Algoritmos e Fundamentos da Teoria de Computação

Lista de Exercícios 01

1 Seja M a máquina de Turing definida pela função δ abaixo.

δ	B	a	b	c
q_0	q_1, B, R			
q_1	q_2, B, L	q_1, a, R	q_1, c, R	q_1, c, R
q_2		q_2, c, L		q_2, b, L

a. Construa o *trace* da computação de M para a *string* de entrada $aabca$.

b. Construa o *trace* da computação de M para a *string* de entrada $bcbc$.

c. Apresente o diagrama de estados de M.

d. Descreva o resultado de uma computação de M.

2 Construa uma máquina de Turing que realiza as computações pedidas nos itens abaixo. (Faça uma máquina para cada item.) Todas as máquinas têm alfabeto de entrada $\Sigma = \{a, b\}$. Note que a cabeça da máquina deve sempre estar na posição 0 da fita quanto a computação termina no estado q_f.

a. Mover a entrada uma posição para a direita: $q_0 BuB \overset{*}{\vdash} q_f BBuB$, aonde $u \in \Sigma^*$.

b. Concatenar uma cópia invertida à *string* de entrada: $q_0 BuB \overset{*}{\vdash} q_f Buu^R B$, aonde u^R é o reverso da *string* u.

c. Inserir um branco entre cada um dos símbolos da entrada, por exemplo: $q_0 BabaB \overset{*}{\vdash} q_f BaBbBaB$.

d. Apagar os b's da entrada, por exemplo: $q_0 Bbabaabab \overset{*}{\vdash} q_f BaaaaB$.

3 Construa uma máquina de Turing que computa as funções especificadas nos itens abaixo. (Faça uma máquina para cada item.) Todas as máquinas têm alfabeto de entrada $\Sigma = \{a, b\}$. Os símbolos u e v representam *strings* arbitrárias sobre Σ^*.

a. $f(u) = aaa$

b. $f(u) = \begin{cases} a & \text{se } \textit{length}\,(u) \text{ é par} \\ b & \text{caso contrário} \end{cases}$

c. $f(u) = u^R$

d. $f(u, v) = \begin{cases} u & \text{se } \textit{length}\,(u) > \textit{length}\,(v) \\ v & \text{caso contrário} \end{cases}$

4 Construa uma máquina de Turing que computa as funções numéricas especificadas nos itens abaixo. (Faça uma máquina para cada item.) Não utilize macros nas construções. Utilize a base de representação que preferir.

a. $f(n) = 2n + 3$

b. $eq(n, m) = \begin{cases} 1 & \text{se } n = m \\ 0 & \text{caso contrário} \end{cases}$

5 Use as macros e máquinas definidas entre as seções 9.2 e 9.4 do livro do Sudkamp para projetar uma máquina que computa a função $f(n) = 2n + 3$.

6 Seja F uma máquina de Turing que computa uma função numérica unária e total f. Projete uma máquina M que retorna o primeiro número natural n tal que $f(n) = 0$. A computação de M deve continuar indefinidamente se tal n não existe. Responda os itens abaixo.

a. Apresente M e explique o seu funcionamento.

b. O que aconteceria com a execução de M se a função computada por F não fosse total?

Obs.: Para construir M *você pode utilizar qualquer máquina ou macro vista até aqui.*

7 Seja F uma máquina de Turing que computa a função numérica unária e total f. Projete uma máquina G que computa a função

$$g(n) = \sum_{i=1}^{n} f(i) \quad .$$

Obs.: Para construir G *você pode utilizar qualquer máquina ou macro vista até aqui.*

8 Sejam F e G máquinas de Turing que computam, respectivamente, as funções numéricas unárias e totais f e g. Projete uma máquina H que computa a função

$$h(n) = \sum_{i=1}^{n} eq(f(i), g(i)) \quad .$$

Isto é, $h(n)$ é a quantidade de valores entre 1 e n para os quais as funções f e g assumem o mesmo valor.
Obs.: Para construir H *você pode utilizar qualquer máquina ou macro vista até aqui.*