(8)

Universidade Federal de Uberlândia

1ª Lista de Exercícios de Linguagens Formais e Autômatos

2o Semestre de 2008 - Profa. Gina Maira B. de Oliveira

Material Complementar

Exercícios selecionados do livro do

Paulo B. Menezes

2.10 Exercícios

2.1 Sobre as Linguagens Regulares:

a) Qual a importância do seu estudo?

b) Exemplifique suas aplicações (para os diversos formalismos);

c) Você imagina algum tipo de linguagem cujo algoritmo de reconhecimento seja mais eficiente que o das Regulares? E menos eficiente? Explique a sua resposta.

2.2 Desenvolva Autômatos Finitos Determinísticos que reconheçam as seguintes linguagens sobre $\Sigma = \{a, b\}$:

a) { w | w possui aaa como subpalavra }

b) { w | o sufixo de w é aa }

c) { w | w possui número ímpar de a e b }

d) { w | w possui número par de a e ímpar de b ou w possui número par de b e ímpar de a }

e) { w | o quinto símbolo da direita para a esquerda de w é a }

2.3 Desenvolva Autômatos Finitos Não-Determinísticos, com ou sem movimentos vazios, que reconheçam as seguintes linguagens:

a) sobre o alfabeto $\Sigma = \{a, b, c\}$:

{ w | aa ou bb é subpalavra e cccc é sufixo de w }

b) sobre o alfabeto $\Sigma = \{a, b\}$:

b.1) { $w_1 w_2 w_1$ | w_2 é qualquer e $|w_1| = 3$ }

b.2) { w | o décimo símbolo da direita para a esquerda de w é a }

b.3) { w | w possui igual número de símbolos a e b e (qualquer prefixo de w possui, no máximo, dois a a mais que b ou qualquer prefixo de w possui, no máximo, dois b a mais que a) }

2.4 Desenvolva Expressões e Gramáticas Regulares que gerem as seguintes linguagens sobre $\Sigma = \{a, b\}$:

a) { w | w tem no máximo um par de a como subpalavra e no máximo um par de b como subpalavra }

b) { w | qualquer par de a antecede qualquer par de b }

c) { w | w não possui aba como subpalavra }

2.5 Represente a seguinte linguagem baseada em unidades léxicas da linguagem de programação Pascal (ou alguma outra de seu domínio), usando os formalismos Autômato Finito Determinístico, Expressão Regular e Gramática Regular:

{ w | w é número inteiro ou w é número real ou w é identificador da linguagem Pascal }

2.6 Descreva em palavras as linguagens geradas pelas seguintes Expressões Regulares:

a) $(aa + b)^*(a + bb)$

b) $(b + ab)^*(\varepsilon + a)$

c) $(aa + bb + (aa + bb)(ab + ba)(aa + bb))^*$

2.7 Aplique o algoritmo de tradução de formalismo de Expressão Regular para Autômato Finito:

a) $(ab + ba)^*(aa + bb)^*$

b) $ab(abb^* + baa^*)^*ba$

2.8 Aplique os algoritmos de tradução de formalismos apresentados e, a partir da Expressão Regular $(b + \varepsilon)(a + bb)^*$, realize as diversas etapas até gerar a Gramática Regular correspondente (ER → AFε → AFN → AFD → GR).

2.9 Minimize os Autômatos Finitos ilustrados na Figura 2.36.

2.10 Por que pode-se afirmar que um Autômato Finito Determinístico sempre pára (ao processar qualquer entrada)? O mesmo pode ser afirmado para o não-determinístico? E com movimentos vazios? Em particular, no caso do Autômato Finito com Movimentos Vazios, analise para a seguinte situação de ciclo (suponha que q e p são estados do autômato):

$\delta(q, \varepsilon) = p$

$\delta(p, \varepsilon) = q$

2.11 Complete a prova referente ao teorema: a classe dos Autômatos Finitos com Movimentos Vazios é equivalente à classe dos Autômatos Finitos Não-Determinísticos.

