

Simulador de Máquinas de Turing

As Máquinas de Turing são processadores simbólicos genéricos, dotados de um cabeçote gravador/leitor, que operam sobre uma fita de entrada discretizada em células sucessivas, em que cada célula contém um único símbolo. As computações das Máquinas de Turing sobre os símbolos das células são extremamente limitadas. A cada transição de estado, elas podem:

1. gravar um símbolo na posição do cabeçote, ou
2. mover o cabeçote de leitura uma célula para a esquerda, ou
3. mover o cabeçote de leitura uma célula para a direita.

Há muitas formas de se representar Máquinas de Turing. Uma delas é o diagrama de estado, outra é uma tupla, que caracteriza uma máquina formalmente, conforme abaixo:

$(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$, onde Q, Σ, Γ são todos conjuntos finitos e

1. Q é o conjunto de estados da máquina, representados pelos números 0, 1, 2,...
2. Σ é o alfabeto de entrada, sem o **símbolo em branco** \sqcup .
3. Γ é o alfabeto de fita, onde $\sqcup \in \Gamma$
4. δ é um mapeamento $Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$
5. $q_0 \in Q$ é o estado inicial da máquina, representado por zero
6. $q_{aceita} \in Q$ estado de aceitação
7. $q_{rejeita} \in Q$ estado de rejeição onde $q_{aceita} \neq q_{rejeita}$

Por questões de conveniência as transições para o estado $q_{rejeita}$ são **implícitas** sobre os símbolos dos alfabetos (Γ) que não aparecem nos estados, considere a descrição formal da Máquina Turing M1, a função de transição δ de M1 é apresentada na Tabela 1.

$$Q = \{q_1, q_2, q_3, q_4, q_{aceita}\}$$

$$\Sigma = \{a, b\}$$

$$\Gamma = \{a, b, x, y, \sqcup\}$$

$$q_0 = q_1$$

Estando no estado	lendo	gravar na fita	Mover fita	vá para o estado
q ₁	a	x	D	q ₂
q ₁	y	y	D	q ₄
q ₁	␣	␣	D	q _{aceita}
q ₂	a	a	D	q ₂
q ₂	y	y	D	q ₂
q ₂	b	y	E	q ₃
q ₃	a	a	E	q ₃
q ₃	y	y	E	q ₃
q ₃	x	x	D	q ₁
q ₄	y	y	D	q ₄
q ₄	␣	␣	D	q _{aceita}

Tabela 1: Transições de M1

A Tabela 1, que tem 5 colunas, especificando: o estado em que a máquina se encontra (coluna 1); o símbolo corrente na cabeça de leitura (coluna 2); símbolo que será gravado na fita (coluna 3), ação na fita: mover o cabeçote para a esquerda [E], mover o cabeçote para a direita [D] (coluna 4); e finalmente a transição para o próximo estado (coluna 5), que pode ser um estado de aceitação.

Objetivo

Escreva um programa que simule uma Máquina de Turing genérica, o qual recebe como entrada um arquivo texto composto de:

1. Na primeira linha, os símbolos do alfabeto de entrada (Σ), com letras sempre em **minúsculo** e sem espaços em branco entre os símbolos, concatenados com símbolos do alfabeto fita (Γ), sem o símbolo em branco \sqcup , considere que teremos no máximo 30 símbolos na primeira linha.
2. Na segunda linha, o número de estados, para facilitar a implementação o estado de aceitação sempre será o último estado, no exemplo da máquina M1 o estado de aceitação (q_{aceita}) será o 5, considere também que o **estado inicial** será sempre o primeiro estado e qualquer máquina terá no **máximo 50 estados**.
3. Na terceira linha, o número de n transições da máquina e nas próximas n linhas seguem as transições especificadas no seguinte formato, separadas por espaço em branco:

<estado corrente> branco <símbolo em Γ > branco <gravar na fita> branco <mover> branco <estado chegada>

onde as ações de mover a cabeça de leitura são representadas pelas letras **E** (move para a esquerda), **D** (move para a direita). O símbolo em branco \sqcup será representado pelo caractere – (hífen).

4. Ao final das n transições, teremos uma linha com um inteiro especificando o número de palavras que deverão ser testadas na máquina de Turing, as palavras estão armazenadas na fita da máquina e não conterão espaços em branco e estarão cada uma em uma linha contendo os símbolos do alfabeto de entrada (Σ). Considere também, que a fita terá no máximo 100 caracteres.

Saída

Este programa deverá, conforme especificado no arquivo de entrada, processar a fita, usando a máquina de Turing lida, e depois, escrever o resultado na tela com as seguintes sentenças: o ordem da fita processada, a palavra lida na fita e a informação "OK" (para o caso de a máquina tenha parado o processamento no estado de aceitação e "not OK" caso contrário.

Exemplo de entrada

```
abxy
5
11
1 a x D 2
1 y y D 4
1 - - D 5
2 a a D 2
2 y y D 2
2 b y E 3
3 a a E 3
3 y y E 3
3 x x D 1
4 y y D 4
4 - - D 5
4
aabb
aaabbb
abab
bbaa
```

Exemplo de saída

```
1: aabb OK
2: aaabbb OK
3: abab not OK
4: bbaa not OK
```

Observações importantes:

O programa deve estar bem documentado e implementado na linguagem C/C++ e pode ser feito em grupo de até **2 alunos**, não esqueçam de colocar o **nome dos integrantes** do grupo no fonte do trabalho. A entrega do trabalho deve ser feita pelo **BlackBoard** (**não serão aceitos trabalhos entregues via e-mail**) e será avaliado de acordo com os seguintes critérios:

- Funcionamento do programa;
- O quão fiel é o programa quanto à descrição do enunciado;
- Indentação, comentários e legibilidade do código;
- Clareza na nomenclatura de variáveis e funções;

Para auxiliar na documentação do código e entendimento do que é um programa com boa legibilidade siga as dicas apresentadas nas páginas abaixo para:

- <http://www.ime.usp.br/~pf/algoritmos/aulas/layout.html>
- <http://www.ime.usp.br/~pf/algoritmos/aulas/docu.html>

Como este trabalho pode ser feito em **grupo**, evidentemente você pode “*discutir*” o problema dado com outros **grupos**, inclusive as “*dicas*” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa.