

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 AFNs: são apenas um apoio gráfico a uma aula específica dada em sala.

Uma possível forma

Simulando um AFN em Python

```
1 def AFN(N, cadeia):
2     (Q, Sigma, delta, q0, F) = N
3     QA = E({q0}, delta)
4     for x in cadeia:
5         novos = set()
6         for q in QA:
7             if (q, x) in delta:
8                 novos.update( E(delta[(q,x)], delta) )
9         QA = novos
10    return len( QA.intersection(F) ) != 0
```

Simulando um AFN em Python

```
1 def E(estados, delta):
2     S = set(estados)
3     nao_explorados = list(estados)
4     while len(nao_explorados) > 0:
5         q = nao_explorados.pop()
6         if (q, 'epsilon') in delta:
7             novos = delta[ (q, 'epsilon') ].difference(S)
8             S.update(novos)
9             nao_explorados.extend(novos)
10    return S
```

Simulando um AFN em Python

```
1 delta = { ('q1', '0'): {'q1'}, ('q1', '1'): {'q1', 'q2'},  
  ↪ ('q2', '0'): {'q3'}, ('q2', '1'): {'q3'}, ('q3', '0'): {'q4'},  
  ↪ ('q3', '1'): {'q4'} }  
2 Q = ['q1', 'q2', 'q3', 'q4']  
3 F = ['q4']  
4 Sigma = ['0', '1']
```

Simulando um AFN em Python

```
1 AFN((Q, Sigma, delta, 'q1', F), '0001010101000')
2 False
3
4 AFN((Q, Sigma, delta, 'q1', F), '000101010100')
5 True
6
7 AFN((Q, Sigma, delta, 'q1', F), '00010101010')
8 False
9
10 AFN((Q, Sigma, delta, 'q1', F), '')
11 False
12
13 AFN((Q, Sigma, delta, 'q1', F), '000000')
14 False
```

Outras formas

Programas prontos

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

