Linguagens Formais e Autômatos

Aula 09 - Autômatos finitos e expressões regulares

Prof. Dr. Daniel Lucrédio
Departamento de Computação / UFSCar
Última revisão: ago/2015

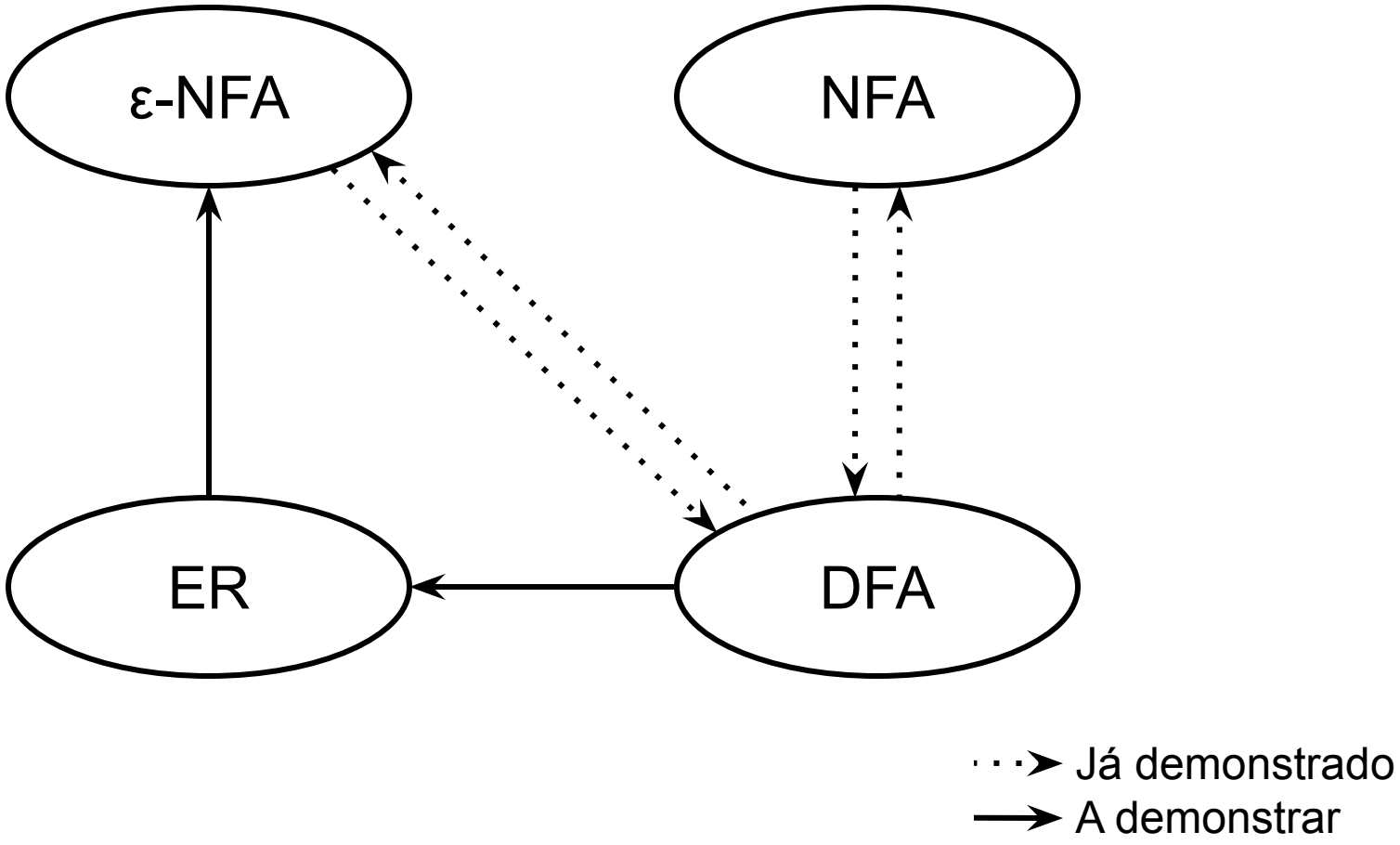
Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
  - Capítulo 3 - Seção 3.2
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
  - Capítulo 1 - Seção 1.3

Autômatos finitos e expressões regulares

- São diferentes na notação
  - Mas tanto autômatos finitos como expressões regulares representam exatamente o mesmo conjunto de linguagens
    - Linguagens regulares
- Ou seja:
  - Toda linguagem definida por um autômato finito também é definida por uma expressão regular
  - Toda linguagem definida por uma expressão regular é definida por um autômato finito

