Trabalho Final - Teoria da Computação

Trabalho Final referente a disciplina de Teoria da Computação.

Universidade Federal de Lavras.

Autores

- **Luis Ferreira** - luis131313
- **Matheus Silva** - matheusGonks

Link para o GitHub

Respostas dos exercícios estão em um arquivo localizado no GitHub.

Exemplo de entrada:

Numero 1: 111

Numero 2: 011

In [ ]:

```
class MT:
    #dicionario funções de transição : estado atual -> transição correspondente
    funcoes_transicao = {("q0", "B") : ("q1","B","D") ,
                         ("q1","1") : ("q1","1","D"),
                         ("q1","0") : ("q1","0","D"),
                         ("q1","B") : ("q2","B","D"),

                         ("q2","1") : ("q2","1","D"),
                         ("q2","0") : ("q2","0","D"),
                         ("q2","B") : ("q3","B","E"),

                         ("q3","1") : ("q4","0","E"),
                         ("q3","0") : ("q3","1","E"),
                         ("q3","B") : ("q6","B","D"),

                         ("q4","1") : ("q4","1","E"),
                         ("q4","0") : ("q4","0","E"),
                         ("q4","B") : ("q5","B","E"),

                         ("q5","1") : ("q5","0","E"),
                         ("q5","0") : ("q1","1","D"),

                         ("q5","B") : ("q1","1","D"),

                         ("q6","1") : ("q6","B","D"),
                         ("q6","B") : ("qf","B","D")
    }

def __init__(self):
    self.__fita = ['B']
    self.__estados = ["q0", "q1", "q2", "q3", "q4", "q5", "q6"]
    self.__alfabetoEntrada = ['0', '1']
    self.__alfabetoFita = ['0' , '1' , 'B']
    self.__cabeca = ['q0', 'B'] #estado atual da maquina
    self.__indice = 0 #podição atual da cabeça de gravação


def executa(self): #inicia execucao da soma
    while(self.__cabeca[0] != "qf"): #enquanto o estado atual da maquina nao for o final
        estadoAtual = self.__cabeca #pego o estado atual dela
        valorCorrespondente = MT.funcoes_transicao.get(tuple(estadoAtual)) #pego a transição correspondente
        self.transicao(valorCorrespondente) #realizo a transição de fato, passando como argumento a tupla c

def transicao(self, proximo):
    (proximoEstado, escrita, direcao) = proximo #proximo = proxEstado da fita - caractere que deve ser escr

    self.__fita[self.__indice] = escrita #escrevero o caractere na posição atual

    if(proximoEstado != "qf"):
        #atualizo o indice/posição atual somente se o proximo estado não for final,
        # caso contrário atualizar o indice fará com que ele esteja fora dos limites da fita
        if(self.__indice == 0):
            self.__fita.insert(0, 'B')
            self.__indice += 1


        if(direcao == 'D'):
            self.__indice += 1
        else:
            self.__indice -= 1

        proxCaractere = self.__fita[self.__indice]
        #depois do indice atual ser atualizado, eu pego o caractere apontado por esse indice

        self.__cabeca = [proximoEstado, proxCaractere]
        #atualizo a o estado atual da máquina


    def recebe_Entrada(self, numero1, numero2):

        for x in list(numero1):
            self.__fita.append(x)

        self.__fita.append('B')

        for x in list(numero2):
            self.__fita.append(x)

        self.__fita.append('B')

    def devolveFita(self):
        while('B' in self.__fita):
            self.__fita.remove('B')

        fita = ''.join(self.__fita)
        print(fita)

minhaMaquina = MT()

primeiroNumero = input("Insira o primeiro numero: ")
segundoNumero = input("Insira o segundo numero: ")

minhaMaquina.recebe_Entrada(primeiroNumero, segundoNumero)
minhaMaquina.executa()
minhaMaquina.devolveFita()
```

In [ ]:

In [ ]:

In [ ]: