Linguagens Formais

UNICAP

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

UNIVERSIDADE CATÓLICA DE PERNAMBUCO

Estrutura da Apresentação

1. Linguagens Regulares
2. Definição
   – Casos base
   – Operadores
3. Equivalência AFε / ER

Linguagens Regulares

- Todos os formalismos *reconhecedores* foram vistos
  - Autômatos Finitos Determinísticos
  - Autômatos Finitos Não-determinísticos
  - Autômatos Finitos com Movimentos ε

- Nesta aula veremos um formalismo *denotacional* ou *gerador*
  - Expressões Regulares

3

Expressões Regulares

As expressões regulares são utilizadas principalmente como descritores de linguagens, ou seja, a partir destas expressões podemos identificar uma linguagem regular e dada uma linguagem podemos escrevê-la de forma simplificada usando expressões (se a linguagem for regular).

slide 4

Expressões Regulares

Dada uma expressão regular **r** qualquer, a Linguagem que ela representa é referenciada por **L(r)**, que você pode ler como “a linguagem de r”.

slide 5

Expressões Regulares

- aceitar ou rejeitar uma palavra? Não mais!
- Utilização
  - localizar cadeias em um texto
  - para criar analisadores léxicos, que são componentes fundamentais dos compiladores
  - Etc...

slide 6

Expressões Regulares

- Padrão que indica o formato geral das palavras de uma linguagem
- Dá uma idéia de como todas as palavras podem ser geradas
- Mais intuitivo do que autômatos

7

Expressões Regulares

- Assim como uma expressão aritmética representa um número natural:

$$(10 + 5) \times 7$$

- Uma ***expressão regular*** representa uma linguagem:

$$(\mathbf{0} + \mathbf{1}) \cdot \mathbf{0}^*$$

slide 8

Expressões Regulares

- Exemplo
  - linguagem regular: o conjunto de cadeias de 0’s e 1’s tais que comece com qualquer quantidade de 1’s (inclusive nenhum), seguidos necessariamente de um 0 e outra seqüencia com qualquer quantidade de 1’s
    - Essa linguagem aparentemente complexa pode ser escrita em forma de expressão regular facilmente: **1*01***

slide 9

Expressões Regulares

- T = {c, d}
- L = {palavra que tem “cc” como subpalavra}

(c+d)(cc)(c+d)**

10

Expressões Regulares

Definição formal: Seja Σ um alfabeto

1. Se $a \in \Sigma$, então **a** é uma expressão regular.
2. Se λ é a palavra nula, então $\boldsymbol{\lambda}$ é uma expressão regular.
3. Se $\varnothing$ é o conjunto vazio, então $\varnothing$ é uma expressão regular.
4. Se R_1 e R_2 são expressões regulares, então $(R_1 + R_2)$ e $(R_1 \bullet R_2)$ são expressões regulares.
5. Se R_1 é uma expressão regular, então $(R_1{}^*)$ é uma expressão regular.

slide 11

Expressões Regulares

São três os tipos de expressões regulares básicas:

- **si**, para todo símbolo si do alfabeto, e representa a linguagem { si}, ou seja, a linguagem formada pela palavra de um símbolo que tem apenas um símbolo si. Assim, podemos escrever que L(si) = { si }.

- ε, que representa a linguagem {ε }, ou seja, a linguagem que contém apenas a palavra vazia. Assim, podemos escrever que L(ε) = { ε }.

- **Ø**, que representa a linguagem { }, ou seja, a linguagem que não tem palavra nenhuma. Assim, podemos escrever que L(**Ø**) = { }.

Os dois primeiros tipos de expressões representam uma única palavra e são idênticos a ela. Já o terceiro tipo é uma expressão que não representa palavra nenhuma.

slide 12

Expressões Regulares

- Na expressão (**0** + **1**) • **0*** :
  - **0** representa o conjunto {0}
  - **1** representa o conjunto {1}
  - (**0** + **1**) representa o conjunto $\{0\} \cup \{1\} = \{0, 1\}$
  - **0*** representa $\{0\}^*$
- Então (**0** + **1**) • **0*** representa a linguagem:

$$\{uv: u \in \{0, 1\} \text{ e } v = 0^n, n \geq 0\}$$

slide 13

Expressões Regulares

Definição:

Se R é uma expressão regular, então L(R) é a linguagem que R representa/descreve.

Ex.: L(**a** + **b**) = {a, b}

slide 14

Expressões Regulares

OPERADORES

slide 15

Expressões Regulares

- **UNIÃO**
  - L = {001, 110} e M= {e, 11, 110}
  - L ∪ M = {001, 110, e, 11}
  - L(**E+F**) = L(**E**) ∪ L(**F**)

- Operador '**+**' e a idéia de '**ou**'

A expressão básica **a** representa a linguagem {a} e a expressão básica **b** representa a linguagem {b}. Logo, a expressão **a+b** representa a linguagem {a, b}.

slide 16

Expressões Regulares

- **CONCATENAÇÃO**
  - L = {001, 110} e M = {e, 11, 110}
  - L.M (com um ponto) ou LM (sem ponto), onde LM = {001, 00111, 001110, 110, 11011, 110110}
  - **L(E.F) = L(E).L(F)**

- **Atenção: LM ≠ ML**

- Operador '**.**'

A expressão regular **a.b** ou **ab** representa que linguagem? As expressões básicas **a** e **b** representam {a} e {b}, logo concluímos que a expressão dada representa {ab}.

slide 17

Expressões Regulares

Quais palavras a expressão **(a+b)c** representa?

Na expressão dada, temos uma união de **a** e **b**, que representa {a,b}. Em seguida, concatenada a **(a+b)** temos a expressão **c** que representa {c}. O resultado da concatenação {a,b}.{c} dá a linguagem {ac, bc}, que é a resposta esperada!

slide 18

Expressões Regulares

- **FECHAMENTO DE KLEENE ou ESTRELA**
  - L ={00, 11}
  - L* = {e, 0011, 001100, 11110011, ...}.
  - L(**E***) = (L(**E**))*

  - A idéia é de que, para formar o fechamento Kleene L*, podemos usar:
  - nenhuma cadeia de L, ou seja, {e};
  - ou cadeias individuais de L, que dá o próprio conjunto L;
  - ou cadeias de L concatenadas aos pares, ou seja, L.L;
  - ou cadeias de L concatenadas de três em três, ou seja, L.L.L;
  - etc.

slide 19

Expressões Regulares

- **FECHAMENTO DE KLEENE ou ESTRELA**
  - Exemplo
  - E a expressão **(0+01)***, que linguagem representa? Veja que a expressão interna **0+01** representa {0, 01}.
    - linguagem M = {0, 01}
    - M0 = {e}
    - M1 = {0, 01}
    - M2 = {00, 0101, 001, 010}, que é formado pela concatenação de pares de cadeias de L
    - M3 = {000, 0001, 01010, 010101, 0010, 00101, 0100, 01001}}, que é formado pela concatenação de cadeias de L

L* = $L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$

slide 20

Expressões Regulares

- Operadores (seja *r* uma expressão)
  – Concatenação sucessiva: ***r****
    - Dá a idéia de zero ou mais repetições de *r*

  – Denota $L = L_r^*$
  $= \{$palavras formadas pela concatenação de zero ou mais palavras de $L_r\}$

  – Exemplo
  a* denota L = { ε, a, aa, aaa, aaaa, ... }
  ab* denota L = { a, ab, abb, abbb, ... }

21

Linguagem Gerada

- Seja *r* uma ER, então ao conjunto de palavras denotado por *r* dá-se o nome de "linguagem gerada" por *r*
  – GERA(r) ou L(r)

- A mesma idéia da "linguagem aceita" nos autômatos
  – ACEITA(M)

22

Exemplos

- Todas as palavras sobre T = {a, b}

  *(a + b)**

- Palavras que terminam com *aa* ou *bb*

  (a+b)(aa+bb)*

23

Expressões Regulares

Precedência de operadores

*

.

\+

slide 24

Expressões Regulares

Precedência de operadores

110* **=** {11, 110, 1100, 11000, ...} = 11(0*)

00*0+011 = (0(0*)0)+(011)

slide 25

Expressões Regulares

- Pratica
  - O conjunto de todas as cadeias de 0's e 1's com exatamente três símbolos
    - **(0+1) (0+1) (0+1)**
  - O conjunto de cadeias de 0's e 1's contendo pelo menos um símbolo 0
    - **0(0+1)* + (0+1)*0 + (0+1)*0(0+1)***
  - Forneça uma descrição em português da expressão: (0+1)*101(0+1)*
    - **o conjunto de todas as cadeias de zeros e uns que contém 101 como subcadeia**

slide 26

Expressões x Linguagens

Alfabeto = {a,b}

Expressão Regular	Linguagem Representada
aa	somente a palavra aa
ba*	todas as palavras que iniciam por b, seguido por zero, um ou mais a
(a + b)*aa(a + b)*	todas as palavras contendo aa como subpalavra
a*ba*ba*	todas as palavras contendo exatamente dois b's
(a + b)*(aa + bb)	todas as palavras que terminam com aa ou bb

slide 27

Expressões x Linguagens

Alfabeto 0, 1

Expressão Regular	Linguagem Representada
(0+1)*(00+01+11)	Todas as cadeias não terminadas em 10
(0+1)*11	Todas as cadeias terminadas em 11
(0+1)(0+1)1(0+1)*	Todas as cadeias cujo terceiro símbolo é 1
0*1*	Todas as cadeias que possuem uma quantidade qualquer de 0s, e depois uma quantidade qualquer de 1s

slide 28

Autômatos Finitos e Expressões Regulares

- Provaremos a equivalência entre ER e AF da seguinte forma: primeiro mostraremos como obter um AF a partir de uma ER e em seguida mostraremos como obter uma ER a partir de um AF.

slide 29

Autômatos Finitos e Expressões Regulares

Propriedades de Linguagens Regulares

Concatenação de dois conjuntos A e B

$A.B = AB = \{ xy \mid x \in A \text{ e } y \in B\}$

EXEMPLO

$\{a, ab\} \cdot \{b, ba\} = \{ab, aba, abb, abba\}$

Se A e B são conjuntos regulares, AB também é.

Autômatos Finitos e Expressões Regulares

Prova Intuitiva

Seja M o autômato para A e N para B.

Construir um novo autômato P cujo os estados são a união dos de M e N.

Todas as transições de M e N serão transições de P.

O estado inicial de M será o de P.

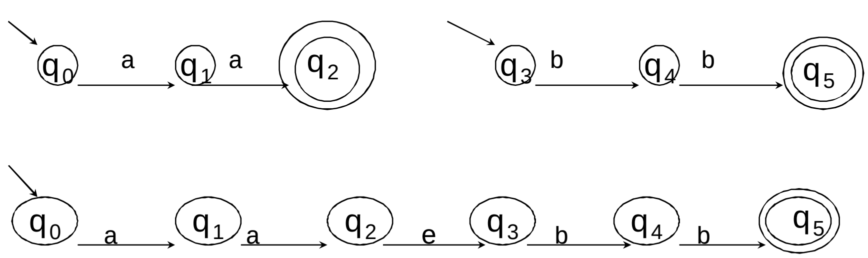
Os estados finais de N serão os de P.

Finalmente, ligue os estados finais de M ao estado inicial de N com uma transição **ε**.

slide 31

Autômatos Finitos e Expressões Regulares

Seja A = {aa}, B = {bb}



Autômatos Finitos e Expressões Regulares

Fecho de Kleene

Se A é regular então A* também é.

$A^* = \{ e\} \cup A \cup A^2 \cup A^3 \cup \ldots = \{ x_1x_2\ldots x_n \mid n >= 0$ e $X_i \in A\}$

Autômatos Finitos e Expressões Regulares

Prova Intuitiva

Seja <u>M</u> o autômato para <u>A</u> então <u>P</u> para <u>A*</u> é como segue:

Comece com todos os estados e transições de <u>M</u>.

Adicione um novo estado ***q*** e uma transição **ε** de ***q*** para o estado inicial de <u>M</u>.

Adicione um novo estado ***qf*** e uma transição **ε** do ultimo estado de M para o estado final qf. Faça *qf* o estado final de <u>P</u>.

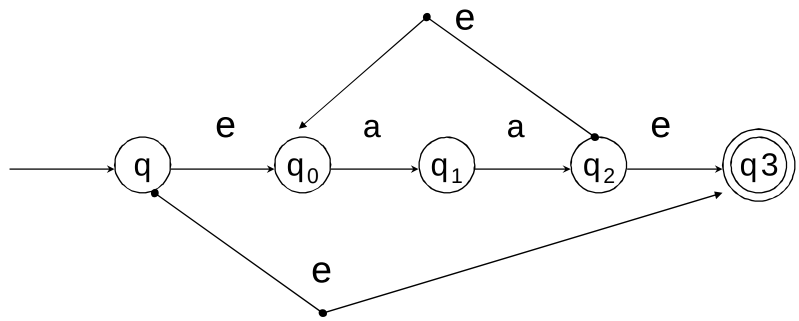
Faça ***qf*** o único estado final de <u>P</u>.