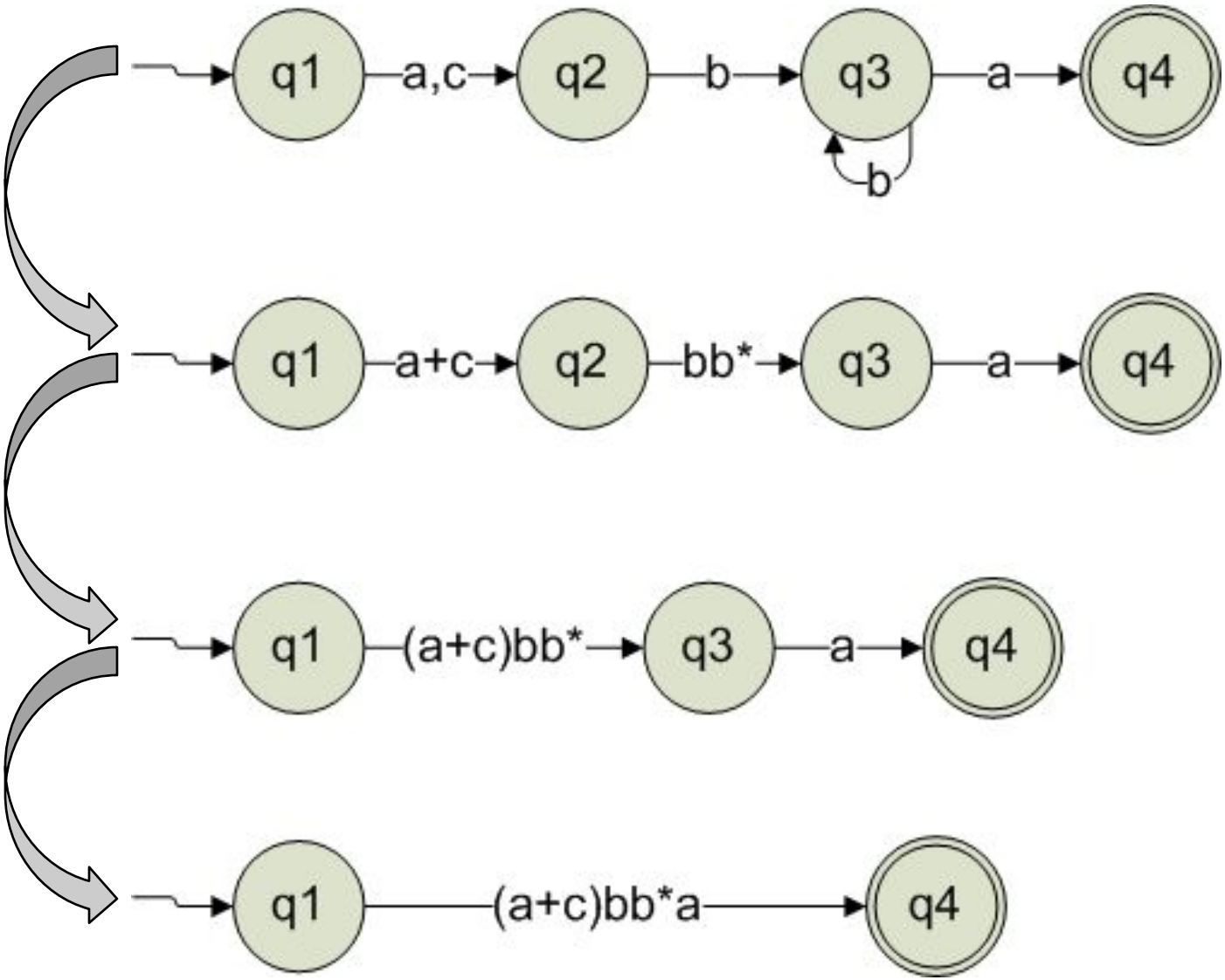
Autômatos finitos e expressões regulares



Conversão DFA→ER

- Teorema: Se L = L(A) para algum DFA A, então existe uma expressão regular R tal que L = L(R)
- Conversão é surpreendentemente complicada
  - Método 1: n^3 expressões, com 4^n símbolos (pior caso)
  - Método 2: eliminação de estados
    - Mais simples, porém também trabalhosa
    - Envolve uma notação mista: autômatos + ERs
      - Autômato finito não-determinístico generalizado
      - Transições são expressões regulares

Autômatos + ERs

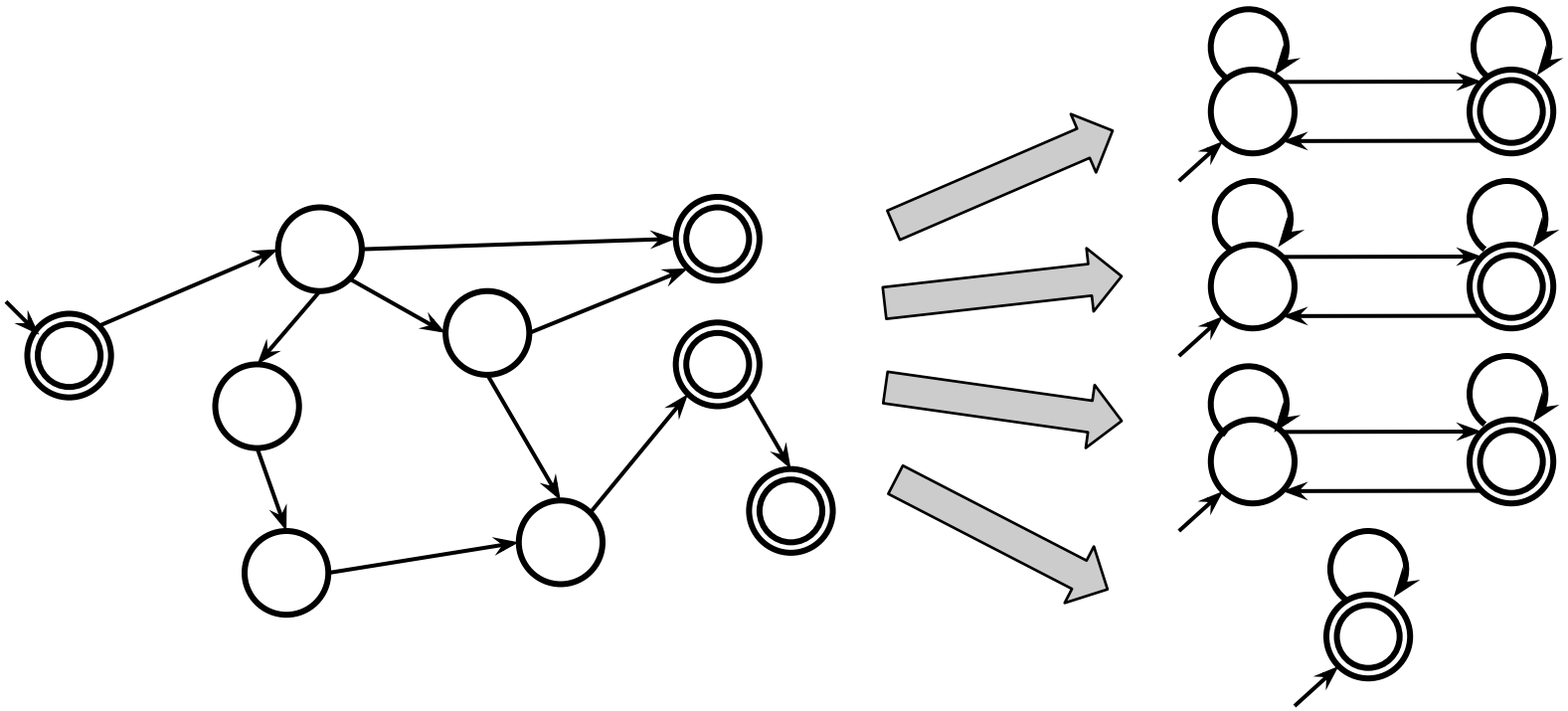


Conversão DFA→ER

- Método da eliminação de estados
- Eliminamos todos os estados, um por um
  - Ao eliminar um estado s, todos os caminhos que passam por s não mais existem no autômato
- Substituiremos símbolos por ER nas transições, para representar as transições eliminadas

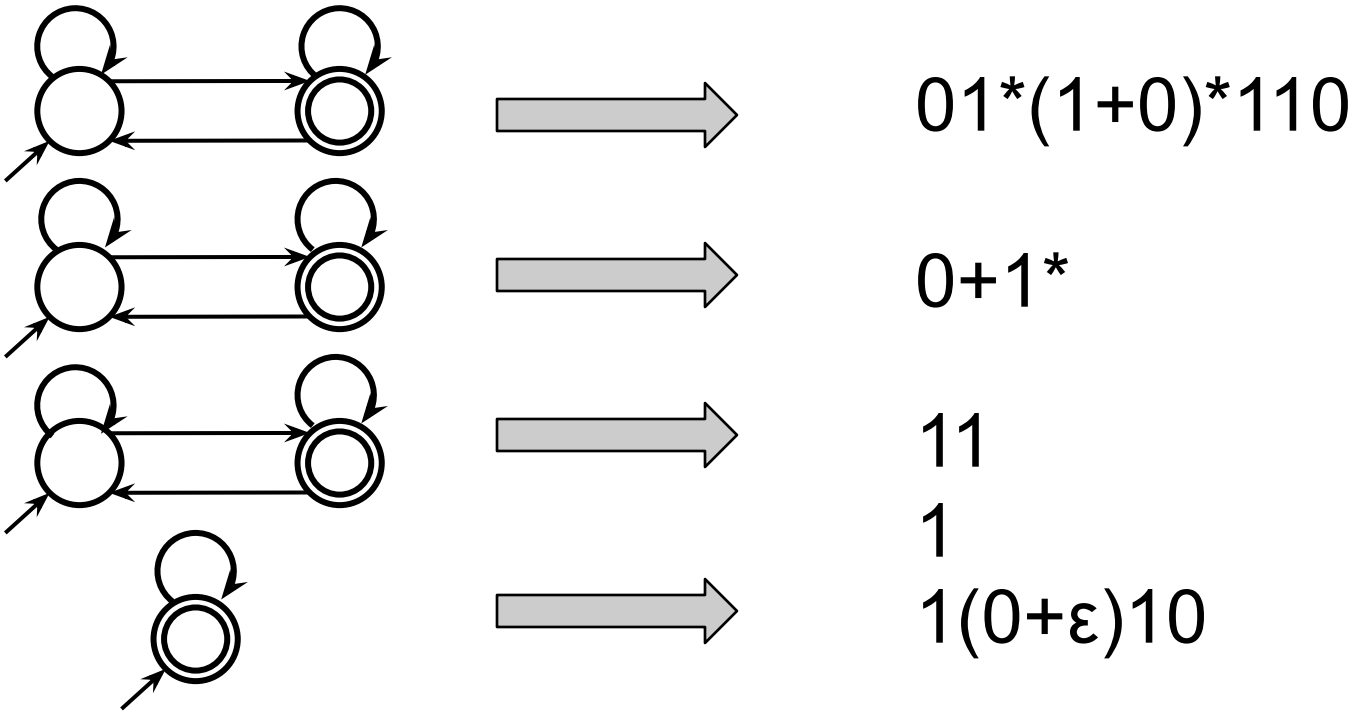
Conversão DFA→ER

- Para cada estado de aceitação q, elimine todos os estados, com exceção de q e q0 (estado inicial)
  - Resultado = um autômato para cada estado de aceitação



Conversão DFA→ER

- Cada autômato terá uma ER equivalente



Conversão DFA→ER

Basta fazer a união de todas as expressões

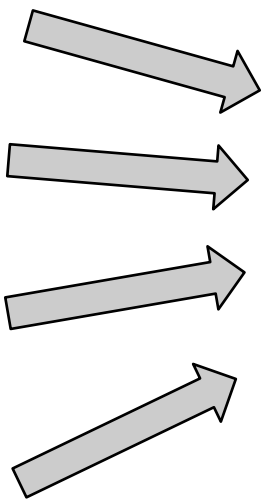
01*(1+0)*110

0+1*

11

1

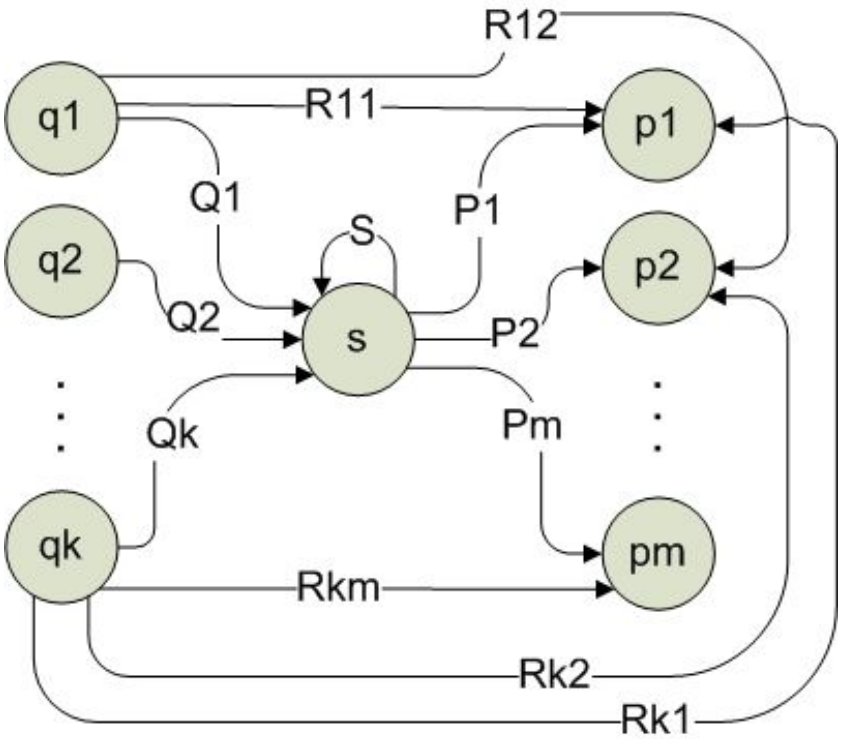
1(0+ε)10

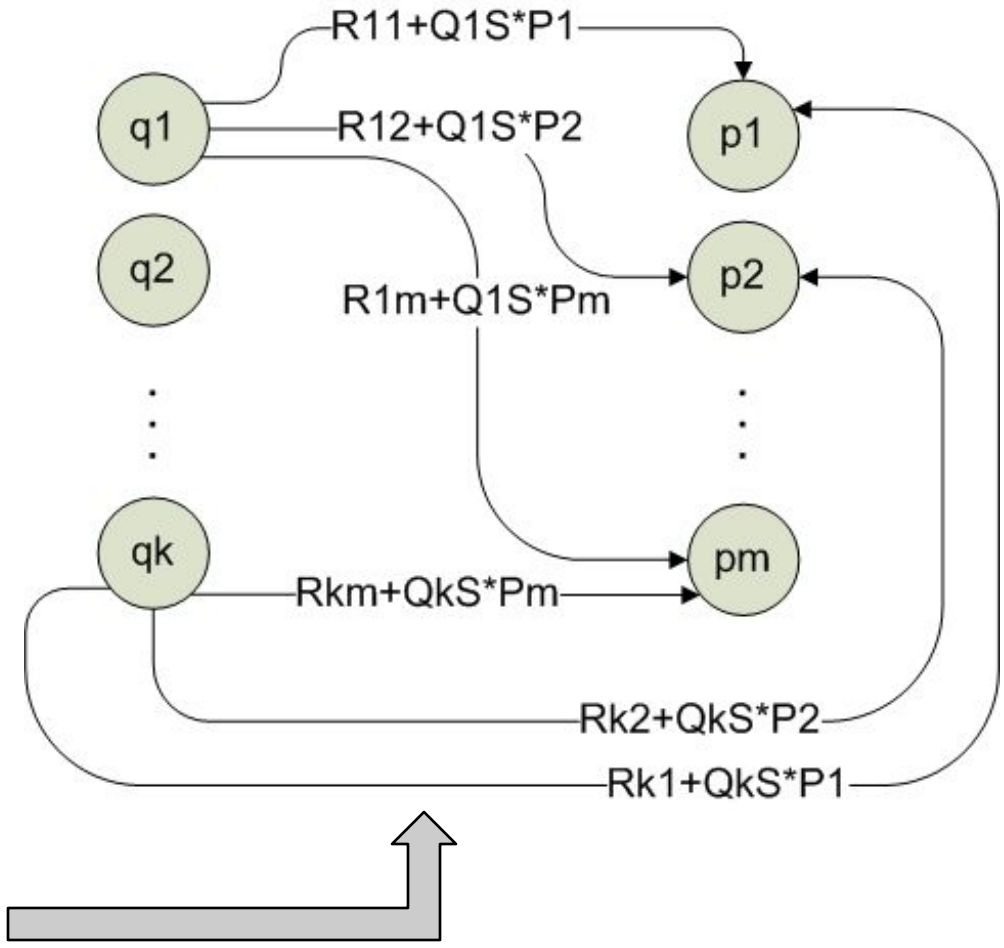


01*(1+0)*110 + 0 +
1* + 111 + 1(0+ε)10

Conversão DFA→ER

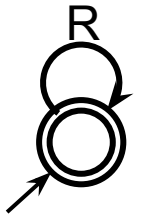
Eliminando um estado s (caso não haja um determinado arco, considerar que existe um arco com rótulo Ø)



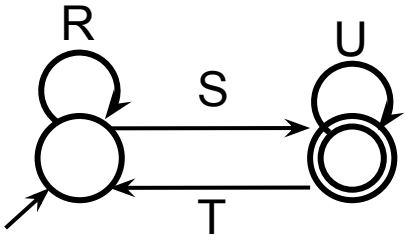


Conversão DFA→ER

- Repetir esse procedimento para todos os estados
- No final, existem duas possibilidades:
  - q0=q
    - Resta um único estado, com uma transição R
    - R é a expressão regular equivalente



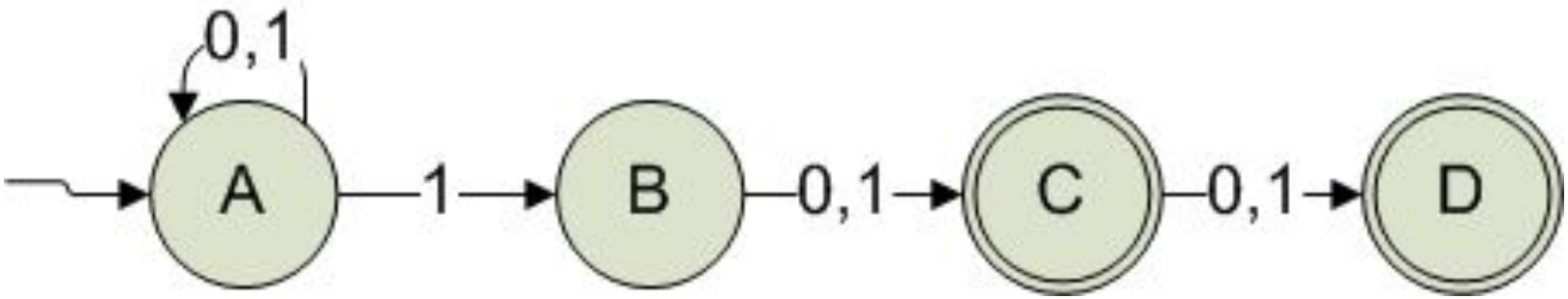
  - q0≠q
    - Restam dois estados, no seguinte formato genérico:



