

Disciplinas MCTA015-13 / CCM-104

Linguagens Formais e Autômatas / Teoria da Computação

Profa. Carla Negri Lintzmayer

`carla.negri@ufabc.edu.br`

`www.professor.ufabc.edu.br/~carla.negri`

Centro de Matemática, Computação e Cognição – Universidade Federal do ABC



Estes slides não contêm um conteúdo completo sobre AFDs: são apenas um apoio gráfico a uma aula específica dada em sala.

Uma possível forma

Simulando um AFD em Python

```
1 def AFD(M, cadeia):
2     (Q, Sigma, delta, q0, F) = M
3     qA = q0
4     for x in cadeia:
5         qA = delta[(qA, x)]
6     return qA in F
```

Simulando um AFD em Python

```
1 def AFD(M, cadeia):
2     (Q, Sigma, delta, q0, F) = M
3     qA = q0
4     for x in cadeia:
5         qA = delta[(qA, x)]
6     return qA in F
```

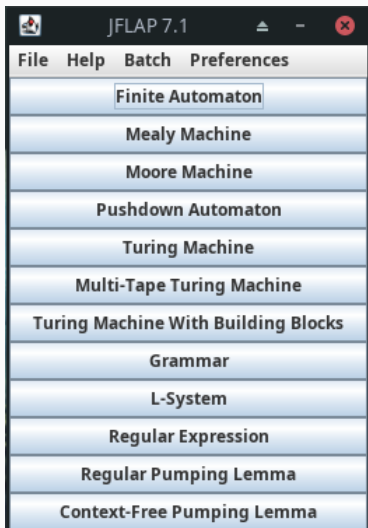
```
1 delta = { ('q0','0'): 'q0', ('q0','1'): 'q1', ('q1','0'): 'q2',
2           ↪ ('q1','1'): 'q1', ('q2','0'): 'q1', ('q2','1'): 'q1' }
3 Q = ['q0', 'q1', 'q2']
4 F = ['q1']
5 Sigma = ['0', '1']
```

Simulando um AFD em Python

```
1 AFD((Q, Sigma, delta, 'q0', F), '0001010101000')
2 False
3
4 AFD((Q, Sigma, delta, 'q0', F), '000101010100')
5 True
6
7 AFD((Q, Sigma, delta, 'q0', F), '00010101010')
8 False
9
10 AFD((Q, Sigma, delta, 'q0', F), '0001010101')
11 True
12
13 AFD((Q, Sigma, delta, 'q0', F), '000101010')
14 False
```

Outras formas

JFLAP: <https://www.jflap.org/>



Programas prontos

JFLAP: <https://www.jflap.org/>