Adicione uma transição **ε** do estado final de <u>M</u> para o estado ***q.***

slide 34

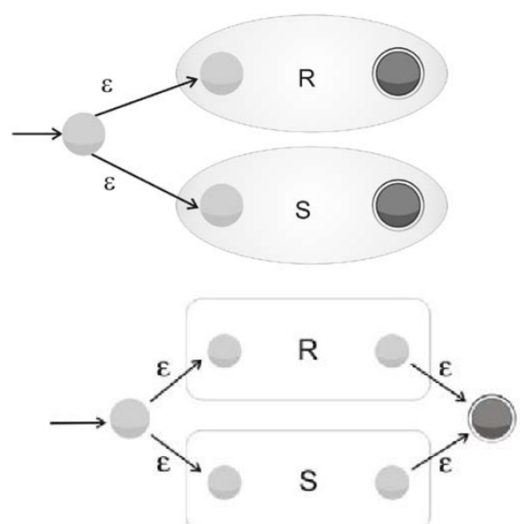
Autômatos Finitos e Expressões Regulares

Dado o autômato para <u>A</u> {aa}, o para A* :



DE Expressões Regulares para Autômatos Finitos

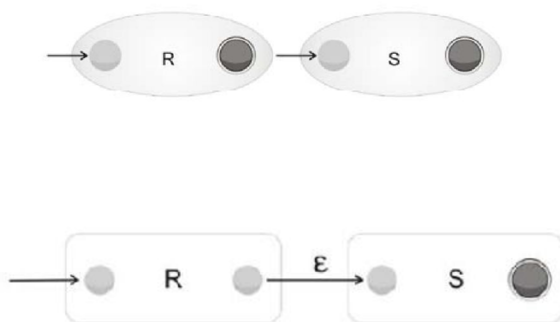
União



slide 36

DE Expressões Regulares para Autômatos Finitos

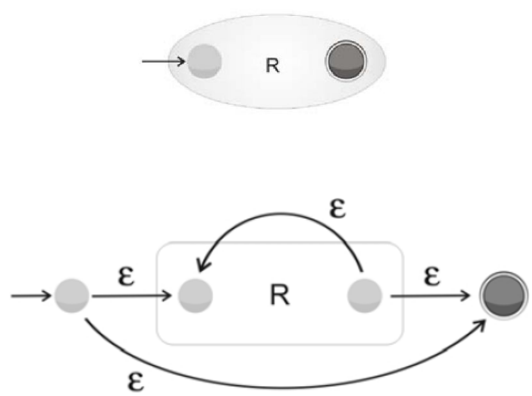
Concatenação



slide 37

DE Expressões Regulares para Autômatos Finitos

Estrela

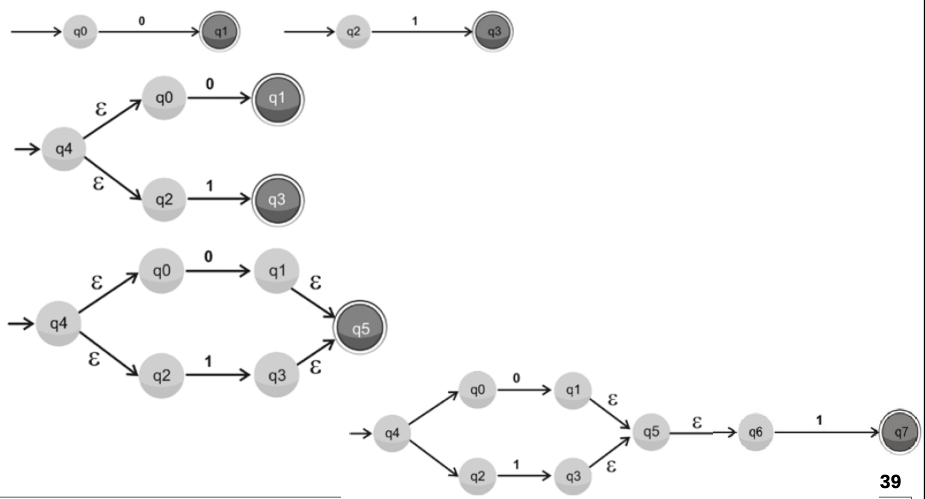


slide 38

Autômatos Finitos e Expressões Regulares

Exemplos:

