômato finito não determinístico $\mathcal{A}_n$ cuja construção *depende* do valor do inteiro $n \geq 0$ escolhido. Os ingredientes de $\mathcal{A}_n$ são os seguintes:

- o alfabeto é Σ;
- o conjunto de estados é $Q = \{q_1, \ldots, q_{n+1}\}$;
- o estado inicial é q_1;
- o conjunto de estados finais é $\{q_{n+1}\}$;
- a função de transição $\delta : Q \times \Sigma \to 2^Q$ é dada por:

$$\delta(q_i, 0) = \begin{cases} \{q_1, q_2\} & \text{se } i = 1 \\ \{q_{i+1}\} & \text{se } 1 < i \leq n+1 \end{cases} \quad \text{e} \quad \delta(q_i, 1) = \begin{cases} \{q_1\} & \text{se } i = 1 \\ \emptyset & \text{se } 1 < i < n+1 \end{cases}$$

Observe que o número de estados de $\mathcal{A}_n$ depende do inteiro n escolhido.

(a) Esboce o grafo de $\mathcal{A}_n$.

(b) Use o algoritmo dado em aula para construir um autômato finito determinístico $\mathcal{D}_n$ que aceite a mesma palavra que $\mathcal{A}_n$. Se não consideramos os estados redundantes, quantos são os estados de $\mathcal{D}_n$?

Linguagens Formais—1º semestre de 1999 –Segunda Prova

(1) Mostre que a gramática livre de contexto G cujas regras são

$$\begin{array}{lll} S \to aB & S \to Ab & A \to aAB \\ B \to ABb & A \to a & B \to b \end{array}$$

é ambígüa.

(2) Considere a linguagem livre de contexto $L = \{0^n 1^{n+1} 0^4 : n \geq 0\}$.
  (a) Dê exemplo de uma gramática que gere L.
  (b) Dê exemplo de um autômato de pilha não determinístico que aceite L. Descreva o alfabeto de entrada e o alfabeto da pilha, o conjunto de estados, o estado inicial, os estados finais e a tabela de transição do autômato.

(3) Use o lema do bombeamento para provar que a linguagem $L = \{wcwcw : w \in \{0,1\}^*\}$ no alfabeto $\{0,1,c\}$ não é livre de contexto.

(4) Considere a linguagem $L = \{0^n 1^{2n} : n \geq 0\}$.
  (a) Construa o esquema de uma máquina de Turing que aceite L.
  (b) Mostre que esta linguagem é recursiva.

(5) Seja G uma gramática livre de contexto com conjunto de terminais T, conjunto de variáveis V e símbolo inicial S. Dizemos que G é *linear* se todas as suas regras são da forma $X \to \alpha Y \beta$, onde X e Y são variáveis e $\alpha, \beta \in T^*$. Isto é, pode haver no máximo uma variável do lado direito de cada regra de G. Uma linguagem é *linear* se pode ser gerada por alguma gramática linear.
  (a) Mostre que a união de linguagens lineares é linear.
  (b) Dê exemplo de uma linguagem linear que não é regular.

1^{o} semestre de 1999–Prova Final

(1) Seja $L = \{w \in \{0,1\}^* : w \neq \epsilon$ e w não contém a seqüência 11 como subpalavra$\}$.
   (a) Construa um autômato finito determinístico $\mathcal{A}$ tal que $L = L(\mathcal{A})$.
   (b) Determine uma gramática regular que gere L.

(2) Considere o autômato finito determinístico M no alfabeto $\{0,1\}$, com conjunto de estados $\{q_1, \ldots, q_4\}$, estado inicial q_1, estados finais $\{q_1\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_3	q_2
q_2	q_1	q_4
q_3	q_4	q_1
q_4	q_4	q_3

   (a) Esboce o grafo de M.
   (b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(3) Seja $L = \{wc^4w^r : w \in \{0,1\}\}$.
   (a) Mostre que L não é uma linguagem regular.
   (b) Descreva uma gramática livre de contexto G que gere L.
   (c) Descreva um autômato de pilha não determinístico que aceite L.

(4) Construa o diagrama de uma máquina de Turing que aceite $L = \{wcw : w \in \{0,1\}^*\}$.

(5) Seja L uma linguagem regular. Definimos o *posto* de L como sendo o *menor* inteiro positivo k para o qual existe um autômato finito determinístico A com k estados e tal que $L = L(A)$.
   (a) Se L_1 e L_2 são linguagens regulares cujos postos são k_1 e k_2 respectivamente, determine m (em termos de k_1 e k_2) de modo que o posto de $L_1 \cdot L_2$ seja menor ou igual a m.
   (b) Considere a afirmação: se $L_1 \subseteq L_2$ são linguagens regulares então o posto de L_1 tem que ser menor ou igual que o posto de L_2. Esta afirmação é verdadeira ou falsa? Justifique sua resposta com cuidado!

2^o semestre de 1999–Primeira Prova–Turma MAI

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, q_2, q_3\}$, estado inicial q_1, estados finais $\{q_1\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_1	q_3
q_2	q_1	q_3
q_3	q_2	q_3

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(2) Seja L a linguagem que consiste das palavras no alfabeto $\{0, 1\}$ cujo terceiro símbolo, contado a partir da *direita*, é 1.
(a) Construa uma expressão regular que denote L.
(b) Construa um autômato finito não determinístico que aceite L.
(c) Construa um autômato finito determinístico que aceite $\overline{L}$.
(d) Construa uma gramática linear à direita que gere L.

(3) Mostre que a linguagem $\{w1w : w \in \{0, 1\}^*\}$ não é regular, usando o lema do bombeamento.

(4) Seja L uma linguagem regular. O *posto* de L é o *menor* inteiro positivo k para o qual existe um autômato finito determinístico $\mathcal{M}$ com k estados e tal que $L = L(\mathcal{M})$.
(a) Determine um inteiro positivo m (em termos de k) de modo que o posto de L^* seja menor ou igual a m.
(b) Pode acontecer que L^* tenha posto menor ou igual a k?
Em cada caso justifique cuidadosamente sua resposta.

2^o semestre de 1999–Primeira Prova–Turma MAJ

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, q_2, q_3\}$, estado inicial q_1, estados finais $\{q_2\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_2	q_2
q_2	q_3	q_1
q_3	q_2	q_2

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(2) Considere as seguintes linguagens no alfabeto $\{0, 1\}$:
- L_1 é a linguagem denotada pela expressão regular 0^*101^*;
- L_2 é o conjunto das palavras nas quais toda vez que aparece um 0 ele está seguido de pelo menos um 1.

(a) Construa um autômato finito determinístico que aceite L_1, e um que aceite L_2.
(b) Construa um autômato finito não determinístico que aceite $L_1 \cup L_2$.
(c) Construa uma gramática linear à direita que gere $\overline{L}_1$.

(3) As linguagens desta questão estão definidas no alfabeto $\{0\}$.
(a) Seja $(a_0, a_1, \ldots)$ uma progressão aritmética de razão r. Construa uma expressão regular que denote a linguagem infinita $L_1 = \{0^{a_0}, 0^{a_1}, \ldots\}$.
(b) Mostre que $L_2 = \{0^{n!} : n \geq 1\}$ não é uma linguagem regular usando o lema do bombeamento.

(4) Seja M um autômato finito determinístico com apenas dois estados. Considere a afirmação: "se $\epsilon \notin L(M)$ então $L(M)$ tem que ser infinita".
(a) Esta afirmação é verdadeira ou falsa?
(b) E se ϵ pertencer a $L(M)$?
Justifique cuidadosamente suas respostas.

2^o semestre de 1999 –Segunda Prova–Turma MAI

(1) Construa uma máquina de Turing que *aceite* o complementar da linguagem $\{ww^R : w \in (0 \cup 1)^*\}$.

(2) Seja $L = \{a^n b^m c : n \geq m \geq 1\}$.
   (a) Dê exemplo de uma gramática livre de contexto que gera L.
   (b) Construa um autômato de pilha não determinístico cujo alfabeto da pilha tem apenas um símbolo e que aceita L.

(3) Mostre que $L = \{0^{n!} : n \geq 1\}$ não é uma linguagem livre de contexto, usando o lema do bombeamento.

(4) Sejam M_1 e M_2 autômatos de pilha não determinísticos com ingredientes $(\Sigma, \Gamma_1, q_1, F_1, \delta_1)$ e $(\Sigma, \Gamma_2, q_2, F_2, \delta_2)$, respectivamente.
   (a) Descreva um autômato de pilha não determinístico M, construído a partir dos ingredientes de M_1 e M_2 que aceita $L(M_1) \cup L(M_2)$.
   (b) É possível construir um autômato de pilha não determinístico com apenas dois estados que aceita $L(M_1) \cup L(M_2)$?

(5) Uma gramática livre de contexto com apenas uma variável tem que ser ambígüa? Justifique sua resposta.

2^{o} semestre de 1999 –Segunda Prova–Turma MAJ

(1) Construa uma máquina de Turing que decida a linguagem $\{0^{2n} : n \in \mathbb{N}\}$ no alfabeto $\{0, 1\}$.

(2) Seja $L = \{0^m 1^n : n \leq m \leq 2n\}$.
 (a) Dê exemplo de uma gramática livre de contexto que gera L.
 (b) Dê exemplo de um autômato de pilha não determinístico que aceita L.

(3) Mostre, usando o lema do bombeamento, que a linguagem $\{0^k 1^k 0^k : k \geq 0\}$ não é livre de contexto.

(4) Sejam Σ um alfabeto e seja $\phi : \Sigma \to \Sigma^*$ uma aplicação. Se $w = \sigma_1 \cdots \sigma_t \in \Sigma^*$, então

$$\phi(w) = \phi(\sigma_1) \cdots \phi(\sigma_t).$$

Assim, se L for uma linguagem no alfabeto Σ então $\phi(L) = \{\phi(w) : w \in L\}$ é uma outra linguagem em Σ.
 (a) Sejam $L = \{0^n 1^n : n \geq 0\}$ e $\phi : \{0, 1\} \to \{0, 1\}^*$ definida por $\phi(0) = 01^5$ e $\phi(1) = 1^4$. Calcule $\phi(L)$.
 (b) Mostre que se L é uma linguagem livre de contexto no alfabeto Σ e $\phi : \Sigma \to \Sigma^*$ então $\phi(L)$ também é livre de contexto.

Justifique cuidadosamente suas respostas.

(5) Uma linguagem regular pode ser inerentemente ambígüa? Justifique sua resposta.

2^{o} semestre de 1999 –Prova Final–Turma MAI

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_4\}$, estado inicial q_1, estados finais $\{q_3\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_1
q_4	q_4	q_4

  (a) Esboce o grafo de M.
  (b) Ache uma expressão regular para a linguagem L_1 aceita por M usando o algoritmo de substituição?
  (c) Esboce o grafo de um autômato finito não determinístico que aceite a linguagem L_2 denotada pela expressão regular $0(10^*1)^*10^*$.
  (d) Determine uma gramática linear à direita que gere L_1.

(2) Seja $L = \{a^{i+3}b^{2i+1} : i \geq 0\}$.
  (a) Mostre que L não é regular.
  (b) Determine uma gramática livre de contexto que gere L.
  (c) Determine um autômato de pilha não determinístico que aceite L.

(3) Construa uma máquina de Turing M que calcule a função $f : \mathbb{N} \to \mathbb{N}$ onde $f(n)$ é o resto da divisão de n por 3. A entrada da máquina deve ser $\rhd\sqcup 0^n$ e a saída deve ser $\rhd \sqcup 0^r$ onde r é o resto da divisão de n por 3.

(4) A união de uma infinidade de linguagens regulares tem que ser recursivamente enumerável? Justifique cuidadosamente sua resposta.

2^o semestre de 1999 –Prova Final–Turma MAJ

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_4\}$, estado inicial q_1, estados finais $\{q_3\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_2	q_4
q_2	q_3	q_1
q_3	q_3	q_4
q_4	q_4	q_4

(a) Esboce o grafo de M.
(b) Ache uma expressão regular para a linguagem L_1 aceita por M usando o algoritmo de substituição?
(c) Esboce o grafo de um autômato finito determinístico que aceite a linguagem L_2 denotada pela expressão regular $0^*0(1 \cup 100^*)$.
(d) Determine uma gramática linear à direita que gere L_1.

(2) Seja $L = \{a^i b^j c^j d^i e^3 : i, j \geq 0\}$.
(a) Mostre que L não é regular.
(b) Determine uma gramática livre de contexto que gere L.
(c) Determine um autômato de pilha não determinístico que aceite L.

(3) Construa uma máquina de Turing M que aceite o complementar da linguagem $\{0^i 1^i : i \geq 0\}$.

(4) O complemento de uma linguagem recursivamente enumerável que *não é recursiva* pode ser finito? Justifique cuidadosamente sua resposta.

Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

Não são aceitas respostas sem justificativa. Explique tudo o que você fizer.

Linguagens Formais—1º semestre de 2000 –Primeira Prova

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_5\}$, estado inicial q_1, estados finais $\{q_1, q_4\}$ e função de transição dada pela tabela abaixo:

δ_1	0	1
q_1	q_2	q_3
q_2	q_3	q_1
q_3	q_4	q_5
q_4	q_4	q_5
q_5	q_5	q_5

(a) Esboce o grafo de M.

(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(2) Considere a linguagem no alfabeto $\{0, 1\}$ denotada pela expressão regular $r = ((0^* \cdot 0) \cup 1^*)$.

(a) Construa, usando os métodos descritos na aula, um autômato finito não determinístico que aceite L. A construção deve começar a partir dos autômatos finitos que aceitam os símbolos do alfabeto. Ao final da construção esboce o grafo do autômato obtido eliminando os estados redundantes.

(b) Dê exemplo de uma expressão regular diferente de r mas que também denote L.

(3) Mostre, usando o lema do bombeamento, que a linguagem $\{ww : w \in \{0, 1\}^*\}$ não é regular, .

(4) Seja L uma linguagem no alfabeto $\{0, 1\}$ e defina L' como sendo o conjunto das palavras de L cujo comprimento é ímpar. Considere a afirmação:

se L for regular então L' também será regular.

Esta afirmação é verdadeira ou falsa? Prove a afirmação, se for verdadeira, ou dê um exemplo de uma linguagem regular L para a qual L' não é regular, se for falsa.

(5) Seja G a gramática linear à direita com conjunto de terminais $\{0, 1\}$, variáveis $\{S, X, Y\}$, símbolo inicial S e regras:

$$S \to 0^2X, \quad S \to 0Y, \quad X \to 1Y, \quad X \to 1S \quad \text{e} \quad Y \to 1.$$

Use os algoritmos descritos no curso para:

(a) Construir um autômato finito não determinístico M tal que $L(M)$ é a linguagem gerada por G.

(b) Construir, a partir de M, um autômato finito determinístico que aceita o *complementar* da linguagem gerada por G.

Linguagens Formais—1^o semestre de 2000 –Segunda Prova

(1) Dê exemplo de uma gramática livre de contexto G que gere a linguagem L no alfabeto $\{a, b, c\}$ dada por

$$L = \{wc^r : w \in (a \cup b)^* \text{ e } r = |w|\}.$$

Descreva todos os ingredientes de G.

(2) Construa um autômato de pilha não determinístico M tal que $L(M)$ seja o *complementar* da linguagem $\{ww^r : w \in (0 \cup 1)^*\}$. Descreva todos os ingredientes de M.

(3) Seja $w \in (0 \cup 1)^*$ uma palavra com r zeros e s uns. Construa uma máquina de Turing que, tendo como entrada a fita na forma $\triangleright \sqcup w$ rearranja os zeros e uns de modo que ao final da computação a fita está na forma $\triangleright \sqcup 0^r 1^s$.
 (a) Descreva em português o comportamento de M.
 (b) Esboce o diagrama desta máquina de Turing.

(4) Mostre, usando o lema do bombeamento, que a linguagem $L = \{0^{n^3} : n \geq 0\}$ não é livre de contexto.

(5) Seja L uma linguagem no alfabeto $\{0, 1\}$ e defina L' como sendo o conjunto das palavras de L cujo comprimento é ímpar. Considere a afirmação:
 se L for *recursiva* então L' também será *recursiva*.
Esta afirmação é verdadeira ou falsa? Prove a afirmação, se for verdadeira, ou dê um exemplo de uma linguagem recursiva L para a qual L' não é recursiva, se for falsa.

Linguagens Formais—1º semestre de 2000–Prova Final

(1) Considere o autômato finito determinístico M no alfabeto $\{0,1\}$, com conjunto de estados $\{q_1,\ldots,q_4\}$, estado inicial q_1, estados finais $\{q_4\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_4
q_4	q_4	q_4

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar a linguagem aceita por M.
(c) Ache uma gramática linear à direita que gera $L(M)$.
(d) Construa um autômato finito não determinístico M' tal que

$$L(M') = (L(M) \cup (001))^*.$$

(2) Considere a L a linguagem livre de contexto dada por:

$$L = \{wc^r : w \in (a \cup b)^* \text{ e } r > |w|\}.$$

(a) Mostre que L não é regular usando o lema do bombeamento.
(b) Construa uma gramática livre de contexto que gere L.
(c) Construa um autômato de pilha não determinístico que aceite L.
(d) Construa uma máquina de Turing que decida L.

(3) Seja L uma linguagem no alfabeto $\{0\}$, gerada por uma gramática livre de contexto G cujas regras são da forma $X \to \alpha Y \beta$ ou $X \to \alpha$, onde X e Y são variáveis e $\alpha, \beta \in 0^*$. Mostre que L é regular.

Linguagens Formais—2^o semestre de 2000 –Primeira Prova

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_5\}$, estado inicial q_1, estados finais $\{q_1, q_4\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_5
q_2	q_4	q_3
q_3	q_1	q_5
q_4	q_5	q_1
q_5	q_5	q_5

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(2) O cofre do banco *Imprudente S.A.* é aberto digitando uma senha em um teclado de três botões, marcados com as letras a, b e c. Por questões de segurança o banco decidiu que as senhas legítimas podem ter um número qualquer de símbolos, mas precisam satisfazer às seguintes restrições:
(a) toda senha começa por a;
(b) pelo menos dois dos três símbolos têm que aparecer ao menos uma vez na senha.

A chave eletrônica do cofre opera em duas etapas: primeiro o próprio dispositivo verifica se a senha digitada satisfaz os requisitos impostos pelo banco para toda senha legítima, se for este o caso, a senha é enviada por um canal seguro para ver se o usuário desta senha tem mesmo autorização para abrir o cofre.

Você foi contratado pelo banco para projetar um programa que implemente a primeira parte do reconhecimento de senhas; isto é, que reconheça se a senha digitada satisfaz os requisitos de uma senha legítima. Faça isso realizando cada uma das etapas seguintes:
(a) Determine a expressão regular r que denota a linguagem formada pelas senhas legítimas.
(b) Construa um autômato finito não determinístico M tal que $L(r)$ é a linguagem aceita por M.

(3) Mostre, usando o lema do bombeamento, que a linguagem

$$L = \{0^k 1^m : k, m \in \mathbb{N} \text{ e } \quad k + 2 \geq m\}$$

não é regular.

(4) Seja L uma linguagem no alfabeto $\{0, 1\}$ e defina $\text{pref}(L)$ como sendo o conjunto das palavras que são prefixos de alguma palavra de L. Isto é,

$$\text{pref}(L) = \{u \in (0 \cup 1)^* : \quad \text{existe} \quad v \in (0 \cup 1)^* \text{ tal que } \quad uv \in L\}.$$

Prove que se L é uma linguagem regular então $\text{pref}(L)$ também é uma linguagem regular.

(5) Seja G a gramática linear à direita com conjunto de terminais $\{0, 1\}$, variáveis $\{S, X, Y\}$, símbolo inicial S e regras:

$$S \to 01X, \quad S \to 0Y, \quad X \to Y, \quad X \to 1S \quad \text{e} \quad Y \to 1.$$

Use os algoritmos descritos no curso para:

(a) construir um autômato finito não determinístico M tal que $L(M)$ é a linguagem gerada por G.

(b) construir, a partir de M, um autômato finito determinístico que aceita a linguagem gerada por G.

Linguagens Formais— 2^{o} semestre de 2000–Segunda Prova

(1) Considere a linguagem
$$L = \{a^i b^j c^k : 0 < j \leq 2 \text{ e } i \geq 2k\}$$
no alfabeto $\{a, b, c\}$.
   (a) Dê exemplo de uma gramática livre de contexto G que gere a linguagem L e que seja *ambígüa*.
   (b) Construa duas árvores gramaticais distintas para a mesma palavra na sua gramática.
   (c) Dê exemplo de um autômato de pilha não determinístico que aceite L.

(2) Considere a linguagem L formada pelas palavras no alfabeto $\{a, b, c\}$ que têm uma quantidade de bs e cs que é múltipla de 3. Construa o diagrama de uma máquina de Turing que *decida* L.

(3) Mostre, usando o lema do bombeamento, que a linguagem $L = \{0^{3^n} : n \geq 0\}$ não é livre de contexto.

(4) Dê exemplos de linguagens L_1 e L_2 no alfabeto $\{0, 1\}$ tais que:
   (a) L_1 e L_2 são livres de contexto e $L_1 \cap L_2$ é regular.
   (b) L_1 e L_2 *não* são recursivas mas $L_1 \cup L_2$ é regular.

Linguagens Formais—2º semestre de 2000 –Prova Final

(1) Considere o autômato finito determinístico M no alfabeto $\{0,1\}$, com conjunto de estados $\{q_1,\ldots,q_5\}$, estado inicial q_1, estados finais $\{q_5\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_3
q_2	q_3	q_4
q_3	q_3	q_4
q_4	q_5	q_6
q_5	q_6	q_6
q_6	q_6	q_6

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.
(c) Dê exemplo de uma gramática linear à direita que gere L.
(d) Dê exemplo de um autômato finito não determinístico que aceite $L(M)^*11$.

(2) Considere a linguagem

$$L = \{wb^iw^R : w \in \{a,c\}^* \ \text{ e } \ i \geq 1\}$$

no alfabeto $\{a,b,c\}$. Lembre-se que w^R é a palavra obtida refletindo-se w.
(a) Prove, usando o lema do bombeamento, que L não é regular.
(b) Construa uma gramática livre de contexto que gere L.
(c) Construa uma autômato de pilha com apenas dois símbolos no alfabeto da pilha e que aceite L.
(d) Construa uma máquina de Turing que decida L.

(3) Prove que a união de uma linguagem regular com uma linguagem recursivamente enumerável é recursivamente enumerável.

Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

1^{o} semestre de 2001 –Primeira Prova–Turma MAJ

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_5\}$, estado inicial q_1, estados finais $\{q_2, q_4\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_5
q_2	q_5	q_3
q_3	q_1	q_4
q_4	q_2	q_5
q_5	q_5	q_5

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.
(c) Determine uma gramática linear à direita que gere $L(M)$.

(2) Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$((0 \cdot 1)^* \cup (1 \cdot 0)^*)^*.$$

(3) Considere as linguagens no alfabeto $\{0, 1\}$ definidas abaixo:

$L_1 = \{w : w$ tem comprimento pelo menos 3 e seu terceiro símbolo é zero $\}$

$L_2 = \{01^{5^n} : n \geq 0\}$.

(a) Determine uma expressão regular que denote L_1.
(b) Mostre, usando o lema do bombeamento, que L_2 não é regular.

(4) Sejam M e M' autômatos finitos *determinísticos* no alfabeto Σ cujos ingredientes são $(\Sigma, Q, q_1, F, \delta)$ e $(\Sigma, Q', q'_1, F', \delta')$, respectivamente. Defina um novo autômato finito *determinístico* N construído a partir de M e M' como segue:

Alfabeto: Σ;

Estados: pares (q, q') onde $q \in Q$ e $q' \in Q'$;
Estado inicial: (q_1, q'_1);
Estados finais: pares (q, q') onde $q \in F$ ou $q' \in F'$;
Transição: a função de transição é definida em um estado (q, q') de N e símbolo $\sigma \in \Sigma$ por

$$\hat{\delta}((q, q'), \sigma) = (p, p'),$$

onde $p = \delta(q, \sigma)$ e $p' = \delta'(q', \sigma)$.

Calcule $L(N)$ em função de $L(M)$ e $L(M')$ e prove que a sua resposta está correta.

1^{o} semestre de 2001 –Primeira Prova–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, \ldots, q_5\}$, estado inicial q_1, estados finais $\{q_2, q_4\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_2	q_3
q_4	q_5	q_4
q_5	q_5	q_4

(a) Esboce o grafo de M.
(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.
(c) Determine uma gramática linear à direita que gere $L(M)$.

(2) Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$((0 \cdot 0)^* \cdot 1) \cup (1 \cdot 0)).$$

Obedeça ao posicionamento dos parêntesis na expressão regular!

