

Máquinas de Turing

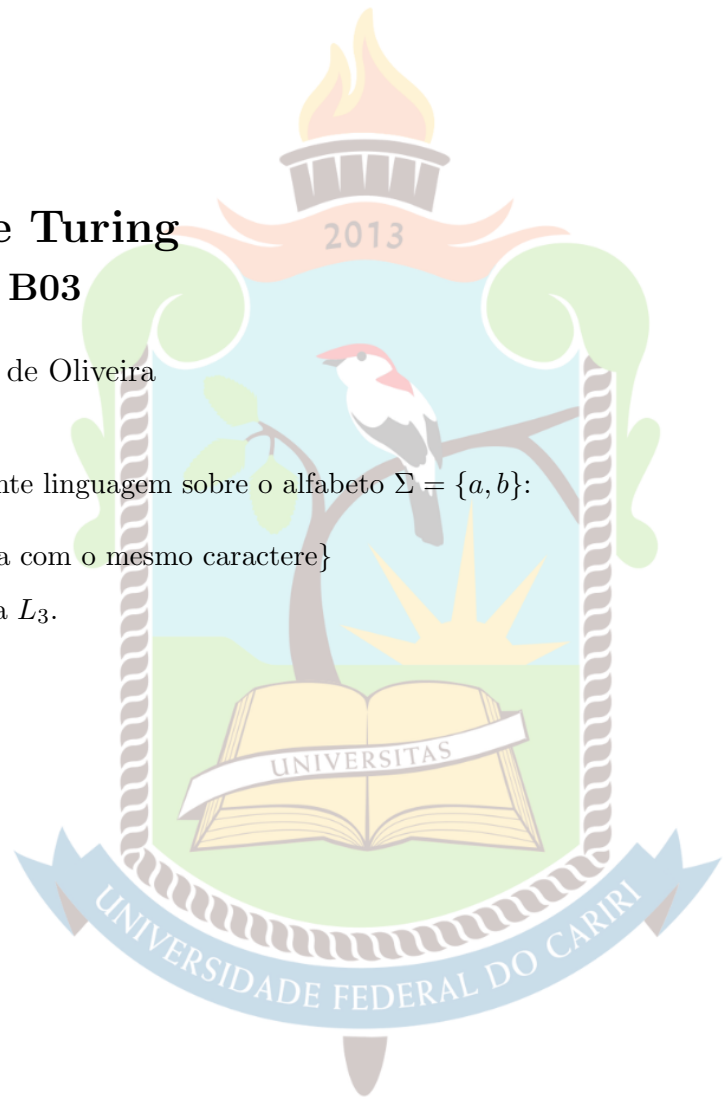
Exercício B03

Autor: Tony Esaú de Oliveira

Descrição do Problema: considere a seguinte linguagem sobre o alfabeto $\Sigma = \{a, b\}$:

$$L_3 = \{w \in \Sigma^* | w \text{ começa e termina com o mesmo caractere}\}$$

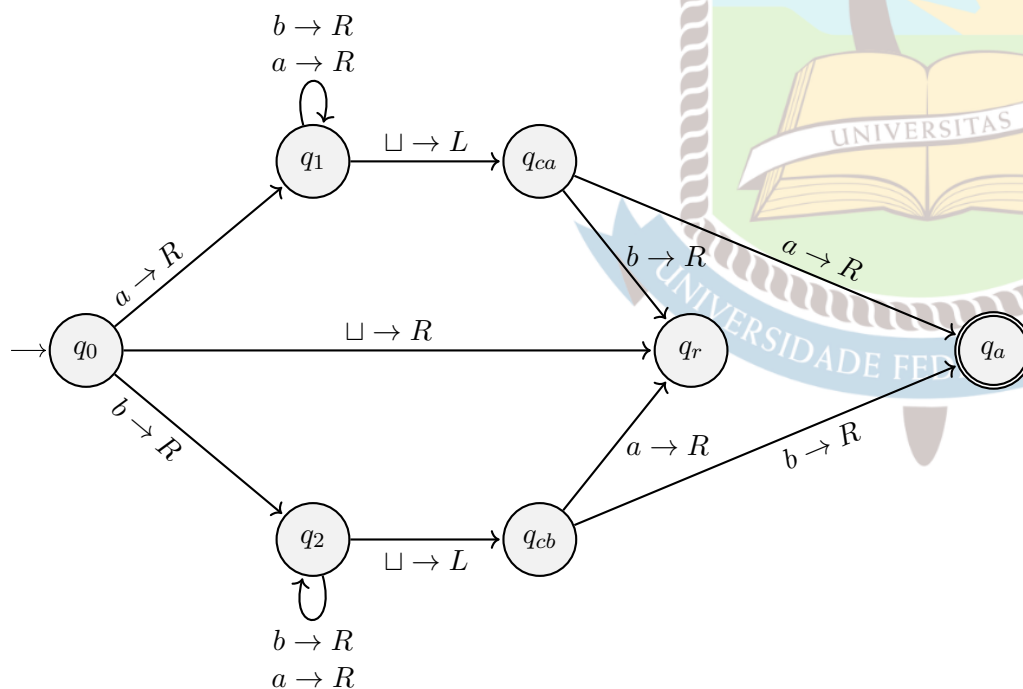
Implemente uma Máquina de Turing que decida L_3 .