2.12 Demonstre a equivalência dos quatro tipos de Gramáticas Lineares.

2.13 Demonstre as seguintes propriedades das Expressões Regulares (suponha que r, s e t são Expressões Regulares):

a) *Comutatividade da União.*

$r + s = s + r$

6

simples,
composto,
enquanto-faça,
repita-até;

- comando simples:
  qualquer palavra de {a, b}*;
- comando composto:
  i (início),
  seguido de um ou mais comandos separados por ";",
  seguidos de t (término);
- comando enquanto-faça:
  e (enquanto),
  seguido de uma expressão,
  seguida de f (faça),
  seguida de um comando;
- comando repita-até:
  r (repita),
  seguido de um comando,
  seguido de a (até),
  seguida de uma expressão;
- expressão: como definida na linguagem L_7 no Exercício 3.2, excetuando-se a palavra vazia.

3.5 Considere a seguinte gramática:
$G = (\{S\}, \{a, b\}, P, S)$, onde:
$P = \{S \to SS \mid aSa \mid bSb \mid \varepsilon\}$

a) Qual a linguagem gerada?

b) A gramática é ambígua?

c) Para a palavra aabbaaaa:
- construa uma árvore de derivação;
- para a árvore construída, determine a derivação mais à esquerda e a mais à direita.

3.6 No Exemplo 5, foi afirmado que a gramática abaixo é ambígua:
$G_2 = (\{E\}, \{+, *, [,], x\}, P_2, E)$, onde:
$P_2 = \{E \to E{+}E \mid E{*}E \mid [E] \mid x\}$

Construa uma gramática não-ambígüa equivalente.

Sugestão: faça uma pesquisa bibliográfica e verifique como são definidas, usando gramáticas, expressões em algumas linguagens de programação reais. A definição de uma "expressão simples" na linguagem Pascal é um bom exemplo.

3.7 Para qualquer Linguagem Livre do Contexto é possível garantir que existe um Autômato com Pilha que aceita a linguagem e que sempre pára para qualquer entrada? Por quê?

3.8 Demonstre que se L é uma Linguagem Livre do Contexto, então L^* também é Livre do Contexto.

3.9 Demonstre que o Autômato com Pilha sem usar a estrutura de pilha para armazenar informações do processamento possui o mesmo poder computacional do Autômato Finito.

3.10 Estenda a função programa do Autômato com Pilha usando como argumento um estado e uma palavra, de forma similar à realizada para os Autômatos Finitos.

3.11 Considere a seguinte gramática:
$G = (\{S, X, Y, Z, A, B\}, \{a, b, u, v\}, P, S)$, onde:
$P = \{ S \to XYZ,$
$X \to AXA \mid BXB \mid Z \mid \varepsilon,$
$Y \to AYB \mid BYA \mid Z \mid \varepsilon,$
$A \to a,\ B \to b$
$Z \to Zu \mid Zv \mid \varepsilon\}$

a) Qual a linguagem gerada?

b) Simplifique a gramática.

3.12 Sobre os algoritmos de simplificação de Gramáticas Livres do Contexto:

a) por que, no algoritmo referente ao tratamento dos símbolos inúteis, se a etapa *qualquer símbolo é atingível a partir do símbolo inicial* for executada antes da etapa *qualquer variável gera palavra de terminais*, o resultado pode não ser o esperado?

b) por que a execução combinada dos algoritmos de simplificação (produções vazias, produções da forma $A \to B$ e símbolos inúteis) não deve ser realizada em qualquer ordem?

3.13 Para as gramáticas abaixo, construa as gramáticas equivalentes na Forma Normal de Chomsky e na de Greibach:

a) $L8 = \{w \mid w$ é Expressão Regular sobre o alfabeto $\{x\}\}$ introduzida no Exercício 3.3.

b) Gramática construída para a linguagem de programação do Exercício 3.4;

- parte do símbolo inicial S;
- foi incluída em D_0 ("/0");
- todo o lado direito da produção foi analisado com sucesso (o marcador "." está no final de α). □

Note-se que, para otimizar as etapas a) e b) do algoritmo acima, os ciclos repita-até podem ser restritos exclusivamente às produções recentemente incluídas em D_r ou em D_0 ainda não-analisadas.