A expressão regular final é (R+SU*T)*SU*

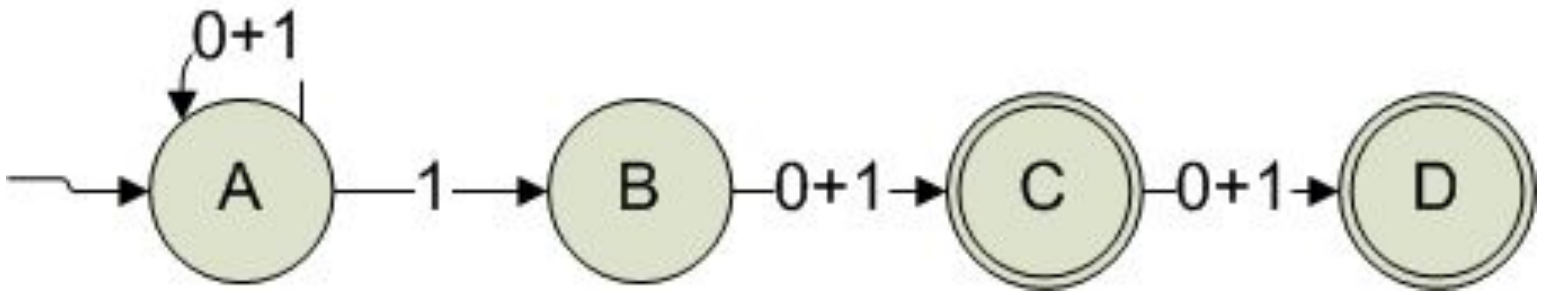
Conversão DFA→ER

- Exemplo: cadeias com símbolo 1 a duas ou três posições a partir do final



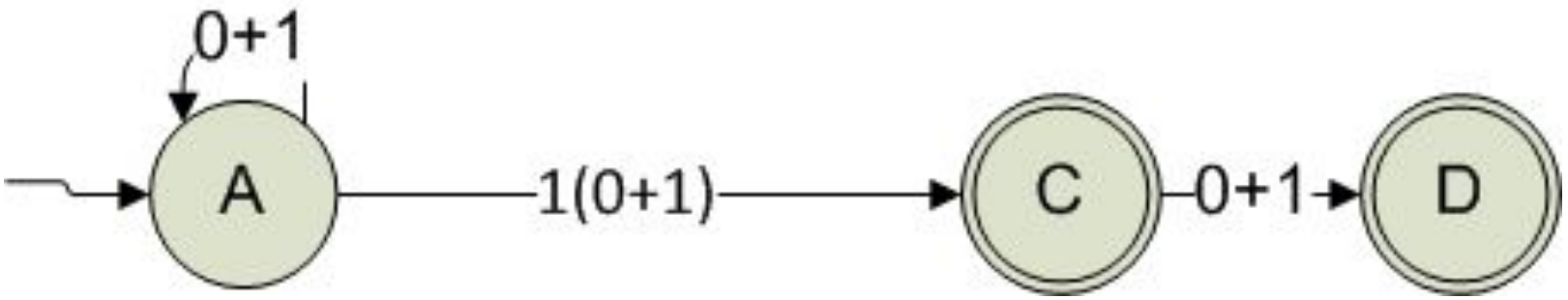
Conversão DFA→ER

- Primeiro passo, substituir as transições rotuladas 0,1 por 0 + 1



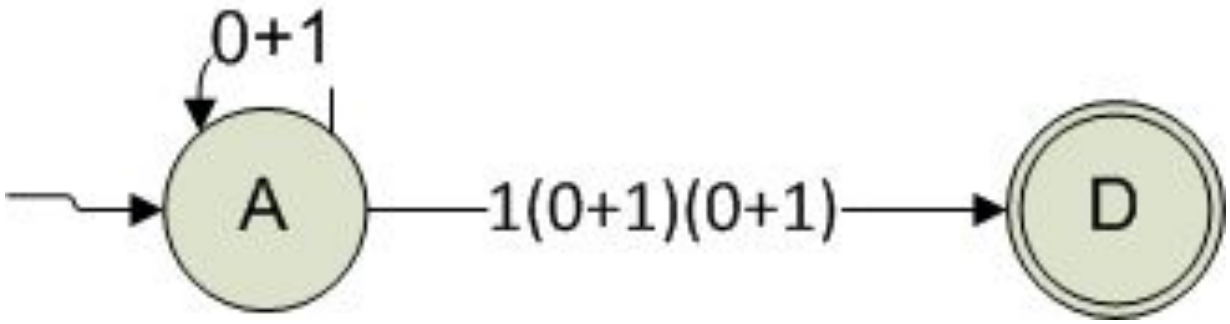
Conversão DFA→ER

- Eliminando B (assim dá para reaproveitar em outras reduções)
  - Q1=1
  - P1=0+1
  - R11=Ø
  - S=Ø
- Arco de A para C = R11+Q1S*P1 = Ø+1Ø*(0+1)
  - Simplificando: 1(0+1)



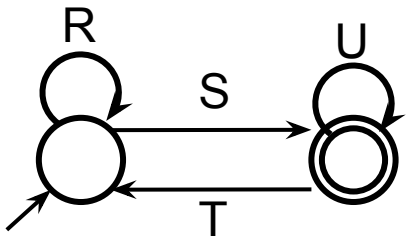
Conversão DFA→ER

- Eliminando C
  - Q1=1(0+1)
  - P1=0+1
  - R11=Ø
  - S=Ø
- Arco de A para D = R11+Q1S*P1 = Ø+1(0+1)Ø*(0+1)
  - Simplificando: 1(0+1)(0+1)