(3) Considere as linguagens no alfabeto $\{0, 1\}$ definidas abaixo:

$L_1 = \{w : \text{o comprimento de } w \text{ é um múltiplo de 3 ou 5 maior que 10}\}$

$L_2 = \{0^n 1 0^{n+1} : n \geq 0\}$.

(a) Determine uma expressão regular que denote L_1.
(b) Mostre, usando o lema do bombeamento, que L_2 não é regular.

(4) Seja L uma linguagem regular e seja M um autômato finito determinístico com o menor número possível de estados que aceita L. O *posto* de L é definido como sendo igual ao número de estados de M. Considere a linguagem $L_i = \{0^i\}$ no alfabeto $\{0\}$. Isto é, $L_1 = \{0\}$, $L_2 = \{0^2\}$, $L_5 = \{0^5\}$, e assim por diante. Determine o posto de L_i em função de i e prove que sua resposta está correta.

CUIDADO: se você chegar à conclusão de que L_i tem posto p não se esqueça de que é preciso mostrar que não existe nenhum autômato finito determinístico com menos de p estados que aceita L_i.

Linguagens Formais— 1^o semestre de 2001–Segunda Prova–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere a gramática livre de contexto com terminais $\{0, 1\}$, variáveis $\{S, X, Y\}$, símbolo inicial S e regras:

$$S \to XY, \quad X \to 0X0, \quad X \to \epsilon, \quad Y \to 1Y,$$

$$Y \to \epsilon, \quad Y \to W0 \quad \text{e} \quad W \to 0.$$

(a) Mostre que G é ambígüa.

(b) Mostre que $L(G)$ é regular.

(2) Dê exemplo de um autômato de pilha não determinístico que aceite a linguagem no alfabeto $\{a, b, c\}$ definida por

$$L = \{a^r b^s c^t : r \geq s + 2 \quad \text{e} \quad t \geq 2\}.$$

(3) Esboce o diagrama de uma máquina de Turing M que calcule o *quociente* da divisão por 2 de um número inteiro positivo n dado na fita. O inteiro n deve ser representado na fita em unário na forma $\triangleleft \sqcup 0^n$, e o quociente deve ser representado de forma semelhante ao final da computação.

(4) Mostre, usando o lema do bombeamento, que a linguagem $L = \{a^n b a^{2n} b a^{3n} : n \geq 0\}$ não é livre de contexto.

(5) Seja L uma linguagem no alfabeto Σ cujo complementar $\overline{L} = \Sigma^* \setminus L$ é livre de contexto. Prove que existe uma máquina de Turing que aceita L.

Linguagens Formais— 1º semestre de 2001–Segunda Prova–Turma MAJ

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere a linguagem

$$L = \{a^r b^s c^{r+2s} : r, s \geq 0\}$$

no alfabeto $\{a, b, c\}$.

(a) Dê exemplo de uma gramática livre de contexto G que gere a linguagem L e que seja *ambígüa*.

(b) Esboce o diagrama de uma máquina de Turing que decida a linguagem L.

(2) Dê exemplo de autômato de pilha não determinístico que aceite a linguagem

$$L = \{a^r b^s : r + s \quad \text{é par}\}.$$

(3) Mostre, usando o lema do bombeamento, que a linguagem $L = \{a^n b^n a^n b^n : n \geq 0\}$ não é livre de contexto.

(4) Seja A um autômato finito determinístico cujos ingredientes são $(\Sigma, Q, q_1, F, \delta)$. Descreva, detalhadamente, os ingredientes de uma máquina de Turing M que simula o comportamento de A. Em particular, M *aceita* exatamente as palavras que A aceita.

Sugestão: Cuidado com o que acontece nos estados finais de A, porque A pode continuar a computação depois de passar por um estado final mas M não pode. Assim, M deverá ter um estado final f criado especialmente.

Linguagens Formais— 1º semestre de 2001–Prova Final–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, q_2, q_3\}$, estado inicial q_1, estados finais $\{q_1\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_3
q_2	q_3	q_1
q_3	q_1	q_2

(a) Esboce o grafo de M.

(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(c) Construa, a partir de M, um autômato finito não determinístico que aceite a linguagem $L(M) \cdot (0^*11)$.

(2) Considere a linguagem L no alfabeto $\{0, 1\}$ formada pelas palavras com um número ímpar de símbolos e cujo símbolo do meio é 0.

(a) Dê exemplo de uma gramática livre de contexto G que gere a linguagem L.

(b) Construa, a partir de G, o autômato de pilha não determinístico que aceita L.

(c) Esboce o diagrama de uma máquina de Turing que decida a linguagem L.

(d) Prove que L não é uma linguagem regular usando o lema do bombeamento.

(3) Mostre que se L é uma linguagem regular, então existe um autômato finito *não* determinístico com *apenas um estado final* que aceita L

Linguagens Formais— 1^o semestre de 2001–Prova Final–Turma MAJ

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere o autômato finito determinístico M no alfabeto $\{0, 1\}$, com conjunto de estados $\{q_1, q_2, q_3, q_4\}$, estado inicial q_1, estados finais $\{q_2\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_2	q_4
q_2	q_3	q_1
q_3	q_3	q_1
q_4	q_4	q_4

(a) Esboce o grafo de M.

(b) Determine, pelo algoritmo de substituição uma expressão regular para $L(M)$.

(c) Construa, a partir de M, um autômato finito não determinístico que aceite a linguagem $L(M)^*$.

(2) Considere a linguagem $L = \{a^i b^j : i = j + 2 \quad \text{ou} \quad j = i + 2\}$ no alfabeto $\{a, b\}$.

(a) Dê exemplo de uma gramática livre de contexto G que gere a linguagem L.

(b) Construa, a partir de G, o autômato de pilha que aceita L.

(c) Esboce o diagrama de uma máquina de Turing que decida a linguagem L.

(d) Prove que L não é uma linguagem regular usando o lema do bombeamento.

(3) Dê exemplo de uma linguagem livre de contexto L, no alfabeto $\{0, 1\}$, que *não seja regular*, e cujo complemento $\overline{L} = (0 \cup 1)^* \setminus L$ também seja livre de contexto. Justifique cuidadosamente sua resposta.

Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

1^o semestre de 2003 –Primeira Prova–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Considere o autômato finito determinístico M no alfabeto $\{0,1\}$, com conjunto de estados $\{q_1,\ldots,q_3\}$, estado inicial q_1, estado final $\{q_3\}$ e função de transição dada pela tabela abaixo:

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_1

(a) Esboce o grafo de M.

(b) Use o algoritmo de substituição para achar uma expressão regular que denote a linguagem $L(M)$.

(c) Determine uma gramática linear à direita que gere $L(M)$.

(2) Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$((0\cup 1)\cdot(1\cdot 0)^*)^*.$$

Elimine os estados redundantes do autômato ao final da construção.

(3) Considere as linguagens definidas abaixo:

$L_1 = \{w \in (0\cup 1)^* : w$ tem comprimento par e seu terceiro símbolo é zero $\}$

$L_2 = \{w \in (0\cup 1)^* :$ o comprimento de w é um quadrado$\}$

(a) Determine uma expressão regular que denote L_1.

(b) Mostre, usando o lema do bombeamento, que L_2 não é regular.

(4) Considere um autômato finito não determinístico M com 2 estados, *nenhum dos quais é redundante*. Denote por M^d o autômato finito determinístico (sem estados redundantes) obtido a partir de M pela construção de subconjuntos.

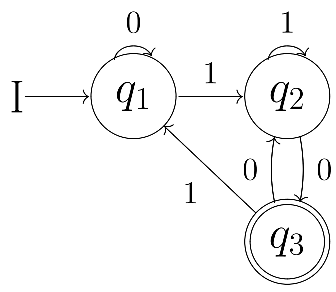
(a) Prove que M^d não pode ter apenas um estado.

(b) Dê um exemplo que ilustre porque (a) é falso se M tiver estados redundantes.

LEMBRETE: Um estado de um autômato finito é *redundante* se não é acessível a partir do estado inicial do autômato.

Solução

1(a) O grafo é:



1(b) As equações para as linguagens são:

$$L_1 = 0L_1 \cup 1L_2$$
$$L_2 = 0L_3 \cup 1L_2$$
$$L_3 = 0L_2 \cup 1L_1 \cup \epsilon.$$

Substituindo a última na segunda:

$$L_2 = 0(0L_2 \cup 1L_1 \cup \epsilon) \cup 1L_2 = (00 \cup 1)L_2 \cup 01L_1 \cup 0.$$

Aplicando o lema de Arden:

$$L_2 = (00 \cup 1)^*(01L_1 \cup 0).$$

Substituindo na primeira equação:

$$L_1 = 0L_1 \cup 1((00 \cup 1)^*(01L_1 \cup 0)) = (0 \cup 1((00 \cup 1)^*(01))L_1 \cup 1((00 \cup 1)^*0.$$

Aplicando o lema de Arden:

$$L_1 = (0 \cup 1((00 \cup 1)^*(01))^*1((00 \cup 1)^*0,$$

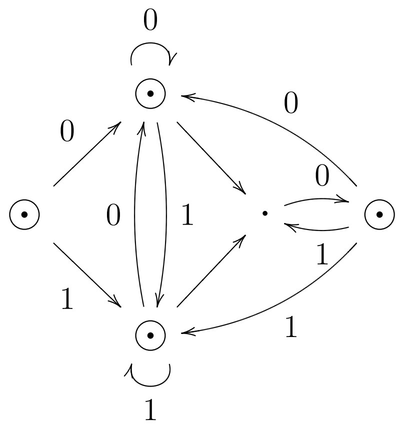
que é a linguagem aceita pelo autômato dado.

1(c) A gramática obtida diretamente a partir do autômato tem os seguintes ingredientes:

Terminais: $\{0, 1\}$

Variáveis: $\{q_1, q_2, q_3\}$
Símbolo inicial: q_1
Regras: $\{q_1 \to 1q_2, q_1 \to 0q_1, q_2 \to 1q_2, q_2 \to 0q_3, q_3 \to 0q_2, q_3 \to 1q_1, q_3 \to \epsilon\}$.

2. O grafo final, sem os estados redundantes, é o seguinte:



3a. Para estar em L_1 a palavra w tem ser da forma

$$\sigma_1\sigma_2 0\sigma_3 u,$$

onde $\sigma_1, \sigma_2, \sigma_3 \in \{0, 1\}$ e u é uma palavra de comprimento par em $\{0, 1\}$. Mas as palavras de comprimento par em $\{0, 1\}$ têm expressão regular

$$(00 \cup 01 \cup 10 \cup 11)^*.$$

Portanto, a expressão regular para L_1 é

$$(0 \cup 1)(0 \cup 1)0(0 \cup 1)(00 \cup 01 \cup 10 \cup 11)^*.$$

2(b). Suponhamos, por contradição, que L_2 seja aceita por um autômato finito determinístico com n estados. Vamos escolher $w = 0^{n^2} \in L_2$. Então $|w| = n^2 \geq n$. Aplicando o lema do bombeamento a w, obtemos que existe uma decomposição $w = xyz$ tal que

(1) $y \neq \epsilon$,
(2) $|xy| \leq n$,
(3) $xy^k z \in L_2$ para todo $k \geq 0$.

Escrevendo

$$x = 0^i, \quad y = 0^j \quad \text{e} \quad z = 0^{n^2-i-j},$$

temos que $j > 0$ e que

$$xy^k z = 0^i(0^j)^k 0^{n^2-i-j} = 0^{n^2+(k-1)j} \in L_2,$$

para todo $k \geq 0$. Portanto, $n^2 + (k-1)j$ tem que ser um quadrado para todo $k \geq 0$. Mas o conjunto

$$S = \{n^2 + (k-1)j : k \geq 0\}$$

é uma progressão aritmética, enquanto a seqüência formado pelos quadrados de inteiros não têm um passo fixo. Logo nem todos os elementos de S podem ser quadrados, o que contradiz o lema do bombeamento. Assim, podemos concluir que L_2 não é regular.

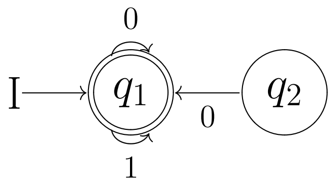
4(a). Denotaremos por δ a função de transição do AFND M e por δ_d a função de transição do AFD M^d.

Suponha que M tem estados q_1 e q_2 e que q_2 não é inacessível a partir de q_1. Isto significa que existe uma transição de q_1 para q_2. Digamos que $q_2 \in \delta(q_1, \sigma)$, onde σ é um símbolo do alfabeto do autômato M. Neste caso, M^d tem pelo menos dois estados. De fato,

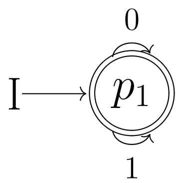
$$q_2 \in \delta(q_1, \sigma) = \delta_d(\{q_1\}, \sigma);$$

de modo que $\delta_d(\{q_1\}, \sigma) \neq \{q_1\}$. Assim, $\{q_1\}$ e $\delta_d(\{q_1\}, \sigma)$. correspondem a estados diferentes de M^d.

4(b). Se M tiver estados inacessíveis a partir de q_1, então o resultado é falso. Por exemplo, se M for



então M^d, sem os estados redundantes, é



onde $p_1 = \{q_1\}$.

Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

1^o semestre de 2003 –Segunda Prova–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

Considere as linguagens

$$L_1 = \{a^{3n}b^{5n}c^k : n, k \geq 0\}$$
$$L_2 = \{a^r b^m c^{2m} : m, r \geq 0\}$$

(1) Dê exemplo de uma gramática livre de contexto que gere L_1 e que não seja ambígüa.

(2) Prove que a gramática que você construiu em (1) não é ambígüa.

(3) Dê exemplo de um autômato de pilha não determinístico, *com apenas um símbolo na pilha*, que aceita L_2.

(4) Prove, usando o lema do bombeamento para linguagens livres de contexto, que $L = L_1 \cap L_2$ não é uma linguagem livre de contexto.

(5) Construa uma máquina de Turing M que decida a linguagem L_2.
   (a) Descreva o comportamento de M em palavras.
   (b) Esboce o diagrama de M.

(6) Sejam L_1, L_2 e L_3 linguagens recursivamente enumeráveis no alfabeto Σ. Suponha que
- $L_1 \cap L_2 = L_1 \cap L_3 = L_2 \cap L_3 = \emptyset$, e que
- $\Sigma^* = L_1 \cup L_2 \cup L_3$.

Mostre que L_1, L_2 e L_3 são recursivas.

GABARITO

(1) Dê exemplo de uma gramática livre de contexto que gere L_1 e que não seja ambígüa.

A gramática pode ser:

Terminais: $\{a, b, c\}$.
Variáveis: $\{S, X, Y\}$.
Símbolo inicial: S.
Regras: $\{S \to XY, X \to a^3Xb^5, X \to \epsilon, Y \to cY, Y \to \epsilon\}$.

(2) Prove que a gramática que você construiu em (1) não é ambígüa.

Basta provar que cada palavra de L_1 admite apenas uma derivação mais à esquerda nesta gramática. Seja $a^{3n}b^{5n}c^k$ a palavra de L_1 que queremos derivar na gramática da questão anterior.

Em primeiro lugar, qualquer derivação de uma palavra deve começar do símbolo inicial S. Como há apenas uma regra com S à esquerda da seta, então a derivação tem que começar com

$$S \Rightarrow XY.$$

Como a derivação é mais à esquerda, a regra seguinte vai incidir sobre X. Há duas regras com X à esquerda, mas $X \to \epsilon$ faz o X desaparecer. Contudo ainda precisamos pôr os as e bs. Mas isto só pode ser feito aplicando a regra $X \to a^3Xb^5$ n vezes, de modo que a derivação fica:

$$S \Rightarrow XY \Rightarrow a^3Xb^5Y \Rightarrow \cdots \Rightarrow a^{3n}Xb^{5n}Y.$$

Mas as regras sobre X só incidem sobre os as e bs. Como já temos todos os as e bs necessários em posição, precisamos fazer o X desaparecer usando $X \to \epsilon$. O que nos dá:

$$S \Rightarrow XY \Rightarrow a^3Xb^5Y \Rightarrow \cdots \Rightarrow a^{3n}Xb^{5n}Y \Rightarrow a^{3n}b^{5n}Y.$$

Agora, a variável mais á esquerda é o Y, que só pode ser usado para pôr os cs. Como há k cs, teremos a seguinte derivação ao final:

$$S \Rightarrow XY \Rightarrow a^3Xb^5Y \Rightarrow \cdots \Rightarrow a^{3n}Xb^{5n}Y \Rightarrow a^{3n}b^{5n}Y \Rightarrow a^{3n}b^{5n}cY \Rightarrow \cdots$$
$$\Rightarrow a^{3n}b^{5n}c^kY \Rightarrow a^{3n}b^{5n}c.$$

Como vimos, cada etapa desta derivação mais à esquerda fica completamente determinada pela palavra que está sendo derivada. Logo, cada palavra de L_1 admite apenas uma derivação mais à esquerda nesta gramática, o que mostra que a gramática não é ambígüa.

(3) Dê exemplo de um autômato de pilha não determinístico, *com apenas um símbolo na pilha*, que aceita L_2.

Podemos criar um autômato que começa por ignorar os as, já que não importa quantos as a palavra tem. Ao alcançar os bs, o autômato deve pôr na pilha dois cs para cada b que encontrar. Por fim o autômato desempilha um c para cada c que aparecer na entrada. Para assegurar que o autômato só aceite palavras em que os as, bs e cs aparecem na ordem correta, vou mudar de estado cada vez que o autômato passa a tratar de um símbolo diferente.

Este autômato tem os seguintes ingredientes:

Alfabeto de entrada: $\{a, b, c\}$.
Alfabeto da pilha: $\{c\}$.
Estados: $\{q_1, q_2, q_3\}$.
Estado inicial: q_1.
Estados finais: $\{q_3, q_4\}$.
Transições:

(4) Prove, usando o lema do bombeamento para linguagens livres de contexto, que $L = L_1 \cap L_2$ não é uma linguagem livre de contexto.

Temos que

$$L = L_1 \cap L_2 = \{a^{3n}b^{5n}c^{10n} : n \geq 0\}.$$

Estado	Entrada	Topo da Pilha	Transição
q_1	a	ϵ	(q_1, ϵ)
q_1	b	ϵ	(q_2, cc)
q_2	b	ϵ	(q_2, cc)
q_2	c	c	(q_3, ϵ)
q_3	c	c	(q_3, ϵ)
q_1	ϵ	ϵ	(q_4, ϵ)

Suponhamos, por contradição, que L seja livre de contexto. Então, pelo lema do bombeamento, que exista um inteiro $\rho > 0$ tal que, como

$$a^{3\rho}b^{5\rho}c^{10\rho}$$

tem comprimento $18\rho > \rho$, então esta palavra admite uma decomposição na forma

$$a^{3\rho}b^{5\rho}c^{10\rho} = uvxyz,$$

onde:

(1) $vy \neq \epsilon$;
(2) $|vxy| \leq \rho$;
(3) $uv^kxy^kz \in L$ para todo $k \geq 0$.

Como $|vxy| \leq \rho$, mas a quantidade de as e bs excede ρ, então se vxy tiver um a, não pode ter um c. Analogamente, se vxy tiver um c, não pode conter um a. Mas isto significa que ao bombear a palavra, a quantidade de apenas dois dos três símbolos que constituem a palavra é alterada. Assim, a proporção entre os três símbolos muda e a palavra não pode mais pertencer a L.

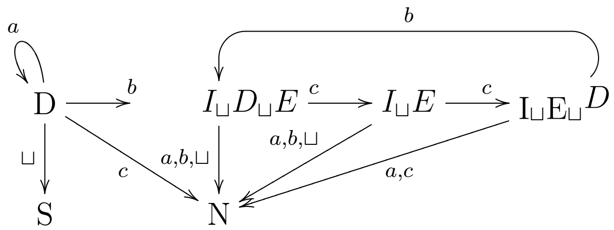
(5) *Construa uma máquina de Turing M que decida a linguagem L_2.*

(1) *Descreva o comportamento de M em palavras.*
(2) *Esboce o diagrama de M.*

Lembre-se que a entrada deve ser da forma $\triangleright\sqcup w$, onde $w \in (a \cup b \cup c)^*$. A máquina deve ignorar os as, e comparar o número de bs com o de cs. Para cada b que ela achar, deve esperar encontrar dois cs. Se isto não ocorrer a palavra é rejeitada. Para implementar isto, faremos a máquina andar para a direita sobre os as ignorando-os.

Ao encontrar um b, a máquina o apaga, move-se para a extrema esquerda e, andando para a esquerda, procura e (se encontrá-los), apaga dois cs. Depois o processo é reiniciado movendo-se o cabeçote para o primeiro vazio à direita da última casa considerada. Ao final, a máquina espera encontrar a fita vazia. Qualquer coisa que sair errado, a máquina rejeita a entrada.

O diagrama desta máquina é o seguinte:



(6) *Sejam L_1, L_2 e L_3 linguagens recursivamente enumeráveis no alfabeto Σ. Suponha que*

- $L_1 \cap L_2 = L_1 \cap L_3 = L_2 \cap L_3 = \emptyset$, *e que*
- $\Sigma^* = L_1 \cup L_2 \cup L_3$.

Mostre que L_1, L_2 e L_3 são recursivas.

Escolha um inteiro k entre 1 e 3. Como $L_k \cap L_j = \emptyset$, sempre que $j \neq k$, temos que $L_k \cap \mathcal{L} = \emptyset$, onde $\mathcal{L}$ é a união das linguagens L_i com $i \neq k$. Como $\Sigma^* = L_k \cup \mathcal{L}$, concluímos que

$$\Sigma^* \setminus L_k = \mathcal{L}.$$

Mas a união de linguagens recursivamente enumeráveis é uma linguagem recursivamente enumerável; de modo que $\mathcal{L}$ é recursivamente enumerável. Portanto, L_k e $\Sigma^* \setminus L_k$ são recursivamenente enumervéis. Mas isto implica (exercício 15 da lista) que L_k e $\Sigma^* \setminus L_k$ são recursivas. Como isto vale para qualquer $1 \leq k \leq 3$, concluímos que $L_1, \ldots, L_3$ são recursivas.

Universidade Federal do Rio de Janeiro

Departamento de Ciência da Computação

1^o semestre de 2003 –Prova Final–Turma MAI

Não serão aceitas respostas sem justificativa. Explique tudo o que você fizer.

(1) Seja M o autômato finito determinítico no alfabeto $\{0, 1\}$, com estados $\{q_1, q_2, q_3\}$, estado inicial q_1, estados finais $\{q_1, q_2\}$ e função de transição dada por

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_1
q_3	q_2	q_3

(a) Determine, pelo algoritmo de substituição, uma expressão regular para a linguagem aceita por M.

(b) Determine uma gramática linear à direita que gere M.

(2) Construa, passo-a-passo, um autômato finito não determinístico que aceite a linguagem denotada por $(00^* \cup 1)^*$.

(3) Considere a linguagem no alfabeto $\{a, b, c\}$ dada por

$$L = \{a^r b^s c^t : r = s + 2t\}$$

(a) Dê exemplo de uma gramática livre de contexto que gere L.

(b) Dê exemplo de um autômato de pilha não determinístico que aceite L.

(c) Prove que L não é regular usando o lema do bombeamento.

(4) Seja L a linguagem denotada pela expressão regular aba^*b no alfabeto $\{a, b\}$. Construa uma máquina de Turing que ao receber $\triangleright\underline{\sqcup}w$, onde $w \in (a \cup b)^*$ pára com $\triangleright\underline{\sqcup}0$ na fita se $w \notin L$ e pára com $\triangleright\underline{\sqcup}1$ na fita se $w \in L$.

(5) Determine se cada uma das afirmações abaixo é verdadeira ou falsa. *Justifique cuidadosamente suas respostas.*

(a) Se L e L' são linguagens recursivamente enumeráveis em um mesmo alfabeto então $\overline{L} \cap L'$ também é recursivamente enumerável, onde $\overline{L}$ é o complementar de L.

(b) Uma linguagem regular não pode ser inerentemente ambígüa.

Gabarito

(1a) As equações são:

$$L_1 = 0L_1 \cup 1L_2 \cup \epsilon$$
$$L_2 = 0L_3 \cup 1L_1 \cup \epsilon$$
$$L_3 = 0L_2 \cup 1L_3.$$

Pelo lema de Arden, segue da terceira equação que $L_3 = 1^*0L_2$, e substituindo na segunda equação, obtemos

$$L_2 = 01^*0L_2 \cup 1L_1 \cup \epsilon.$$

Aplicando o lema de Arden novamente

$$L_2 = (01^*0)^*(1L_1 \cup \epsilon).$$

De modo que a primeira equação se torna

$$L_1 = 0L_1 \cup 1(01^*0)^*(1L_1 \cup \epsilon) \cup \epsilon = (0 \cup 1(01^*0)^*1)L_1 \cup 1(01^*0)^* \cup \epsilon.$$

Aplicando o lema de Arden uma última vez:

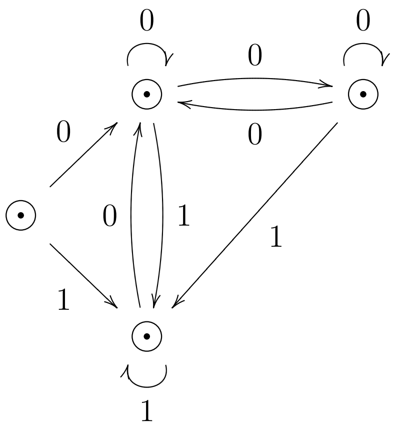
$$L_1 = (0 \cup 1(01^*0)^*1)^*(1(01^*0)^* \cup \epsilon),$$

que é a linguagem aceita pelo autômato dado.