Obs.: na próxima página haverá a resolução do problema. Por isso, antes de ver, tente implementar a máquina por si próprio.

Ideia de Resolução: inicialmente, se a cadeia for vazia, rejeita, pois a cadeia vazia não possui caracteres. O estado inicial q_0 bifurca entre dois estados. Esses dois estados são responsáveis por guardar a informação de qual foi o caractere lido inicialmente, q_1 para a e q_2 para b. Em seguida, a fita toda é percorrida com os laços desses estados, ignorando a e b até que se chegue ao fim da fita. Quando isso acontece, a cabeça de leitura volta uma posição e pausa sobre o último símbolo lido, enquanto a máquina avança para um dos estados de comparação. Os estados de comparação, c_a e c_b , são responsáveis por comparar a próxima transição com o caractere que estava sendo visto pelo estado anterior. Por exemplo, se o estado c_a foi atingido na computação, o primeiro símbolo lido foi a, então nesse estado uma transição lendo a deve aceitar a cadeia, e rejeitar caso contrário, o mesmo acontece para c_b quando a cadeia começa com b.

Diagrama Formal:



Código da máquina:

```

input: 'abbbaabaabab'
blank: ' '
start state: q_0

table:

# Estado inicial bifurca nas duas possibilidades de início de cadeia.
# Cadeia vazia não possui caracteres, rejeita.
q_0:
a: {R: q_1}
b: {R: q_2}
' ': {R: q_rejeita}

# Estado de 'memória' para a, se bate no fim da fita volta ao último caractere para

```

```
fazer comparação.  
q_1:  
a: {R: q_1}  
b: {R: q_1}  
' ': {L: q_comp_0}
```

```
# Estado de 'memória' para b, se bate no fim da fita volta ao último caractere para  
fazer comparação.
```

```
q_2:  
a: {R: q_2}  
b: {R: q_2}  
' ': {L: q_comp_1}
```

```
# Compara leitura do último caractere com a, se for a aceita, caso contrário, rejeita.
```

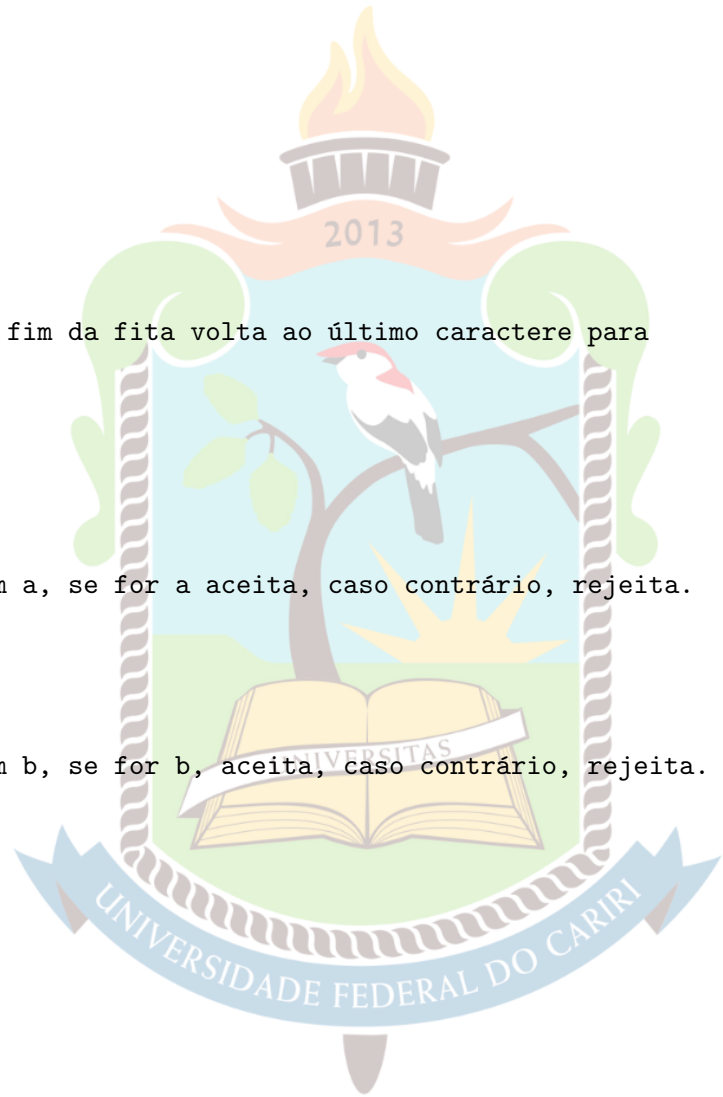
```
q_comp_0:  
a: {R: q_aceita}  
b: {R: q_rejeita}
```

```
# Compara leitura do último caractere com b, se for b, aceita, caso contrário, rejeita.
```

```
q_comp_1:  
a: {R: q_rejeita}  
b: {R: q_aceita}
```

```
q_aceita:
```

```
q_rejeita:
```



[Acesse o repositório aqui!](#)