

# Linguagens Formais

UNICAP

Eduardo Araújo Oliveira  
<http://sites.google.com/site/eaoufpe>



## Estrutura

- Definição de AFND-e
- Equivalência  $AF\epsilon$  / AFN
- JFLAP

## AF $\epsilon$

slide 3

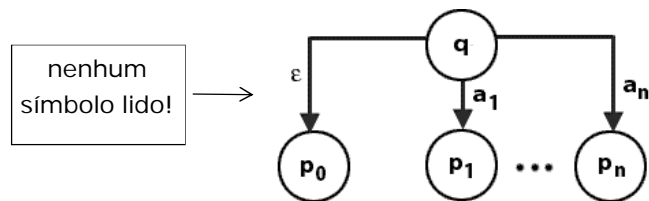
## AFND $\epsilon$

- Uma extensão do formalismo AFND
  - A diferença é que permite movimentos vazios
- Movimento vazio (transição  $\epsilon$ )
  - Uma transição sem leitura de símbolo
  - Transição não obrigatória
  - A fita não se altera

4

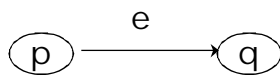
## AFND $\epsilon$

- A diferença para os AFNDs é a função de transição
  - Além dos símbolos, agora também está definida para  $\epsilon$  (ausência de símbolo)



5

## AFND $\epsilon$



o autômato vai do estado **p** para o estado **q** sem ler um símbolo de entrada.

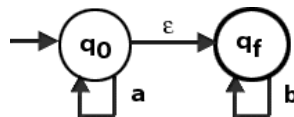
slide 6

## AFND $\epsilon$

- Exemplo
  - ACEITA(M) = {  $w$  | todo  $a$  antecede todo  $b$  }

$M$ :

$\delta$ :

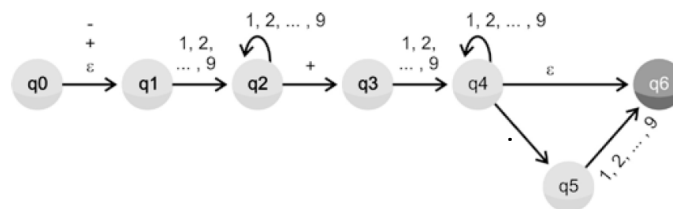


	$a$	$b$	$\epsilon$
$q0$	{ $q0$ }	{ }	{ $qf$ }
$qf$	{ }	{ $qf$ }	{ }

7

## AFND $\epsilon$

Suponha um autômato que reconhece a soma de um inteiro positivo ou negativo com um decimal positivo.

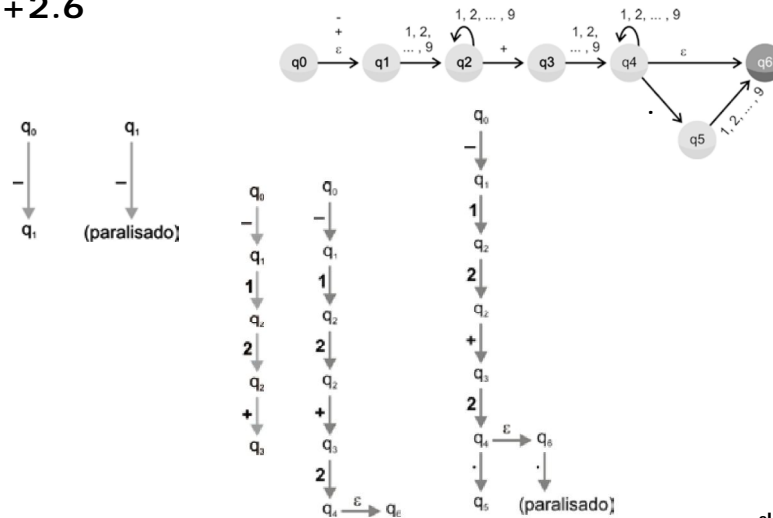


slide 8

## AFND $\epsilon$

Acompanhar este autômato com a expressão

**-12+2.6**



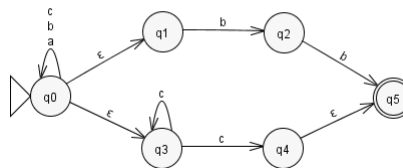
slide 9

## AFND $\epsilon$

Mostre como o autômato finito não-determinístico com transições vazias (AFND- $\epsilon$ ) se comporta ao receber a palavra **abc**.

Para isso, mostre os conjuntos de estados atingidos após a leitura de cada símbolo da palavra. Lembre-se de considerar as transições  $\epsilon$  antes e depois de fazer a transição para os símbolos da palavra.

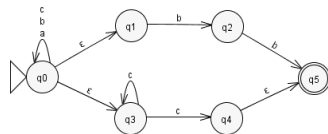
autômato D:



slide 10

## AFND $\epsilon$

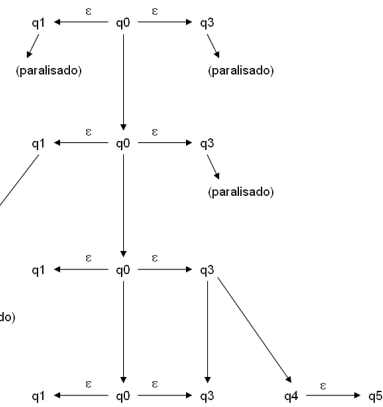
autômato D:



a

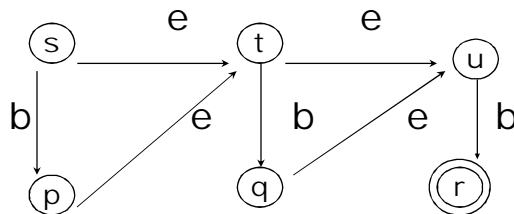
b

c



slide 11

## AFND $\epsilon$



Estando no estado  $s$  e recebendo o símbolo  $b$ :

- ler  **$b$**  e ir para  **$p$**
- ir para  **$t$**  e então ler  **$b$**  e ir para  **$q$**
- ir para  **$t$** , ir para  **$u$**  e então ler  **$b$**  e ir para  **$r$** .

O conjunto  $\{b, bb, bbb\}$  é aceito pelo autômato.

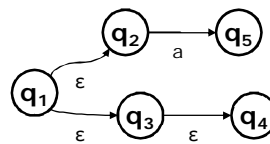
slide 12

## AFND $\epsilon$

- Função FECHO- $\epsilon$ 
  - Estados alcançáveis sem leitura de símbolo, inclui
    - O próprio estado
    - Estados alcançáveis por transições  $\epsilon$

- Exemplo

- FECHO- $\epsilon(q_1)$   
= {  $q_1, q_2, q_3, q_4$  }



13

## AF $\epsilon$ vs. AFN

- Apesar do novo tipo de transição, o AF $\epsilon$  não reconhece linguagens mais complexas
- Equivalência AFN  $\rightarrow$  AF $\epsilon$ 
  - Trivial, decorrente da definição
- Equivalência AF $\epsilon \rightarrow$  AFN
  - Dado um AF $\epsilon$  qualquer  $M$ , construir um AFN equivalente  $M'$

14

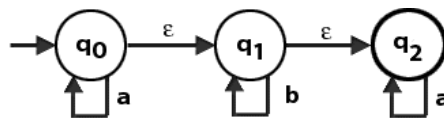
## Equivalência $AF\epsilon$ / AFN

- Construindo  $M'$ 
  - Muda apenas  $\delta'$  e  $F'$  !
  - $\delta'(q, a) = \underline{\delta}(\{q\}, a)$ 
    - Para todo estado  $q$  todo símbolo  $a$
  - $F' = \{q \mid \text{FECHO-}\epsilon(q) \text{ contém estado de } F\}$   
 $= \{q \mid \text{FECHO-}\epsilon(q) \cap F \neq \emptyset\}$

15

## Equivalência $AF\epsilon$ / AFN

- Exemplo 1
  - Criar AFN equivalente ao  $AF\epsilon$

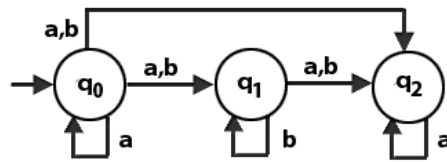


16



## Equivalência $AF\epsilon$ / AFN

- Exemplo 2
  - Resultado:



17

## $AF\epsilon$

- Portanto, o formalismo  $AF\epsilon$  não é mais expressivo do que o AFN nem do que o AFD
  - São formalismos equivalentes
- Os três reconhecem a mesma classe de linguagens
  - Linguagens Regulares (ou Tipo-3)

18

## JFLAP

slide 19

### JFLAP

- Simulador de formalismos, desenvolvido em Java, disponível no site da disciplina
- Permite criar um autômato qualquer
  - Opção “Finite Automaton”
- Trata todos os autômatos como AFE

20

## JFLAP

- Editor
  - Permite adicionar estados e transições
  - Ao adicionar a transição, clique no campo e digite um caractere
    - Se não digitar nada, a aresta será  $\epsilon$  (ou  $\lambda$ )
  - Clique com o botão direito no estado e escolha se é inicial/final

21

## JFLAP

- Caminho de reconhecimento de uma palavra
  - Input->Fast Run
- Converter de  $AF\epsilon$  para um AFD
  - Convert->Convert to DFA
    - DFA – Deterministic Finite Automaton

22

## Referências

- Dois primeiros capítulos do livro em formato eletrônico
  - <http://teia.inf.ufrgs.br/library.html>

23

## Linguagens Formais

UNICAP

Eduardo Araújo Oliveira  
<http://sites.google.com/site/eaoufpe>