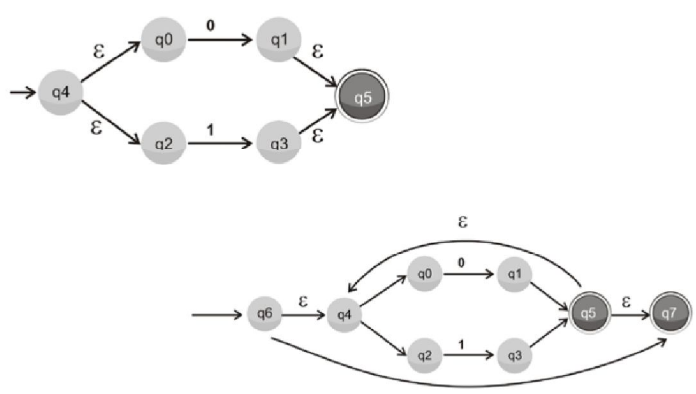
- **(0+1)1**



Autômatos Finitos e Expressões Regulares

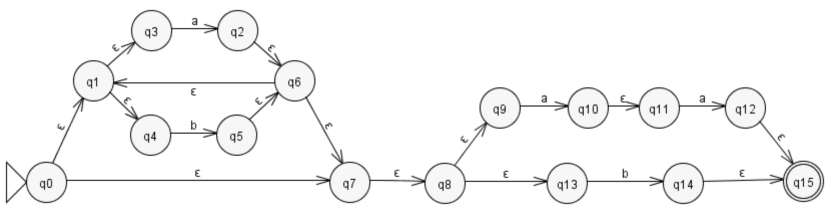
Exemplos:

- **(0+1)***



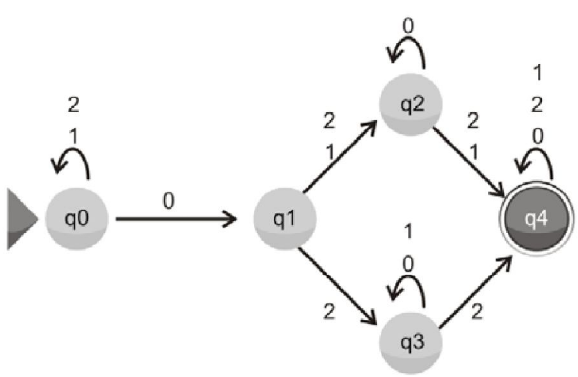
Expressões Regulares

- Pratica
  - Para cada uma das expressões regulares abaixo, criadas sobre o alfabeto {a,b}, crie um autômato finito equivalente:
    - a*
    - (a+b)a*
    - (aa+bb)
    - (a+b)*(aa+b)



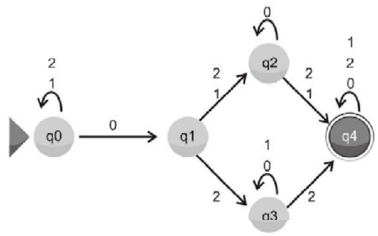
slide 41

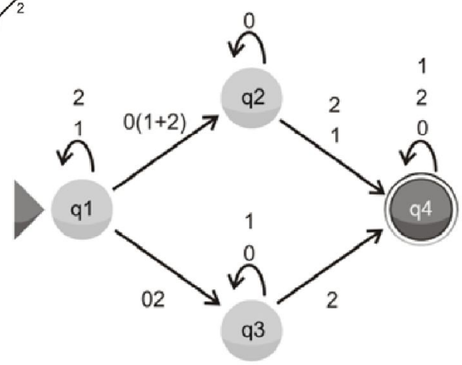
AFD/AFND para Expressões Regulares



slide 42

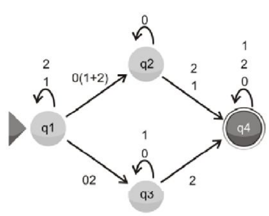
AFD/AFND para Expressões Regulares

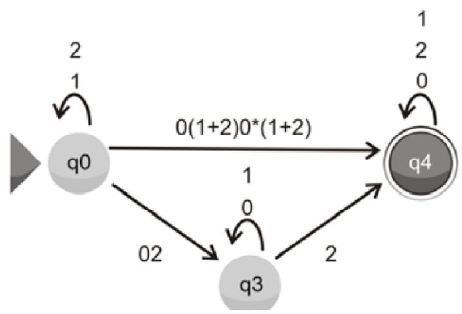




slide 43

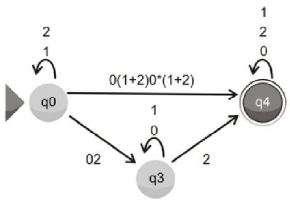
AFD/AFND para Expressões Regulares

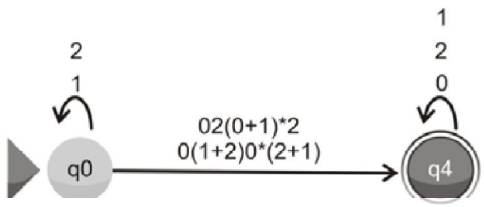




slide 44

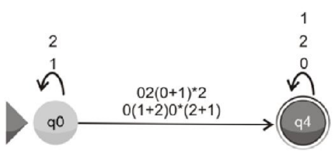
AFD/AFND para Expressões Regulares

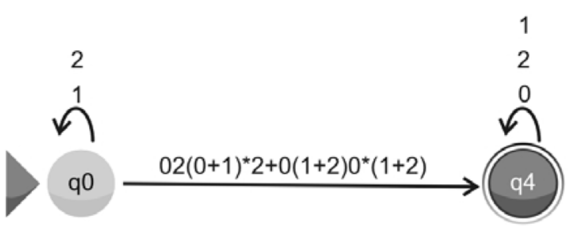




slide 45

AFD/AFND para Expressões Regulares

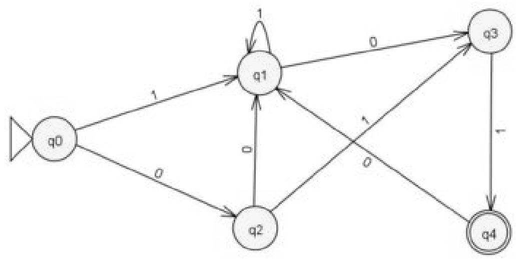




$$\underbrace{(1+2)^*}_{q_0}\ \underbrace{02(0+1)^*2+0(1+2)0^*(1+2)}_{q_0 \text{ a } q_4}\underbrace{(0+1+2)^*}_{q_4}$$

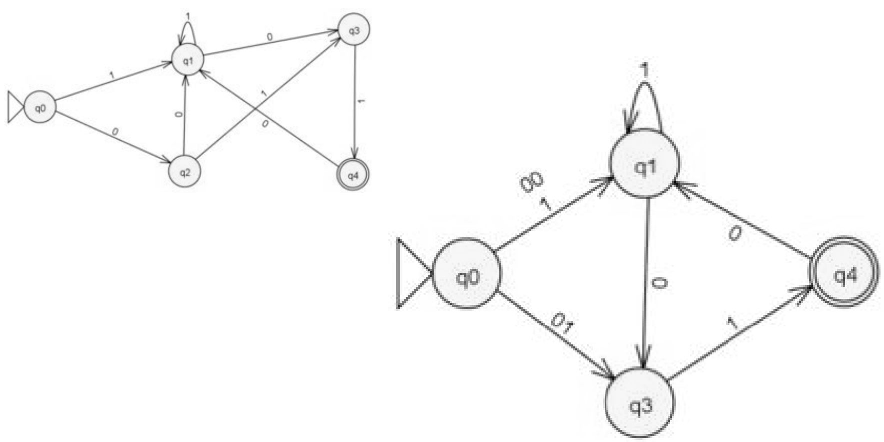
slide 46

AFD/AFND para Expressões Regulares



slide 47

AFD/AFND para Expressões Regulares

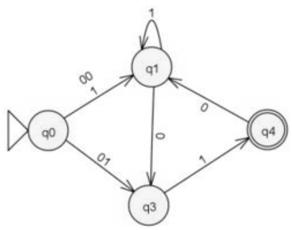


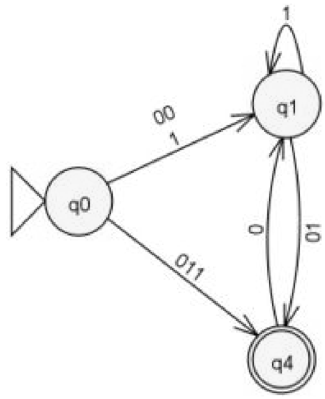
Escolha um dos estados para "colapsar". No nosso exemplo, **q2** foi escolhido.

Verifique quais são os caminhos possíveis passando por aquele nó. Crie novas transições ligando os estados remanescentes como descrito acima. No exemplo, criaremos uma transição ligando **q0** a **q1** com rótulo **00** e de **q0** a **q3** com rótulo **01**.

slide 48

AFD/AFND para Expressões Regulares

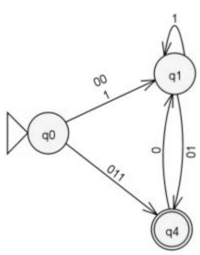


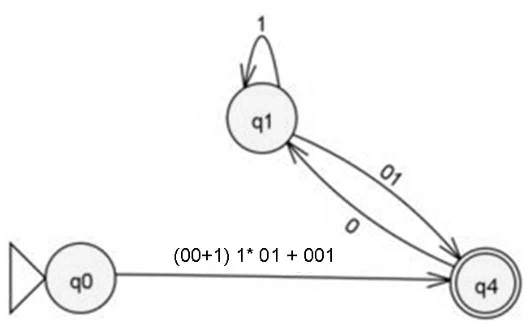


Escolha um novo estado para remover.
Escolha outro estado para remover. No exemplo, removeremos agora **q3**.
Novamente crie novas transições representando os caminhos passando por aquele estado. Foi criada uma transição entre **q0** e **q4** com rótulo **001** e entre **q1** e **q4** com rótulo **01**.

slide 49

AFD/AFND para Expressões Regulares

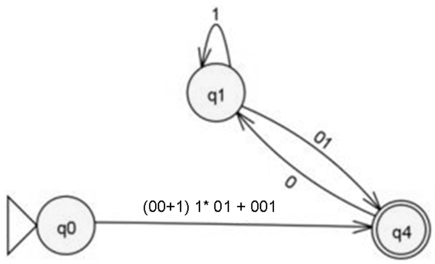


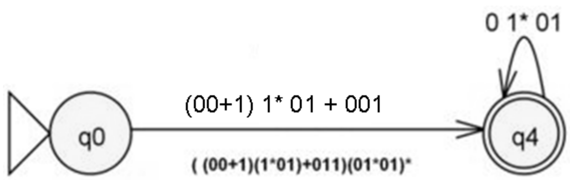


Algumas vezes, você precisa representar caminhos circulares usando a operação de *fecho de Kleene* (* - asterisco), e fazer passos intermediários. Na figura, os caminhos entre **q0** e **q4** que passavam por **q1** foram convertidos para transições

slide 50

AFD/AFND para Expressões Regulares

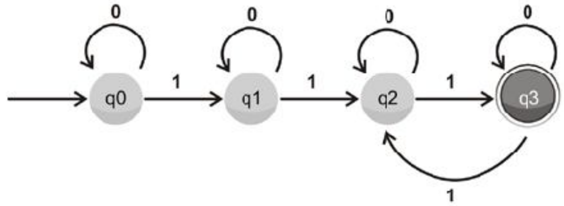




slide 51

AFD/AFND para Expressões Regulares

Exercício prático



slide 52

Equivalência ER / AFε

- As expressões regulares não conseguem representar mais linguagens do que os três tipos de autômatos finitos
  - Todos reconhecem Linguagens Regulares
- Para provar, vamos mostrar como construir um AFε a partir de uma ER
  - Mostraremos conversão para cada caso

53

Aplicações

- Especificar endereços de e-mail válidos
- Procura (avançada) por arquivos
- Para especificar Linguagens de Programação
  - Especificar identificadores
  - Especificar números inteiros
  - Especificar números decimais

54

Linguagens Formais

UNICAP

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

UNIVERSIDADE CATÓLICA DE PERNAMBUCO