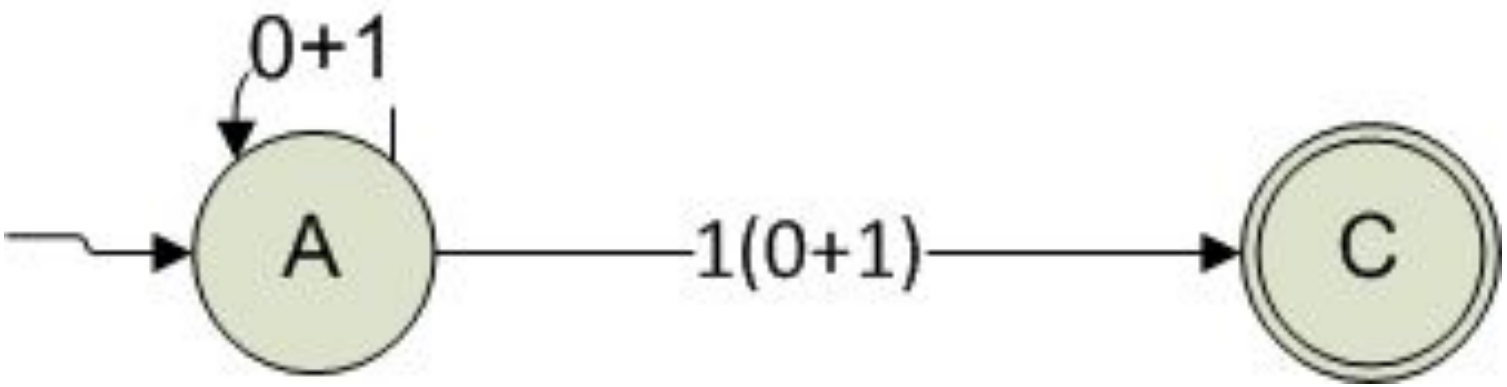
Conversão DFA→ER

- Restou um autômato de 2 estados
  - R=0+1
  - S=1(0+1)(0+1)
  - U=Ø
  - T=Ø
- Fórmula: (R+SU*T)*SU*
  - Resultado: (0+1+1(0+1)(0+1)Ø*Ø)*1(0+1)(0+1)Ø*
  - Simplificando: (0+1)*1(0+1)(0+1)



Conversão DFA→ER

- Eliminando D agora
  - Não há sucessor, portanto não haverá mudanças de arcos
  - Da mesma forma, restou um autômato de 2 estados
    - Expressão regular resultante: (0+1)*1(0+1)

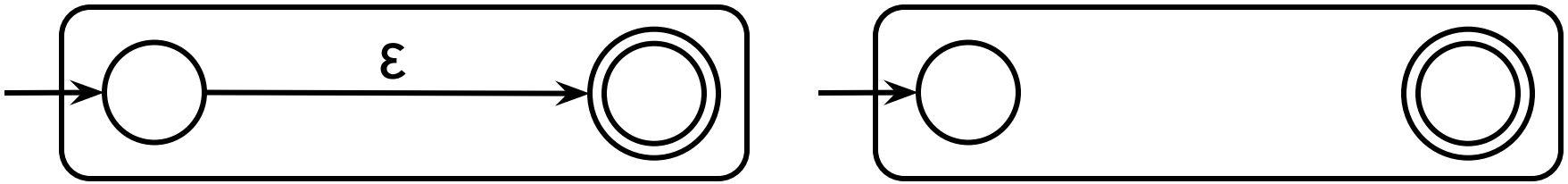


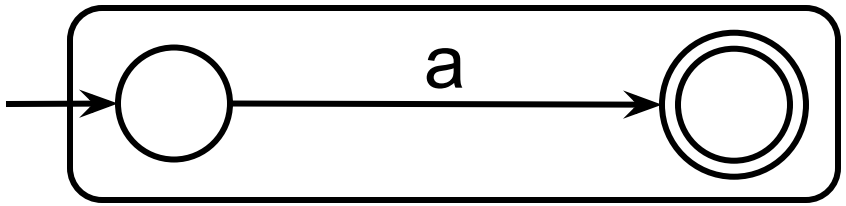
Conversão DFA→ER

- Haviam dois estados de aceitação
  - Foram obtidos dois autômatos
    - Duas expressões regulares equivalentes
      - A expressão regular final é a união dessas duas:
      - (0+1)*1(0+1)+(0+1)*1(0+1)(0+1)
- Simplificando
  - (0+1)*1(0+1)(0+1)?

Conversão ER→ε-NFA

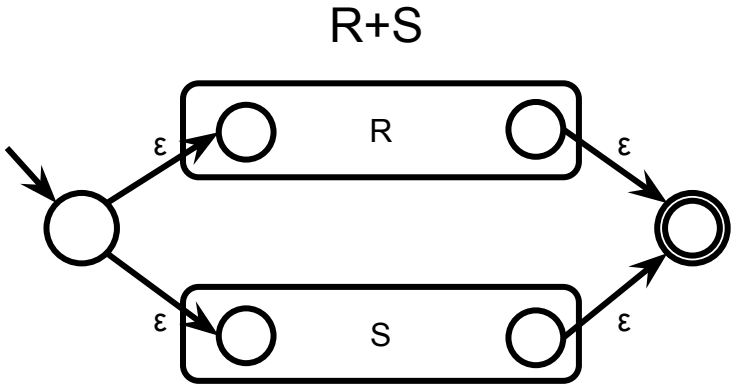
- Teorema: Toda linguagem definida por uma expressão regular também é definida por um autômato finito
- Prova por construção: ER→ε-NFA
  - Base + indução
- Base: ε, Ø e a (um símbolo qualquer)