(1b) A gramática é dada por

Terminais: $\{0, 1\}$.
Variáveis: $\{q_1, q_2, q_3\}$.
Símbolo inicial: q_1.
Regras: $\{q_1 \to 0q_1, q_1 \to 1q_2, q_1 \to \epsilon, q_2 \to 0q_3, q_2 \to 1q_1, q_2 \to \epsilon, q_3 \to 0q_2, q_3 \to 1q_3\}$.

(2) O grafo, com os estados redundantes já removidos, é o seguinte:



(2a) A gramática pode ser:

Terminais: $\{a, b, c\}$.
Variáveis: $\{S, X\}$.
Símbolo inicial: S.
Regras: $\{S \to a^2Sc, S \to X, X \to aXb, X \to \epsilon\}$.

(2b) O autômato pode ser construído a partir da gramática que já foi obtida no item anterior. Temos, então:

Alfabeto de entrada: $\{a, b, c\}$.
Alfabeto da pilha: $\{a, b, c, S, X\}$.
Estados: $\{i, f\}$.
Estado inicial: i.
Estados finais: $\{f\}$.
Transições:

(2c) Suponha, por contradição, que L seja uma linguagem aceita por um autômato finito determinístico com n estados. Seja

$$w = a^{3n}b^nc^n.$$

Estado	Entrada	Topo da Pilha	Transição
i	ϵ	ϵ	(f, S)
f	ϵ	S	(f, a^2Sc)
f	ϵ	S	(f, X)
f	ϵ	X	(f, aXb)
f	ϵ	X	(f, ϵ)
f	a	a	(f, ϵ)
f	b	b	(f, ϵ)
f	c	c	(f, ϵ)

Como $|w| = 4n \geq n$ então, pelo lema do bombeamento, existe uma decomposição $w = xyz$, onde

- $y \neq \epsilon$;
- $|xy| \neq n$;
- $xy^k z \in L$, para todo $k \geq 0$.

De (2) temos que x e y só contêm as. Portanto,

$$x = a^i,\ y = b^j \ \text{ e } \ z = a^{3n-i-j} b^n c^n.$$

Bombeando y, obtemos que

$$a^i (a^j)^k a^{3n-i-j} b^n c^n = a^{3n+(k-1)j} b^n c^n$$

pertence a L qualquer que seja $k \geq 0$. Mas isto implica que

$$2n + n = 3n + (k-1)j,$$

ou seja, que $(k-1)j = 0$ para todo $k \geq 0$; o que contradiz (1). Portanto, L não é uma linguagem regular.

(4) O diagrama da máquina de Turing é o seguinte:

$$\begin{tikzcd}
D \arrow[r, "a"] \arrow[dd, "b{,}\sqcup"] & I_\sqcup D \arrow[r, "b"] \arrow[ddl, "a{,}\sqcup"'] & I_\sqcup D \arrow[r, bend left, "a"] \arrow[d, "b"] \arrow[ddll, "\sqcup"] & I_\sqcup \arrow[l, bend left] \\
 & & I_\sqcup D \arrow[r, "\sqcup"] & E_\triangleright DDI_1 \\
D_\sqcup E \arrow[r, bend left, "\neq\triangleright"] \arrow[dr, "\triangleright"'] & I_\sqcup \arrow[l, bend left] & & \\
 & DDI_0 & &
\end{tikzcd}$$

(5a) Falso. Seja $L = \mathcal{L}_0$, a linguagem que construímos e que é recursivamente enumerável mas não é recursiva. O complementar desta linguagem não é recursivamente enumerável. Portanto, $\overline{L}$ não será recursivamente enumerável. Seja agora, $L' = \Sigma^*$, onde Σ é o alfabeto da linguagem $L = \mathcal{L}_0$. Então

$$\overline{L} \cap L' = \overline{L} \cap \Sigma^* = \overline{L},$$

que não é recursivamente enumerável.

(5b) Verdadeiro. Suponha que L seja regular. Então L é aceita por um autômato finito determinístico M. Seja G a gramática linear à direita construída a partir de M. Como M é determinístico, só existe uma computação em M para cada palavra de L. Como computações em M correspondem passo-a-passo a derivações em G, então cada palavra de L admite uma única derivação em G. Em particular, cada palavra de L tem uma única derivação mais à esquerda em G e, portanto, uma única árvore de derivação. Logo G não pode ser ambígüa. Mas se L pode ser gerada por uma gramática não ambígüa então L não pode ser inerentemente ambígüa.

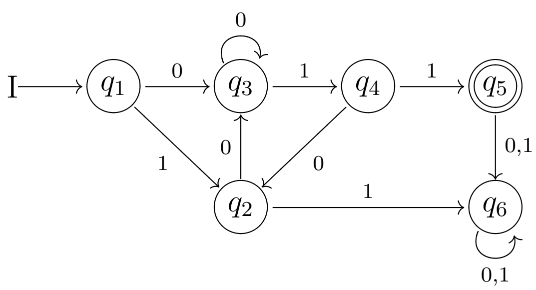
Justifique cuidadosamente as suas respostas.

Teste 1

1. Seja L a linguagem no alfabeto $\{0, 1\}$, formada pelas palavras que acabam em 00 e cuja quantidade de zeros é um múltiplo de 3.

(1) Dê uma expressão regular que denote a linguagem L.
(2) Construa um autômato finito determinístico que aceite L.

2. Use o algoritmo de substituição para achar a expressão regular da linguagem aceita pelo autômato finito determinístico cujo grafo é dado abaixo:

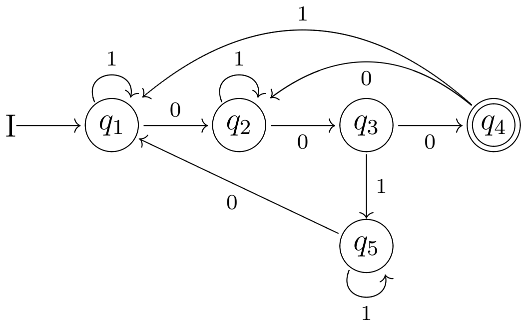


Resolução

1. Uma expressão regular que denota a linguagem desejada é

$$(01^*01^*01^*)^*01^*00.$$

Já um autômato finito determinístico que aceita esta linguagem é dado por



2. Aplicando o algoritmo de substituição, obtemos o seguinte sistema de linguagens

$$
\begin{aligned}
L_1 &= 0L_3 \cup 1L_2 \\
L_2 &= 0L_3 \cup 1L_6 \\
L_3 &= 0L_3 \cup 1L_4 \\
L_4 &= 0L_2 \cup 1L_5 \\
L_5 &= (0 \cup 1)L_6 \cup \epsilon \\
L_6 &= (0 \cup 1)L_6.
\end{aligned}
$$

Como q_6 é um estado morto, temos que $L_6 = \emptyset$, de modo que o sistema fica

$$
\begin{aligned}
L_1 &= 0L_3 \cup 1L_2 \\
L_2 &= 0L_3 \\
L_3 &= 0L_3 \cup 1L_4 \\
L_4 &= 0L_2 \cup 1L_5 \\
L_5 &= \epsilon
\end{aligned}
$$

Portanto,

$$L_4 = 0L_2 \cup 1\epsilon = 0L_2 \cup 1.$$

Substituindo na equação para L_3,

$$L_3 = 0L_3 \cup 1(0L_2 \cup 1) = 0L_3 \cup 10L_2 \cup 11.$$

Assim, pelo lema de Arden

$$L_3 = 0^*(10L_2 \cup 11) = 0^*10L_2 \cup 0^*11.$$

Substituindo isto nas equações correspondentes a L_1 e L_2, obtemos

$$L_1 = 0(0^*10L_2 \cup 0^*11) \cup 1L_2 = (00^*10 \cup 1)L_2 \cup 00^*11$$

$$L_2 = 0(0^*10L_2 \cup 0^*11) = 00^*10L_2 \cup 00^*11$$

Aplicando o lema de Arden à equação que define L_2,

$$L_2 = (00^*10)^*(00^*11),$$

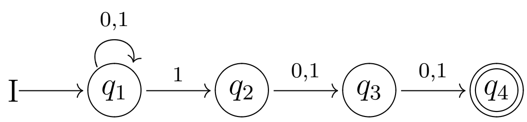
de modo que

$$L_1 = (00^*10 \cup 1)((00^*10)^*(00^*11)) \cup 00^*11,$$

é a linguagem aceita pelo autômato dado.

Teste 2

1. Use a construção de subconjuntos para achar um autômato finito determinístico que aceita exatamente a mesma linguagem que o autômato finito não determinístico cujo grafo é



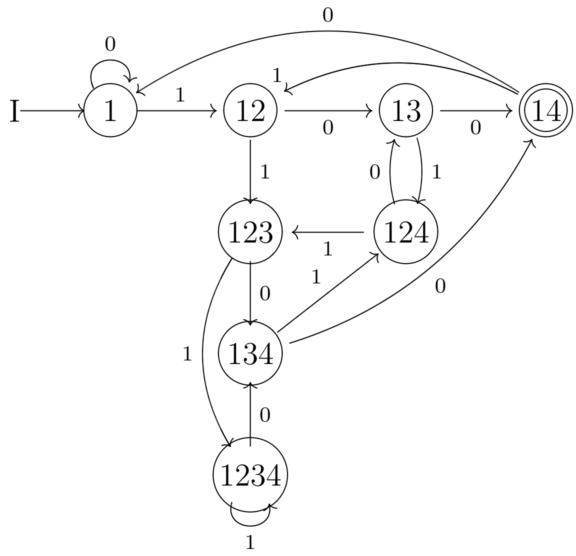
2. Dada uma palavra $v \in \{0,1\}^*$, denote por $\overline{v}$ a palavra obtida trocando cada 0 de w por um 1, e vice-versa. Por exemplo, $\overline{101} = 010$. Mostre, usando o lema do bombeamento, que a linguagem

$$L = \{v\overline{v} : v \in \{0,1\}^*\},$$

não pode ser aceita por nenhum autômato finito determinístico.

Resolução

1. Para simplificar a notação vamos escrever ij para denotar o estado $\{q_i, q_j\}$, e assim por diante.



2. Suponhamos, por contradição, que L seja aceita por um autômato finito determinístico com n estados. Vamos escolher $v = 0^n$, de modo que

$$w = v\overline{v} = 0^n 1^n \in L.$$

Então $|w| = 2n \geq n$. Aplicando o lema do bombeamento a w, obtemos que existe uma decomposição $w = xyz$ tal que

(1) $y \neq \epsilon$,
(2) $|xy| \leq n$,
(3) $xy^k z \in L$ para todo $k \geq 0$.

Escrevendo

$$x = 0^i, \quad y = 0^j \quad \text{e} \quad z = 0^{n-i-j}1^n,$$

temos que $j > 0$ e que

$$xy^k z = 0^i(0^j)^k 0^{n-i-j}1^n = 0^{n+(k-1)j}1^n \in L,$$

para todo $k \geq 0$. Mas, para que isto seja verdade, é preciso que

$$0^{n+(k-1)j}1^n = u\overline{u},$$

para alguma palavra $u \in (0 \cup 1)^*$. Entretanto, se $k > 1$, então a metade inicial dos símbolos de $0^{n+(k-1)j}1^n$ será formada por zeros. Portanto, u só vai conter zeros, de modo que $\overline{u}$ só vai conter 1s. Isto não é possível porque $n + (k-1)j > n$ quando $k > 1$, de modo que sempre vão sobrar alguns zeros na segunda metade da palavra. Temos, assim, uma contradição, e podemos concluir que L não é regular.

Teste 3

1. Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$0^* \cdot (01 \cup 00)^*.$$

Elimine os estados redundantes do autômato à medida que for efetuando a construção.

