Pós-Graduação em Ciência da Computação — DCC/ICEx/UFMG
Teoria de Linguagens — $2^{\underline{o}}$ semestre de 2006
Professor: Newton José Vieira
Terceira Lista de Exercícios — Entrega: até 20/11/06

Todas as MT's referidas nesta lista podem conter qualquer incremento (várias fitas, não determinismo, etc.), a menos que se diga o contrário.

1. Construa uma MT padrão que gere todas as palavras de $\{\texttt{0}^n\texttt{1}^{2n} \mid n \geq 0\}$ em ordem de tamanho. Como essa linguagem é infinita, a MT não deve parar nunca. Ela deve gerar λ, `011`, depois `001111`, depois `000111111` etc., separados por branco, indefinidamente. A fita deverá ficar assim:

$$\langle \sqcup \texttt{011} \sqcup \texttt{001111} \sqcup \texttt{000111111} \sqcup \cdots$$

(Observe que o primeiro $\sqcup$ separa λ de `011`.)

2. Mostre como se pode construir uma MT que reconhece uma linguagem constituída das palavras geradas por uma MT como a da questão anterior. Ou seja, suponha dada uma MT M que gera as palavras de uma linguagem L na forma:

$$\langle w_0 \sqcup w_1 \sqcup w_2 \sqcup \cdots$$

Se L for finita, após alguma palavra w_n ($n \geq 0$), virá dois $\sqcup$s contíguos. Caso contrário, a MT não pára, eventualmente gerando $w_n \in L$ para qualquer $n \geq 0$... O problema é mostrar como obter, a partir de M, uma MT que reconhece L.

3. Construa uma MT que some dois números naturais em notação binária. Por exemplo, se o conteúdo inicial da fita de entrada for $\langle \underline{\texttt{1}}\texttt{01} \sqcup \texttt{1101}$, após o processamento, seu conteúdo será $\langle \underline{\texttt{1}}\texttt{0010}$. Se usar outra(s) fita(s), seu(s) conteúdo(s) ao final do processamento não será(ão) importante(s).

4. Construa MT's, com o menor número de transições possível, que reconheçam <u>por parada</u> as seguintes linguagens de alfabeto $\{\texttt{0}, \texttt{1}\}$:
   (a) O conjunto das palavras que começam com `0`.
   (b) O conjunto das palavras que não começam com `0`.

5. Escreva MT's que reconheçam as linguagens:
   (a) $\{\texttt{a}^m\texttt{b}^n\texttt{a}^m\texttt{b}^n \mid m, n \in \mathbf{N}\}$.
   (b) $\{\texttt{a}^{n^2} \mid n \in \mathbf{N}\}$.
   (c) $\{\texttt{0}^n \mid n \text{ não é primo}\}$.

6. Construa uma gramática irrestrita que gere $\{\texttt{a}^m\texttt{b}^n\texttt{a}^m\texttt{b}^n \mid m, n \in \mathbf{N}\}$.

7. Mostre que para toda gramática irrestrita existe uma equivalente na qual cada regra tem o lado direito maior ou igual ao lado esquerdo ou é regra λ.

8. Mostre que para toda gramática irrestrita que não gere λ existe uma equivalente na qual cada regra tem uma das formas (A, B e C são variáveis e a é terminal):

$$A \rightarrow a$$
$$A \rightarrow aB$$
$$A \rightarrow aBC$$

9. Sejam L uma LRE e R uma linguagem recursiva. Mostre:
   (a) $L - R$ é uma LRE.
   (b) $L - R$ pode não ser recursiva.
   (c) $R - L$ pode não ser uma LRE.
   (d) $R - L$ pode ser regular.

10. Mostre que $\{R\langle M \rangle \mid M$ é MT e $oi \ \in L(M)\}$ é LRE.

11. Mostre que é decidível ou que não é:
    (a) Dada uma MT M e uma palavra w, determinar se M escreve algum símbolo diferente do branco, ao processar w.
    (b) Dada uma MT M e uma palavra w, determinar se M lê algum branco ao processar w.

12. Para cada PD abaixo, mostre que o mesmo é indecidível:
    (a) Dada uma MT M, determinar se a computação de M para a entrada λ escreve oi.
    (b) Dados uma MT M, uma palavra w e um estado e de M, determinar se a computação de M para a entrada w atinge o estado e.
    (c) Dada uma MT M, determinar se a computação de M para a entrada λ "volta" ao estado inicial de M.
    (d) Dados uma MT M e um símbolo a de M, determinar se a computação de M para a entrada a escreve a na fita em algum momento.
    (e) Dados uma MT M e uma expressão regular r, determinar se $L(M) \cap L(r) = \emptyset$.

13. Mostre que os seguinte problemas são ou não decidíveis:
    (a) Dadas duas GLCs G_1 e G_2, determinar se $L(G_1) \cap L(G_2)$ é finito.
    (b) Dado um AP M, determinar se $\overline{L(M)}$ é finito.
    (c) Dadas uma GR G_R e uma GLC G_L, determinar se $L(G_R) \cap L(G_L) = \emptyset$.
    (d) Dadas uma GR G_R e uma GLC G_L, determinar se $L(G_R) \subseteq L(G_L)$.