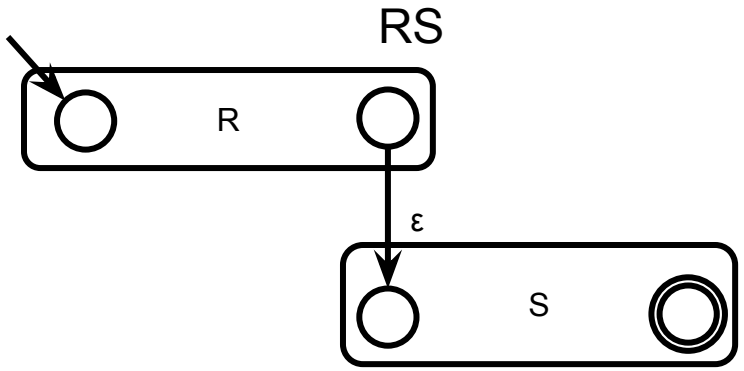


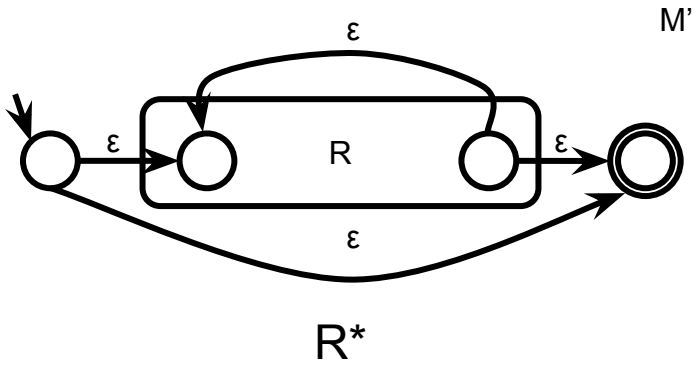


Conversão ER→ε-NFA

- Indução: Os mesmos autômatos das provas sobre o fechamento das operações regulares





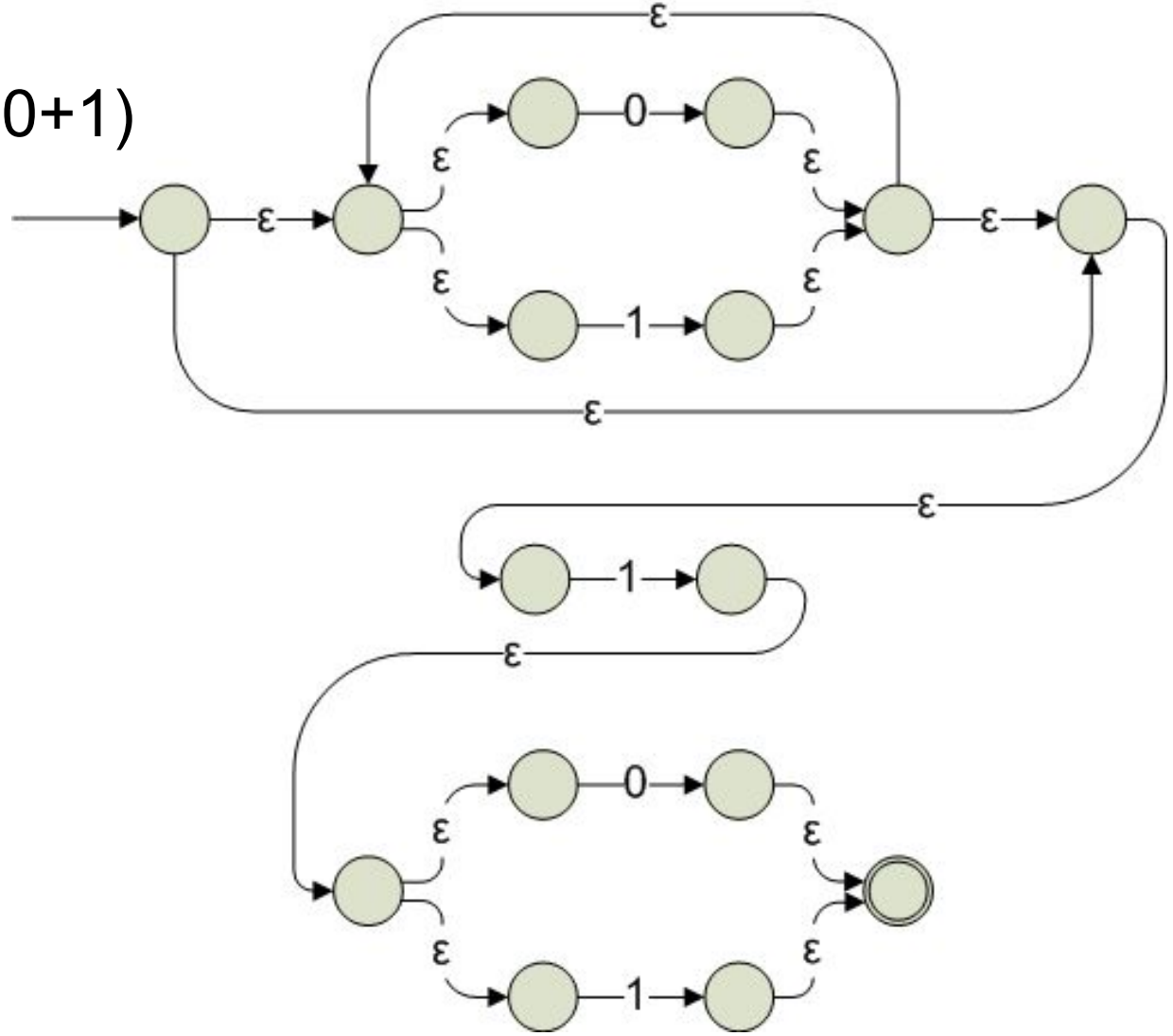


Conversão ER→ε-NFA

- Características do ε-NFA:
  - Possui exatamente um estado de aceitação
  - Nenhum arco chega no estado inicial
  - Nenhum arco sai do estado de aceitação

Conversão ER→ε-NFA

- Ex: (0+1)*1(0+1)



Fim

Aula 09 - Autômatos finitos e expressões regulares