2. Dê exemplo de uma gramática livre de contexto que gere a linguagem no alfabeto $\{a, b\}$ definida por

$$L = a^*b^* \setminus \{a^n b^n : n \geq 0\}.$$

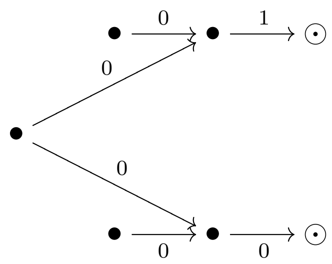
A gramática que você obteve é regular à direita? Explique sua resposta.

Resolução

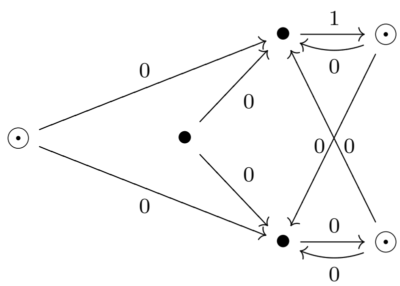
1. Os estados redundantes serão removidos de uma etapa para a outra. Os estados que eram finais nos autômatos da etapa $k-1$ e deixaram de ser finais na etapa k aparecem marcados com $\oslash$ nas figuras.

Na primeira etapa construo 01 e 00:

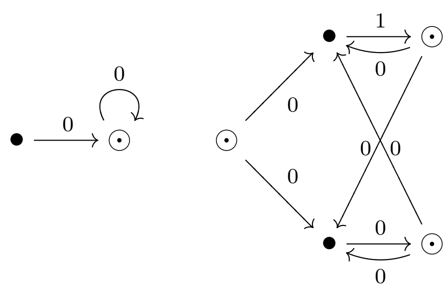
Na segunda etapa construo $01 \cup 00$:



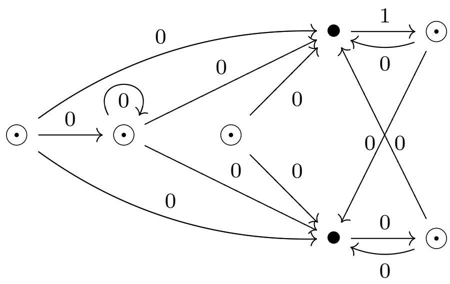
Na terceira etapa construo $(01 \cup 00)^*$:



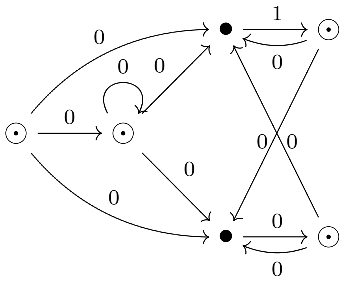
Na quarta etapa construo 0^*:



Na quinta etapa construo $0^* \cdot (01 \cup 00)^*$:



Removendo os estados que redundantes que ainda sobraram, obtemos:



2. Uma gramática possível é a seguinte gramática $\mathcal{G}$:

   Terminais: $\{a, b\}$
   Variáveis: $\{S\}$
   Símbolo inicial: S
   Regras: $\{S \to aSb, S \to aX, S \to Yb, X \to aX, X \to \epsilon, Y \to Yb, Y \to \epsilon\}$.

Uma gramática regular à direita só tem regras do tipo

$$X \to \sigma Y \ \text{ e } \ X \to \epsilon,$$

onde X e Y são variáveis e σ é um terminal. Portanto, $\mathcal{G}$ não é regular à direita já que regras como $S \to aSb$ não são de nenhum dos dois tipos acima.

Teste 4

1. Considere a gramática $\mathcal{G}$ determinada por

Terminais: $\{0, 1\}$.
Variáveis: $\{S\}$
Símbolo inicial: S
Regras: $\{S \to S0^2, S \to 1S, S \to 1\}$

(1) Mostre que $\mathcal{G}$ é ambígua, descrevendo duas árvores gramaticais distintas cuja colheita é igual.
(2) Modifique as regras de $\mathcal{G}$ de modo a obter uma gramática *não* ambígua $\mathcal{G}'$ que gera a mesma linguagem que $\mathcal{G}$. Justifique cuidadosamente as suas respostas

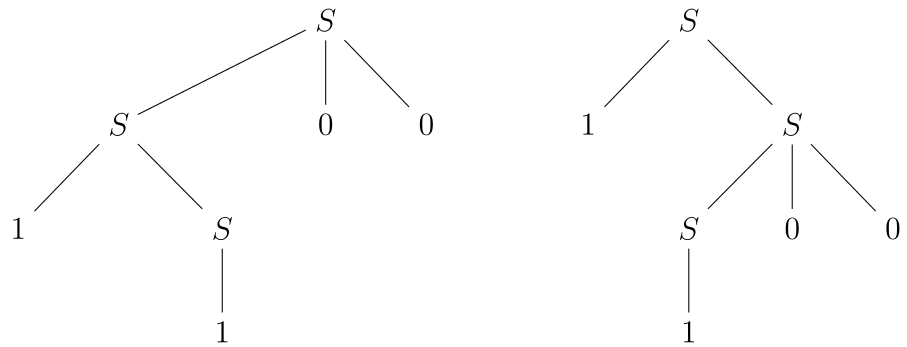
2. Dada uma palavra $s \in \{a, b\}^*$, denote por $\overline{s}$ a palavra obtida trocando cada a de w por um b, e vice-versa. Por exemplo, $\overline{bab} = aba$. Prove, usando o lema do bombeamento para linguagens livres de contexto que a linguagem no alfabeto $\{a, b, c\}$ definida por

$$L = \{sc\overline{s} : s \in \{a, b\}^*\},$$

não é livre de contexto.

Resolução

1.(1) As duas árvores de derivação seguintes são diferentes mas têm ambas colheita 1100:



(2) Para alterar a gramática de modo que deixe de ser ambígua basta criar uma ordem fixa na qual os 0s e 1s são postos em uma derivação. por exemplo, crie uma variável extra X e tome $\mathcal{G}'$ como sendo a gramática:

Terminais: $\{0, 1\}$.
Variáveis: $\{S, X\}$
Símbolo inicial: S
Regras: $\{S \to S0^2, S \to X, X \to 1X, X \to 1\}$

Com esta alteração, precisamos sempre começar pondo os 0s usando $S \to S0^2$. Depois usamos $S \to X$ para criar um X, e aí aplicamos $X \to 1X$ e $X \to 1$ para criar os 1. Portanto, cada palavra tem apenas uma derivação nesta gramática. Em particular, cada palavra tem apenas uma derivação mais à esquerda. Mas isto implica que cada palavra é colheita de apenas uma árvore; isto é, $\mathcal{G}'$ não pode ser ambígua.

2. Suponhamos, por contradição, que L seja livre de contexto. Então, pelo lema do bombeamento, que exista um inteiro $\rho > 0$ tal que, se $s = a^\rho b^\rho$ e

$$w = sc\overline{s} = a^\rho b^\rho c b^\rho a^\rho,$$

então, como $|w| = 4\rho > \rho$, temos que w admite uma decomposição na forma

$$a^\rho b^\rho c b^\rho a^\rho = uvxyz,$$

onde:

(1) $vy \neq \epsilon$;

(2) $|vxy| \leq \rho$;
(3) $uv^kxy^kz \in L$ para todo $k \geq 0$.

Há vários casos a considerar.

Caso 1: vxy pertence a s, isto é está todo antes do c.

Neste caso, bombeando criamos mais símbolos antes do que depois de c, o que não pode acontecer. Logo, temos uma contradição neste caso.

Caso 2: vxy pertence a $\overline{s}$, isto é está todo depois do c.

Neste caso, bombeando criamos mais símbolos depois do que antes de c, o que não pode acontecer. Logo, temos uma contradição também neste caso.

Caso 3: vxy inclui símbolos antes do c e depois do c.

Como $|vxy| \leq \rho$, segue que, ao bombear, os únicos símbolos que podem aumentar de quantidade antes ou depois do c são os bs. Mas se o número de bs antes ou depois do c aumentar, então o número de as também terá que aumentar para a palavra continuar em L. Como isto não pode acontecer, temos uma contradição. Finalmente, apenas o número de cs poderia aumentar, mas neste caso teríamos uma contradição imediata porque todas as palavras de L contêm apenas um c.

TESTE 5

1. Considere a gramática $\mathcal{G}$ determinada por

Terminais: $\{0, 1\}$.
Variáveis: $\{S\}$
Símbolo inicial: S
Regras: $\{S \to 1S0, S \to 1S1, S \to 0S0, S \to 0S1, S \to 0\}$

(1) Construa, a partir de $\mathcal{G}$, um autômato de pilha que aceite $L(\mathcal{G})$.
(2) Construa um autômato de pilha que aceite $L(\mathcal{G})$ e *que tenha apenas um símbolo na pilha.*

2. Construa o diagrama de uma máquina de Turing que *decida* a linguagem no alfabeto $\{a, b, c\}$ definida por $L = \{a^n cb^n : n \geq 0\}$.

3. Sejam L e L' linguagens em um alfabeto Σ. Considere a afirmação:

> Se L é recursiva e L' é recursivamente enumerável então $L \cap L'$ tem que ser recursiva.

Esta afirmação é verdadeira ou falsa? Prove a afirmação, se for verdadeira, ou dê um contra-exemplo, se for falsa.

Resolução

1.(1) O autômato resultante da gramática é o seguinte:

Alfabeto de entrada: $\{0, 1\}$.
Alfabeto da pilha: $\{0, 1, S\}$.
Estados: $\{i, f\}$.
Estado inicial: i
Estados finais: $\{f\}$
Transição:

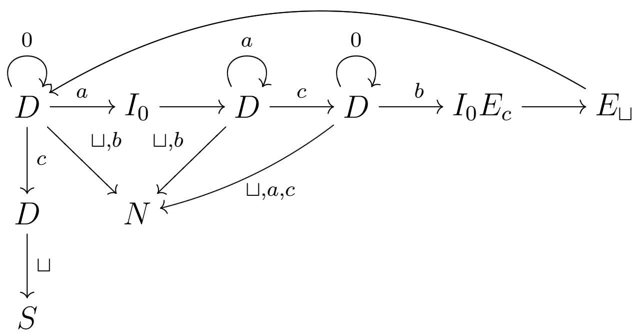
Estado	Entrada	Topo da pilha	Transição
i	ϵ	ϵ	(f, ϵ)
f	ϵ	S	$(f, 1S0), (f, 1S1), (f, 0S1), (f, 0S0)$
f	0	0	(f, ϵ)
f	1	1	(f, ϵ)

1.(2) Um autômato possível com apenas um símbolo na pilha é o seguinte:

Alfabeto de entrada: $\{0, 1\}$.
Alfabeto da pilha: $\{a\}$.
Estados: $\{q_1, q_2\}$.
Estado inicial: q_1
Estados finais: $\{q_2\}$
Transição:

Estado	Entrada	Topo da pilha	Transição
q_1	0	ϵ	(q_1, a)
q_1	1	ϵ	(q_1, a)
q_1	0	ϵ	(q_2, ϵ)
q_2	0	a	(q_2, ϵ)
q_2	1	a	(q_2, ϵ)

2. O gráfico da máquina está esboçado abaixo:



3. A afirmação é falsa. Basta tomar L como sendo Σ^* e L' como sendo uma linguagem recursivamente enumerável que não é recursiva. Então, L é regular e, portanto,

recursiva, mas

$$L \cap L' = \Sigma^* \cap L' = L',$$

não é recursiva por construção.

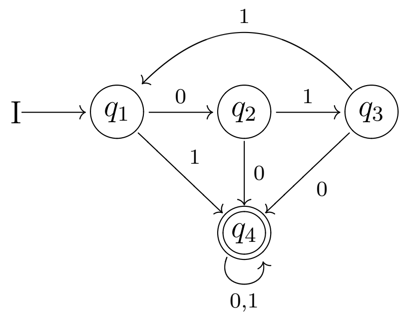
Prova Final

1. Seja L a linguagem no alfabeto $\{0, 1\}$, formada pelas palavras que começam ou acabam em 00 (ou ambos).

(1) Dê uma expressão regular que denote a linguagem L.

(2) Construa um autômato finito determinístico que aceite L.

2. Use o algoritmo de substituição para achar a expressão regular da linguagem aceita pelo autômato finito determinístico cujo grafo é dado abaixo:



3. Mostre, usando o lema do bombeamento, que a linguagem

$$L = \{a^i b^j : i = j \text{ ou } \ i = 2j\},$$

não pode ser aceita por nenhum autômato finito determinístico.

4. Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$0^* \cup ((01)^* \cdot 00)^*.$$

Elimine os estados redundantes do autômato à medida que for efetuando a construção.

