

Prática I. Parte I: Máquina de Turing

Implemente código Python para executar máquinas de Turing. Para tal deverá implementar as seguintes funções:

1. *updateTape*(tape, pos, wrt)

Dada uma fita, uma posição e um carácter esta função simplesmente devolve a fita com a posição pos substituída pelo carácter wrt

2. *updateMachine*(machine, tape, mv, newstate)

Esta função actualiza a posição da máquina usando o código no argumento mv ('L' para esquerda, 'R' para direita)

3. *executeStep*(machine,tape,program)

Esta é a função mais importante. Primeiro deve usar a posição da máquina para ler o correspondente símbolo na fita. Este símbolo juntamente com o estado interno da máquina são os argumentos que devemos usar para buscar a acção prescrita pelo programa para esta condição. Uma vez encontrada a acção, actualizamos a máquina e a fita.

4. *runMachine*(machine,tape,program)

Esta função chama a função (3) até encontrar a condição de paragem (HALT)

A máquina é representada por exemplo por uma tupla (posição, estado)

A fita é uma lista de caracteres por exemplo:

```
[' ', ' ', ' ', 'x', 'x', 'y', 'z', 'z', ' ', ' ', ' ', ' ']
```

O programa exemplo:

```
[ ['0', ' ', ' ', ' ', 'L', '5'],  
  ['0', 'x', ' ', ' ', 'L', '1'],  
  ['0', 'y', ' ', ' ', 'L', '2'],  
  ['0', 'z', ' ', ' ', 'L', '3'],  
  ['1', ' ', 'x', 'R', '4'],  
  ['2', ' ', 'y', 'R', '4'],  
  ['3', ' ', 'z', 'R', '4'],  
  ['4', ' ', ' ', 'R', '0'],  
  ['5', ' ', ' ', 'L', '6'],  
  ['6', ' ', ' ', 'R', 'HALT '],  
  ['6', 'x', 'x', 'L', '6'],  
  ['6', 'y', 'y', 'L', '6'],  
  ['6', 'z', 'z', 'L', '6']]
```