EXEMPLO 19 Algoritmo de Early.

Considere a gramática G_2, introduzida no Exemplo 2, que gera expressões com colchetes balanceados e operação de adição e multiplicação:

$G_2 = (\{E\}, \{+, *, [,], x\}, P_2, E)$, onde:

$P_2 = \{E \to E+E \mid E*E \mid [E] \mid x\}$

O reconhecimento da palavra $x*x$ é como segue:

D_0:

$E \to .T/0$	produções que partem
$E \to .E+T/0$	do símbolo inicial;
$T \to .F/0$	produções que podem ser aplicadas
$T \to .T*F/0$	em derivação mais à esquerda
$F \to .(E)/0$	a partir do símbolo inicial.
$F \to .x/0$	

D_1: reconhecimento de x em $\underline{x}*x$

$F \to x./0$	x foi reduzido à F;
$T \to F./0$	inclui todas as produções de D_0 que
$T \to T.*F/0$	referenciaram .F direta ou indiretamente, (pois $F \to x./0$)
$E \to T./0$	movendo o marcador "."
$E \to E.+T/0$	um símbolo para a direita.

D_2: reconhecimento de * em $x*x$

$T \to T*.F/0$	gerou *; o próximo será gerado por F;
$F \to .(E)/2$	inclui todas as produções de P que
$F \to .x/2$	podem gerar o próximo terminal a partir de F.

D_3: reconhecimento de x em $x*\underline{x}$

$F \to x./2$	x foi reduzido à F;
$T \to T*F./0$	incluído de D_2 (pois $F \to x./2$); a entrada foi reduzida à T;
$E \to T./0$	incluído de D_0 (pois $T \to T*F./0$); a entrada foi reduzida à E;
$T \to T.*F/0$	incluído de D_0 (pois $T \to T*F./0$);
$E \to E.+T/0$	incluído de D_0 (pois $E \to T./0$).

Como $w = x*x$ foi reduzida ao símbolo inicial E, ou seja, $E \to T./0$ pertence a D_3, a entrada foi aceita. □

3.10 Exercícios

3.1 Sobre as Linguagens Livres do Contexto:

a) Qual a importância do seu estudo?

b) Exemplifique suas aplicações (para os formalismos de autômato e gramática);

c) Faça um quadro comparativo com as Linguagens Regulares, destacando as principais características, semelhanças e diferenças.

3.2 Desenvolva Gramáticas Livres do Contexto que gerem as seguintes linguagens:

a) $L_1 = \varnothing$

b) $L_2 = \{\varepsilon\}$

c) $L_3 = \{a, b\}^*$

d) $L_4 = \{w \mid w$ é palíndromo em $\{a, b\}^*\}$, onde palíndromo significa que $w = w^r$

e) $L_5 = \{ww^r \mid w$ é palavra de $\{a, b\}^*\}$. Qual a diferença entre as linguagens L_4 e L_5?

f) $L_6 = \{a^i b^j c^k \mid i = j$ ou $j = k$ e $i, j, k \geq 0\}$

g) $L_7 = \{w \mid w$ é palavra de $\{x, y, (,)\}^*$ com parênteses balanceados$\}$

h) $L_8 = \{w \mid w$ é Expressão Regular sobre o alfabeto $\{x\}\}$

3.3 Desenvolva Autômatos com Pilha que reconheçam as seguintes linguagens:

a) $L_1 = \varnothing$

b) $L_2 = \{\varepsilon\}$

c) $L_3 = \{a, b\}^*$

d) $L_4 = \{w \mid w$ é palíndromo em $\{a, b\}^*\}$

e) $L_8 = \{w \mid w$ é Expressão Regular sobre o alfabeto $\{x\}\}$

f) $L_9 = \{ua^n va^n w \mid n \in \{1, 2\}, u, v, w$ são palavras de $\{a, b\}^*$ e $|u| = |v| = 5\}$

3.4 Construa uma Gramática Livre do Contexto e um Autômato com Pilha que representem a seguinte linguagem de programação:

- os comandos podem ser como segue:

c) Faça um comparativo das gramáticas originais com as correspondentes na Forma Normal de Chomsky e na Forma Normal de Greibach.