5. Seja L a linguagem no alfabeto $\{a, b\}$ definida na questão 3.

(1) Descreva uma gramática livre de contexto que gere L.

(2) Descreva um autômato de pilha não determinístico que aceite L.

6. Construa o diagrama de uma máquina de Turing que *decida* a linguagem no alfabeto $\{a, b\}$ definida na questão 3.

7. Sejam L e L' linguagens recursivas em um alfabeto Σ. Prove que

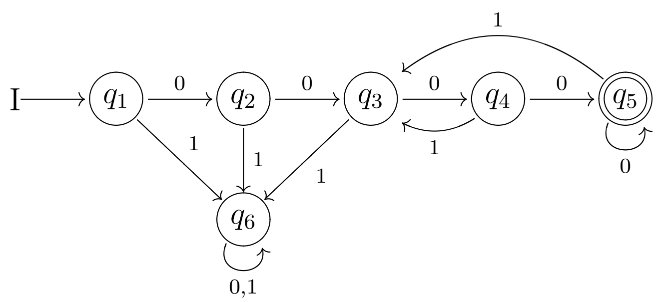
$$L \setminus L' = \{w \in L : w \notin L'\}$$

também é uma linguagem recursiva.

Resolução

1. (1) $(00)(o \cup 1)^* 00$

1. (2)



2. As equações são

$$L_1 = 0L_2 \cup 1L_4$$
$$L_2 = 0L_4 \cup 1L_3$$
$$L_3 = 0L_4 \cup 1L_1$$
$$L_4 = 0L_4 \cup 1L_4 \cup \epsilon$$

Por Arden,

$$L_4 = (0 \cup 1)^*.$$

Substituindo na penúltima equação

$$L_3 = 0(0 \cup 1)^* \cup 1L_1.$$

Substituindo L_4 e L_3 na segunda equação

$$L_2 = 0(0 \cup 1)^* \cup 1(0(0 \cup 1)^* \cup 1L_1),$$

donde

$$L_2 = (0(0 \cup 1)^* \cup 10(0 \cup 1)^*) \cup 11L_1.$$

Na primeira,

$$L_1 = 0L_2 \cup 1L_4 = 0((0(0 \cup 1)^* \cup 10(0 \cup 1)^*) \cup 11L_1) \cup 1(0 \cup 1)^*.$$

Isto é,

$$L_1 = 0L_2 \cup 1L_4 = 0((0(0 \cup 1)^* \cup 10(0 \cup 1)^*) \cup 1(0 \cup 1)^*) \cup 011L_1.$$

Finalmente, aplicando Arden

$$L_1 = (011)^*(0((0(0 \cup 1)^* \cup 10(0 \cup 1)^*) \cup 1(0 \cup 1)^*)).$$

3. Suponhamos, por contradição, que L seja aceita por um autômato finito determinístico com n estados. Vamos escolher $w = a^n b^n$, de modo que $|w| = 2n \geq n$. Aplicando o lema do bombeamento a w, obtemos que existe uma decomposição $w = xyz$ tal que

(1) $y \neq \epsilon$,
(2) $|xy| \leq n$,
(3) $xy^k z \in L$ para todo $k \geq 0$.

Escrevendo

$$x = a^i, \quad y = a^j \quad \text{e} \quad z = a^{n-i-j} b^n,$$

temos que $j > 0$ e que

$$xy^k z = a^i (a^j)^k a^{n-i-j} b^n = a^{n+(k-1)j} b^n \in L,$$

para todo $k \geq 0$. Mas, para que isto seja verdade, é preciso que

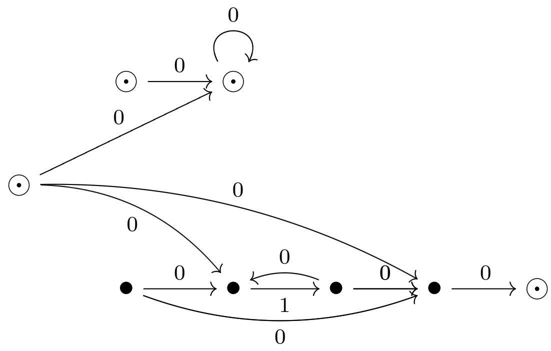
$$a^{n+(k-1)j} b^n,$$

seja da forma $a^m b^{2m}$ ou $a^m b^m$, para algum $m \geq 0$. Entretanto, se $k > 2$, então a palavra resultante do bombeamento tem

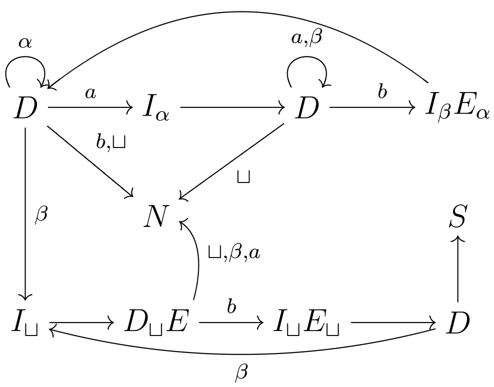
$$n + (k-1)j > n + j > n + 1$$

a's. Como o número de b's devia ser igual ou o dobro de as, então temos uma contradição, e podemos concluir que L não é regular.

4. O autômato resultante, depis de removidos os estados redundantes, é



4. Um diagrama possível para esta máquina é o seguinte:



5. Como L' é recursiva, então seu complemento $\overline{L'}$ também é. Mas

$$L \setminus L' = L \cap \overline{L'}.$$

Como interseção de linguagens recursivas é recursiva, então $L \setminus L'$ é recursiva.

1. Seja L a linguagem no alfabeto $\{0,1\}$, formada pelas palavras em que 11 aparece no máximo uma vez. Por exemplo, a palavra 011010100001 pertence a L, mas 01110 e 011011 não pertencem.

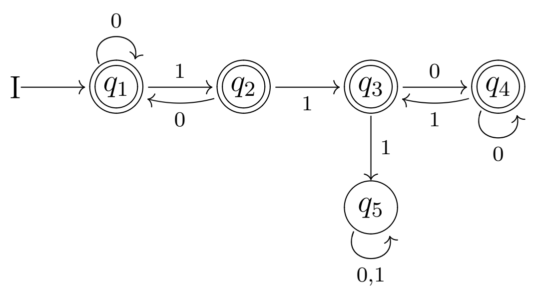
(1) Dê uma expressão regular que denote a linguagem L.
(2) Construa um autômato finito determinístico que aceite L.

Resolução

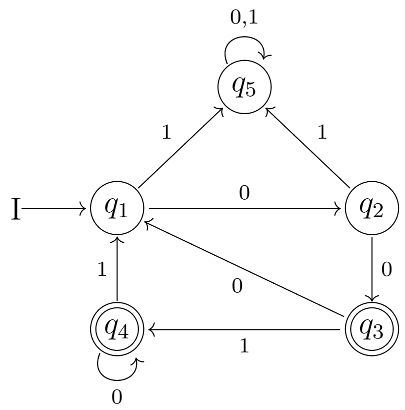
(1) Uma expressão regular para a linguagem dada é

$$(0 \cup 10)^* 11 (0 \cup 01)^* \cup (0 \cup 10)^* (0 \cup 01)^*.$$

(1) Um autômato finito determinístico que aceita L é dado pelo grafo



2. Use o algoritmo de substituição para achar a expressão regular da linguagem aceita pelo autômato finito determinístico cujo grafo é dado abaixo:



Resolução

Levando em conta que q_5 é um estado morto, temos que $L_5 = \emptyset$. Portanto, as equações são dadas por

$$\begin{aligned} L_1 &= 0L_2 \\ L_2 &= 0L_3 \\ L_3 &= 0L_1 \cup 1L_4 \cup \epsilon \\ L_4 &= 0L_4 \cup 1L_1 \cup \epsilon. \end{aligned}$$

Aplicando o lema de Arden à última equação, obtemos

$$L_4 = 0^*(1L_1 \cup \epsilon).$$

Substituindo nas anterior:

$$L_3 = 0L_1 \cup 10^*(1L_1 \cup \epsilon) \cup \epsilon = (0 \cup 10^*1)L_1 10^* \cup \epsilon.$$

Portanto,

$$L_1 = 0L_2 = 0^2L_3 = 0^2((0 \cup 10^*1)L_1 10^* \cup \epsilon) = (0^3 \cup 0^2 10^*1)L_1 0^2 10^* \cup 0^2.$$

Aplicando o lema de Arden mais uma vez:

$$L_1 = (0^3 \cup 0^2 10^*1)^*(0^2 10^* \cup 0^2),$$

que é a linguagem aceita pelo autômato dado.

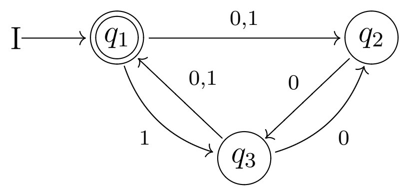
Teste 2

1. Dê exemplo de um autômato finito não determinístico M no alfabeto $\{0,1\}$, que tenha 3 estados, e para o qual vale a seguinte propriedade:

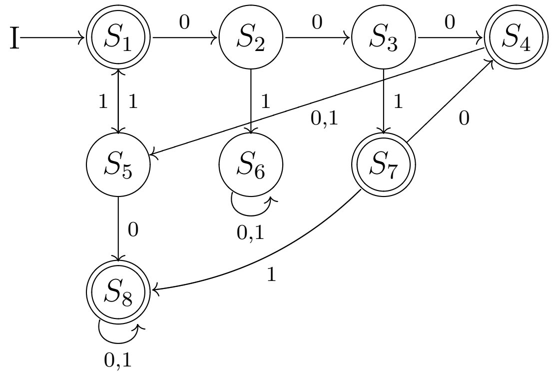
> aplicando a M a construção de subconjuntos e *descartando os estados redundantes*, obtemos um autômato finito determinístico com $2^3 = 8$ estados.

Resolução

Há muitos exemplos de autômatos satisfazendo esta condição, um deles é o seguinte:



O autômato finito determinístico resultante é o seguinte:



onde $S_1 = \{q_1\}$, $S_2 = \{q_2\}$, $S_3 = \{q_3\}$, $S_4 = \{q_1, q_2\}$, $S_5 = \{q_2, q_3\}$, $S_6 = \emptyset$, $S_7 = \{q_1, q_3\}$ e $S_8 = \{q_1, q_2, q_3\}$

2. Seja L a linguagem no alfabeto $\{0,1\}$ formada pelas palavras $w \in (00 \cup 111)^*$ para as quais a quantidade de 0s excede a quantidade de 1s. Mostre, usando o lema do bombeamento, que L não pode ser aceita por nenhum autômato finito determinístico.

Resolução

Suponhamos, por contradição, que L seja aceita por um autômato finito determinístico com n estados. Precisamos achar uma palavra w em L com comprimento maior que n. Para estar em L, w precisa ser formada a partir de 00s e 111s e ter mais zeros que uns. Com isto o número de uns tem que ser múltiplo de 3, e o número de zeros tem que ser múltiplo de 2. Uma escolha possível é

$$w = 1^{3n}0^{6n} \in L.$$

Então $|w| = 9n \geq n$. Aplicando o lema do bombeamento a w, obtemos que existe uma decomposição $w = xyz$ tal que

(1) $y \neq \epsilon$,
(2) $|xy| \leq n$,
(3) $xy^kz \in L$ para todo $k \geq 0$.

Escrevendo

$$x = 1^i, \quad y = 1^j \quad \text{e} \quad z = 1^{3n-i-j}0^{6n},$$

temos que $j > 0$ e que

$$xy^kz = 1^i(1^j)^k1^{3n-i-j}d^{6n} = 1^{3n+(k-1)j}0^{6n} \in L,$$

para todo $k \geq 0$. Mas, escolhendo $k = 3n+1$, a palavra resultante do bombeamento tem

$$3n + (k-1)j = 3n + (3n)j = 3n(j+1)$$

uns. Como $j > 0$, segue-se que $3n(j+1) \geq 3n$, de modo que esta palavra não pode estar em L, o que nos dá uma contradição. Portanto, podemos concluir que L não é regular.

Observação: A razão pela qual escolhemos $1^{3n}0^{6n}$ (em vez de $0^{6n}1^{3n}$) é que, ao fazer isto, os 1s serão bombeados, aumentando de quantidade e levando a uma contradição. Para chegar a uma contradição, aumentando a quantidade de zeros, precisamos garantir que a propor cão entre os 0s e 1 é alterada, o que é mais difícil de fazer.

TESTE 3

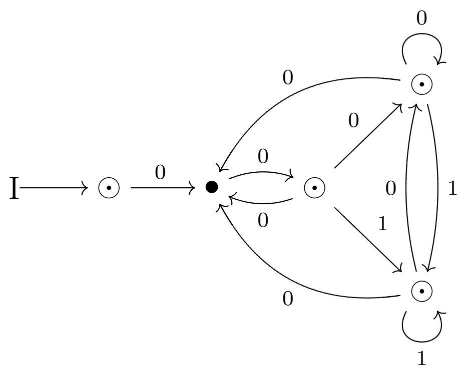
1. Construa, passo a passo (começando dos autômatos que aceitam apenas um símbolo), um autômato finito não determinístico que aceite a linguagem denotada pela expressão regular

$$((0 \cdot 0) \cdot (0^* \cup 1^*))^*.$$

Indique cada passo da construção separadamente e elimine os estados redundantes do autômato à medida que for efetuando a construção.

Resolução

O autômato resultante, depois de eliminados os estados redundantes, é descrito pelo seguinte grafo:



2. Decida se cada uma das duas afirmações abaixo é verdadeira ou falsa. Caso a afirmação seja verdadeira, prove sua resposta; caso seja falsa, dê um contra-exemplo.

Afirmação 1: Uma gramática linear à direita com apenas uma regra não pode gerar uma linguagem com mais de uma palavra.

Afirmação 2: Para que uma gramática linear à direita gere uma linguagem infinita ela precisa ter pelo menos três regras.

Resolução

A afirmação 1 é verdadeira. De fato, para que uma gramática linear à direita gere uma linguagem, é preciso que haja uma regra que leve à redução de pelo menos uma das variáveis apenas a terminais. Por isso, se a gramática tem apenas uma regra, a regra tem que ser da forma $X \to w$, onde w é uma palavra apenas nos terminais da gramática. Contudo, a linguagem gerada por uma tal regra é $\{w\}$, que tem apenas uma palavra.

A afirmação 2 é falsa. A gramática com terminal 0, símbolo inicial (e única variável) S, e regras $S \to 0S \mid 0$, gera a linguagem infinita denotada por 0^*.

DCC-UFRJ–Linguagens Formais–2005/2–Teste 4

Nome: ____

Justifique cuidadosamente as suas respostas.

1. Dê exemplo de uma gramática livre de contexto que gere a linguagem no alfabeto $\{a, b\}$ definida por

$$L = \{a^i b^j : i - j = \pm 2\}.$$

Resolução

Esta linguagem pode ser decomposta como a união de duas, a saber

$$L_+ = \{a^i b^j : i - j = 2\} \text{ e } L_- = \{a^i b^j : i - j = -2\},$$

que podemos reescrever na forma

$$L_+ = \{a^{j+2} b^j : j \geq 0\} \text{ e } L_- = \{a^i b^{i+2} : i \geq 0\}.$$

Para a primeira linguagem, temos a gramática

Terminais: $\{a, b\}$.
Variáveis: $\{S\}$.
Símbolo inicial: S
Regras: $\{S \to aSb, S \to a^2\}$

e, para a segunda,

Terminais: $\{a, b\}$.
Variáveis: $\{S'\}$.

Símbolo inicial: S'
Regras: $\{S' \to aS'b, S \to b^2\}$

Colando as duas em uma união, obtemos

Terminais: $\{a, b\}$.
Variáveis: $\{S_0, S, S'\}$.
Símbolo inicial: S_0
Regras: $\{S_0 \to S, S_0 \to S', S \to aSb, S \to a^2S' \to aS'b, S \to b^2\}$

2. Considere a gramática $G_{u,v}$ com terminais T, cuja única variável é o símbolo inicial S, e cujas regras são

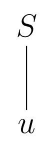
$$S \to uS \mid Sv \mid u \mid v,$$

onde u e v são palavras em T^*. Note que a gramática varia dependendo da escolha de u e v.

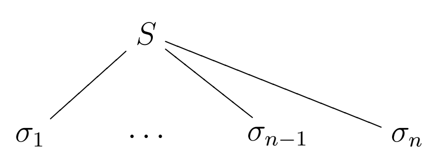
(1) Prove que a gramática $G_{u,v}$ é ambígua para toda escolha de u e v.
(2) Construa uma gramática $\widehat{G}_{u,v}$ que não é ambígua e que gera a mesma linguagem que $G_{u,v}$. Justifique cuidadosamente sua resposta.

Resolução

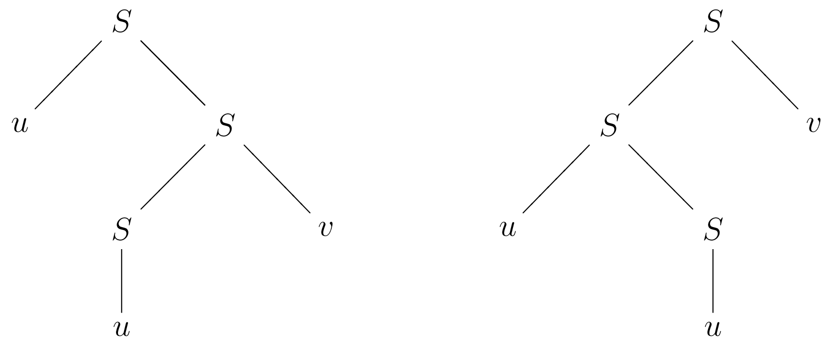
(1) Na construção das árvores abaixo usaremos a seguinte notação. Se $u = \sigma_1 \cdots \sigma_n$, onde $\sigma_1, \ldots, \sigma_n \in T$, escreveremos



para denotar a árvore



Com esta notação, temos as seguintes duas árvores distintas, cuja colheita é u^2v:



(2) Para construir $\widehat{G}_{u,v}$ basta escolher uma nova variável T e definir uma ordem para o uso destas variáveis. Digamos que S só pode ser usada antes de T. Então a gramática $\widehat{G}_{u,v}$ terá os mesmos terminais que $G_{u,v}$, suas variáveis serão S e T, das quais S é o símbolo inicial, e as regras serão

$$S \to uS \,|\, T \,|\, u$$
$$T \to Tv \,|\, v.$$

Cada palavra de $L(G_{u,v})$ é da forma u^nv^m, com $n \geq 0$ e $m \geq 1$, ou $n \geq 1$ e $m \geq 0$, já que as regras nos obrigam a pôr os us sempre à direita dos vs, e tem que haver pelo menos um u ou um v. Mas, podemos gerar estas palavras a partir de $\widehat{G}_{u,v}$, como segue

$$S \Rightarrow^n u^nS \Rightarrow u^nT \Rightarrow^{m-1} u^nv^{m-1}T \Rightarrow u^nv^m,$$

a não que $n = 0$, quando teremos

$$S \Rightarrow T \Rightarrow^{m-1} v^{m-1}T \Rightarrow v^m,$$

ou que $m = 0$, quando teremos

$$S \Rightarrow^n u^nS \Rightarrow u^n.$$

Note que esta é a única derivação possível para u^nv^m, porque podemos passar de S para T, mas não podemos voltar a S depois de chegar a T. Isto nos obriga a sempre pôr todos os us antes de pôr qualquer vs.

Teste 5

1. Considere a linguagem no alfabeto $\{a, b, c\}$ definida por

$$L = \{a^n b^r c^n : n, r \geq 0 \ \text{ e } \ r \geq n\},$$

(1) Prove, usando o lema do bombeamento para linguagens livres de contexto, que L não é livre de contexto.

(2) Construa uma máquina de Turing que *decida* L, esboçando o seu diagrama a partir das máquinas elementares.

Resolução

(1) Suponhamos, por contradição, que L seja livre de contexto. Então, pelo lema do bombeamento, existe um inteiro $\rho > 0$ tal que, se

$$w = a^\rho b^\rho c^\rho$$

então, como $|w| = 3\rho > \rho$, temos que w admite uma decomposição na forma

$$a^\rho b^\rho c^\rho = uvxyz,$$

onde:

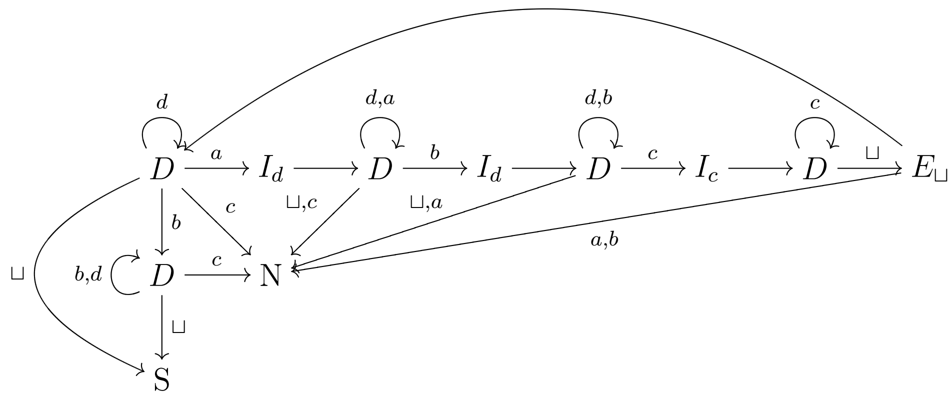
(1) $vy \neq \epsilon$;
(2) $|vxy| \leq \rho$;
(3) $uv^k xy^k z \in L$ para todo $k \geq 0$.

O ponto essencial a ser observado é que, por (2) a subpalavra vxy não pode conter simultaneamente as e cs. Isto significa que, ao efetuarmos o bombeamento, podem acontecer duas coisas:

- os as e/ou bs se alteram, mas não os cs;
- os cs e/ou bs se alteram, mas não os as;

Os dois casos são análogos, e basta analisar um deles, digamos o primeiro. Neste caso, bombeando com $k = 0$, obtemos uma palavra que perdeu alguns as e/ou alguns bs. Se a palavra perdeu as, temos menos as que cs; se perdeu bs, temos menos bs que cs. Contudo, o número de as tinha que ser sempre igual ao de cs, ao passo que o número de bs tinha que ser sempre maior ou igual ao de cs. Assim, qualquer que seja o caso, ao bombear, obtemos uma palavra que está fora da linguagem, de modo que ela não pode ser livre de contexto.

(2) Um diagrama possível é o seguinte:



2. Construa um autômato de pilha não determinístico, *com apenas um símbolo na pilha*, que aceite a linguagem no alfabeto $\{a, b, c\}$ definida por

$$L = \{a^n b^r c^n : n, r \geq 0 \ \text{ e } \ r \geq 3\}.$$

Resolução

O autômato de pilha correspondente é dado por

Alfabeto de entrada: $\{a, b, c\}$.
Alfabeto da pilha: $\{a\}$.
Estados: $\{q_1, q_2, q_3, q_4, q_5\}$.
Estado inicial: q_0
Estados finais: $\{q_5\}$
Transição:

Estado	Entrada	Topo da pilha	Transição
q_1	a	ϵ	(q_1, a)
q_1	b	ϵ	(q_2, ϵ)
q_2	b	ϵ	(q_3, ϵ)
q_3	b	ϵ	(q_4, ϵ)
q_4	b	ϵ	(q_4, ϵ)
q_4	c	a	(q_5, ϵ)
q_5	c	a	(q_5, ϵ)

3. Dê exemplos de linguagens recursivamente enumeráveis L_1 e L_2 tais que:

   (1) $L_1 \setminus L_2$ é recursivamente enumerável.

   (2) $L_1 \setminus L_2$ não é recursivamente enumerável.

Resolução

(1) Basta tomar L_1 e L_2 regulares. Além disso, sabemos que a diferença de duas linguagens regulares é regular. Mas, pela hierarquia de Chomsky, toda linguagem regular é recursivamente enumerável. Logo, neste caso, temos uma diferença de linguagens recursivamente enumeráveis que dá recursivamente enumerável.

(2) Sabemos que existem linguagens que não são recursivamente enumeráveis. Um exemplo, é o complemento da linguagen $\mathcal{L}_0$, cosntruída na aula sobre máquina de Turing Universal. Seja, então, $L_1 = \overline{\mathcal{L}_0}$. Para L_2 tomamos $\emptyset$. Como $\emptyset$ é regular então, pela hierarquia de Chomsky, também é recursivamente enumerável. Portanto,

$$L_1 \setminus L_2 = \overline{\mathcal{L}_0} \setminus \emptyset = \overline{\mathcal{L}_0},$$

que não é recursivamente enumerável.

Obrigado a

- Douglas Cardoso

pelas correções no gabarito.