

Algoritmos e Fundamentos da Teoria de Computação

Lista de Exercícios 04

- 1 Considere o problema de decisão cujas instâncias são definidas pela pergunta: “O número natural n é ímpar?” Defina uma *representação* para as instâncias deste problema e a seguir apresente uma máquina de Turing que decide as instâncias codificadas.

- 2 Seja L a linguagem contendo apenas uma única *string* s , aonde

$$s = \begin{cases} 0 & \text{se nunca será encontrada vida em Marte.} \\ 1 & \text{se algum dia será encontrada vida em Marte.} \end{cases}$$

A linguagem L é decidível? Justifique adequadamente sua resposta.

Obs.: Para os propósitos deste problema, assuma que a questão de existência de vida em Marte possui uma resposta definitiva ‘sim’ ou ‘não’.

- 3 Usando a representação unária de números naturais, projete uma TM que decide se um número natural é primo. *Obs. 1:* Assuma a existência de uma TM DIV que recebe como entrada $BmBnB$ aonde $m, n > 0$ estão escritos em notação unária na fita. A máquina DIV retorna como saída o resultado m/n se este for um número natural e 0, caso contrário. *Obs. 2:* Apresente a sua solução como um algoritmo, isto é, não é necessário apresentar o diagrama completo da máquina, só descrever como ela funciona.

- 4 Considere o problema de decisão abaixo.

DAG (Directed Acyclic Graph)

Input: um grafo direcionado G .

Output: sim; se G é acíclico.
não; caso contrário.

Esse problema **DAG** é decidível, semi-decidível ou indecidível?

1. Se a sua resposta for *decidível*, você deve apresentar uma máquina de Turing que *sempre termina* e responde sim ou não para as instâncias do problema.
2. Se a sua resposta for *semi-decidível*, você deve apresentar uma máquina de Turing que *sempre termina para as instâncias cuja resposta é sim*. Para as instâncias cuja resposta é não, a máquina não precisa terminar.
3. Se a sua resposta for *indecidível*, você deve *provar* porque é impossível projetar uma máquina de Turing para decidir esse problema.

Obs.: Nos casos 1 ou 2, você pode descrever a máquina de Turing como um algoritmo, isto é, não é necessário apresentar o diagrama completo da máquina, só descrever como ela funciona.

- 5 Dada uma TM M arbitrária com *string* de entrada w , é possível determinar se a computação de M para a entrada w *termina* em menos de 100 transições. Descreva uma máquina de Turing que resolve esse problema de decisão.
- 6 Mostre que o problema de decisão abaixo é decidível.

Input: máquina de Turing M

Output: sim; se existe uma *string* $w \in \Sigma^*$ para a qual a computação de M leva mais de 10 transições.
não; caso contrário.

Modifique a máquina de Turing universal apresentada no slide 25 da Aula 04 para realizar a verificação do problema acima. Apresente a máquina como um algoritmo, como feito no slide.