3.14 Explique intuitivamente por que e prove que as seguintes linguagens não são Livres do Contexto:

a) $L_{10} = \{ww \mid w \text{ é palavra de } \{a, b\}^*\}$

b) $L_{11} = \{a^n b^n a^m \mid n \geq 0, m \geq 0 \text{ e } n \neq m\}$

3.15 Demonstre que a Classe das Linguagens Livres do Contexto é fechada para as seguintes operações:

a) União, usando o formalismo de gramática;

b) Concatenação, usando o formalismo de autômato.

3.16 As linguagens geradas pelas gramáticas cujas produções estão representadas abaixo são vazias, finitas ou infinitas?

a)

$S \to AB \mid CA$
$A \to a$
$B \to BC$
$C \to AB \mid \varepsilon$

b)

$S \to aS \mid aSbS \mid X$
$X \to SS$

3.17 Por que os algoritmos de reconhecimento baseados em Autômatos com Pilha são tão ineficientes em termos de tempo de processamento?

3.18 No algoritmo de reconhecimento Autômato com Pilha Descendente, qual a conseqüência se a gramática usada tiver recursão à esquerda?

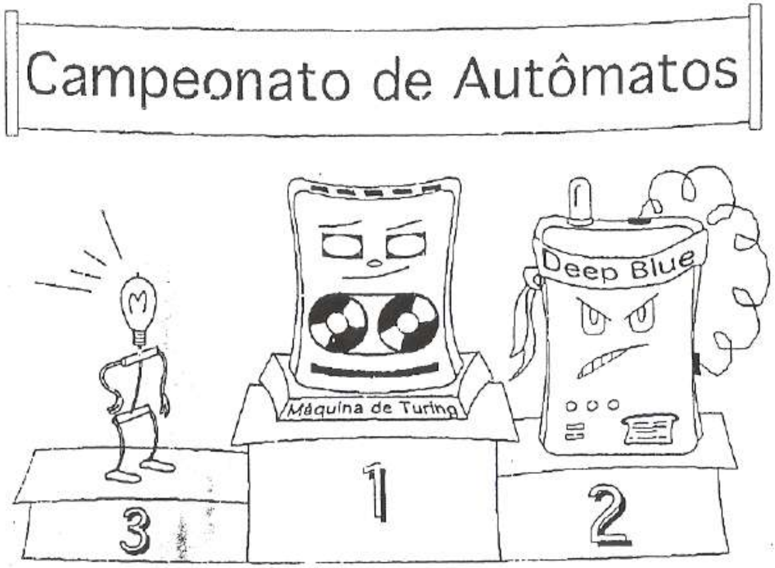
3.19 Para as gramáticas da linguagem $L8 = \{w \mid w \text{ é Expressão Regular sobre o alfabeto } \{x\}\}$ construídas no Exercício 3.13, faça o reconhecimento da entrada $(x+x)^*$ para cada um dos seguintes algoritmos de reconhecimento:

a) Autômato com Pilha a partir da gramática na Forma Normal de Greibach;

b) Autômato com Pilha Descendente;

c) Cocke-Younger-Kasami (CYK);

d) Early.



4 Linguagens Enumeráveis Recursivamente e Sensíveis ao Contexto

Ciência da Computação é o conhecimento sistematizado relativo à computação. Sua origem é remota, tendo exemplos na antiga Grécia (século III a.C., no desenho de algoritmos por Euclides) e Babilônia (com estudos sobre complexidade e reducibilidade de problemas). No início do século XX, diversas pesquisas foram desenvolvidas com o objetivo de definir um modelo computacional suficientemente genérico, capaz de implementar qualquer "função computável".

Em 1936, Alan Turing propôs um modelo conhecido como *Máquina de Turing*. Atualmente, a Máquina de Turing é aceita como uma formalização de um *procedimento efetivo* (*algoritmo* ou *função computável*), ou seja, uma seqüência finita de instruções, as quais podem ser realizadas mecanicamente, em um tempo finito.

5